O'REILLY®
Technical Guide

Unlock Data Agility with Composable Data Architecture

Seamless Interoperability, Efficiency & Adaptability at Scale

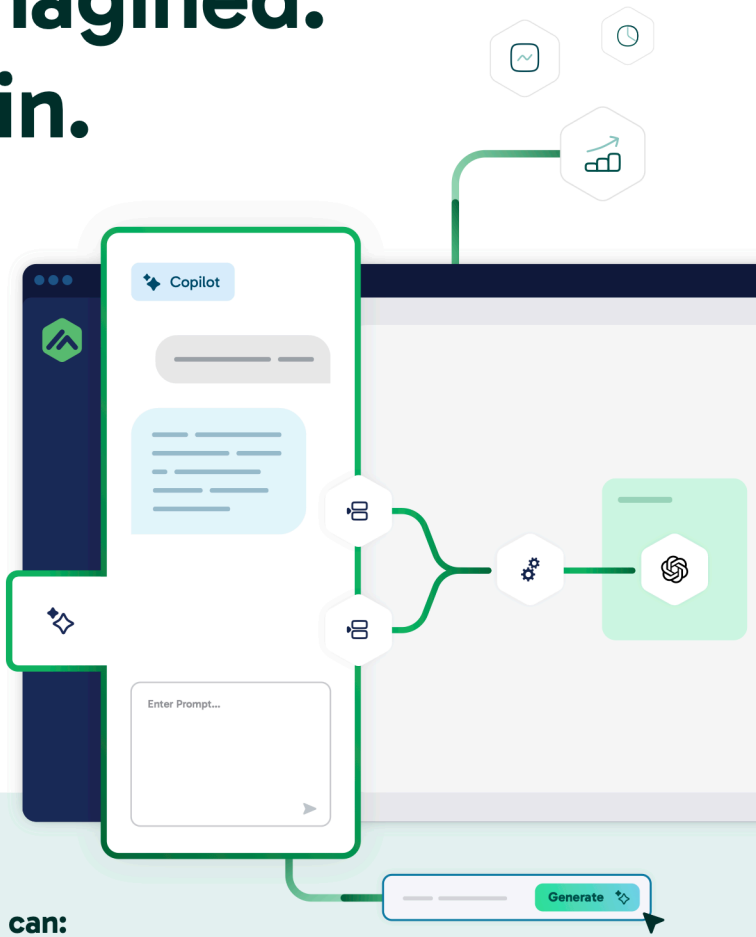
Adam Morton

matillion

ETL reimagined. AI built in.

Matillion's unified platform is the next step in data integration, with AI that enhances data productivity and empowers data teams to deliver powerful analytics at scale and speed.

Unlock the full potential of all your organization's data – structured, semi-structured and unstructured – and leverage GenAI to build pipelines in seconds, not hours.



With one platform you can:

Say yes to multiple apps and use cases.

Quickly and easily create complex pipelines for analytics and AI use cases, enabling accelerated insights for improved decision-making.

Say yes to empowered data teams.

Empower non-technical users with Copilot and the low-code visual Designer. Give skilled engineers the tools they need to focus on value-driving work.

Say yes to limitless scalability.

Leverage elastic pushdown architecture that takes advantage of the full capabilities of your chosen CDW. Effortlessly scale as you grow.

Book a demo

matillion

Learn more

Unlock Data Agility with Composable Data Architecture

Seamless Interoperability, Efficiency & Adaptability at Scale

Adam Morton

O'REILLY®

Unlock Data Agility with Composable Data Architecture
by Adam Morton

Printed in the United States of America.

Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

O'Reilly books may be purchased for educational, business, or sales promotional use. Online editions are also available for most titles (*http://oreilly.com*). For more information, contact our corporate/institutional sales department: 800-998-9938 or *corporate@oreilly.com*.

Acquisitions Editor: Aaron Black
Development Editor: Michele Cronin
Production Editor: Katherine Tozer
Copyeditor: nSight, Inc.
Proofreader: Krsta Technology Solutions
Interior Designer: David Futato
Cover Designer: Susan Brown
Illustrator: Kate Dullea

April 2025: First Edition

Revision History for the First Edition
2025-04-08: First Release

See *http://oreilly.com/catalog/errata.csp?isbn=9781098178949* for release details.

978-1-098-17892-5

[LSI]

Table of Contents

Foreword

As we enter a new era of cloud data and AI, one truth becomes increasingly clear: the ability to build flexible, intelligent data architectures stands as the cornerstone of business innovation. While organizations race to adopt the latest cloud services and AI capabilities, those who cling to rigid, monolithic systems risk falling behind in an increasingly dynamic landscape.

Today's enterprises must orchestrate an expanding ecosystem of data services—from cloud data platforms and AI models to analytics tools and intelligent agents. The fundamental challenge lies not just in connecting these components, but also in creating architectures that can evolve and adapt while maintaining governance and trust.

This book arrives at a crucial moment in our industry's transformation. It presents a comprehensive vision for composable data architectures that leverage AI-driven intelligence to automate and optimize data operations. In this book we demonstrate how modular, interoperable components can work together to create systems that scale efficiently while remaining secure and governable.

As someone deeply involved in intelligent data integration, I've witnessed how organizations that embrace composable architectures and AI-driven automation gain significant advantages in agility and innovation. This book provides a blueprint for building future-ready data platforms that can evolve with changing business needs while maintaining operational excellence.

Whether you're a data leader charting your organization's technical strategy or a practitioner implementing these systems, you'll find practical guidance for navigating the shift toward more intelligent, adaptable architectures. The journey toward next-generation data

architecture may be complex, but with the principles of composability and AI-driven intelligence as your foundation, you can create systems that not only drive innovation but also deliver lasting business value.

— Ed Thompson, CTO and Cofounder, Matillion

CHAPTER 1

Data Management in the Age of AI

Over the past few years, the use of AI within organizations has largely been behind the scenes, hidden from customers. Recommendation engines from companies like Netflix and Amazon quietly suggested what to watch or purchase next, and as consumers, we gave little thought to the vast amounts of personal data being collected. In exchange for improved, personalized experiences, we entered into these agreements, rarely considering how our data might be used. Over time, these companies amassed trillions of data points on us—ranging from individual demographics and buying behaviors to our transactions and browsing habits.

As technology advanced and became more accessible, this enormous pool of data transformed into the fuel for training more sophisticated AI models. By 2022, this quietly progressing trend burst into the public eye with the introduction of ChatGPT. No longer was AI a behind-the-scenes technology used only by organizations—it became available to everyone, reshaping how the public viewed its potential.

In this chapter, we examine how the integration of AI and data management is transforming modern organizations. To provide clarity and focus, we explore three distinct but interrelated areas.

AI's Role in Data Transformation

AI is revolutionizing data management by automating complex tasks such as data integration, preparation, quality assurance, and

ongoing maintenance. These advancements reduce the manual workload for teams and enable more efficient operations, allowing organizations to streamline their data pipelines and improve overall data management processes. Additionally, AI-driven tools are helping organizations maintain consistency, identify anomalies, and ensure data integrity at scale, addressing many traditional challenges in managing vast datasets.

The Modern Data Management Imperative

While AI enhances data management processes, the foundations of effective data management remain vital. Modern data infrastructures are designed to be flexible and scalable, addressing the growing volume and variety of data collected by organizations. These infrastructures must enable rapid ingestion, transformation, and delivery of insights to meet business needs while adhering to an increasingly complex web of regulations. For organizations to succeed, they need robust frameworks that prioritize data quality, governance, and compliance while delivering actionable insights that drive growth and efficiency.

Bridging AI and Data Management

The intersection of AI and data management is where true transformation occurs. Flexible, modular data architectures powered by AI not only enhance operational efficiency but also enable the development of AI-driven applications that support real-time analytics and decision making. These architectures must balance the dual demands of managing data effectively and powering innovative AI solutions. By intertwining these capabilities, organizations can harness the full potential of their data while staying compliant and agile in a rapidly evolving landscape.

Introduction to Data Management

While data is often assumed to be a highly valued asset within organizations, the reality is more complex. Many companies are struggling to keep pace with the sheer volume of data they collect and to manage the tension between protecting this data and using it commercially to generate revenue. Compounding this challenge is

a lack of common understanding within organizations about what data is available and how it contributes to business value.

The vast amounts of data that businesses collect—ranging from customer behaviors to operational processes—hold the potential to fuel customer acquisition strategies and open up new revenue streams. This data also uncovers opportunities to streamline processes and improve efficiency. For business leaders, leveraging data effectively can lead to more informed decision making, helping to reduce organizational risk and drive strategic success. However, without a clear framework for managing, securing, and extracting value from data, many companies find themselves falling short of realizing its full potential.

In too many companies, the benefits of data remain undefined. In a study querying chief information officers, 72% indicated that issues related to data—whether stemming from its poor quality, inadequate accessibility, or lack of effective utilization—are more likely than other factors to jeopardize the achievement of their AI goals by 2025. This highlights the critical need for organizations to not only collect data but also establish robust frameworks to harness its potential effectively.

This requires a data management program that rethinks traditional approaches, ensuring rapid, analytics-based insights that inform decision making and drive growth. The ability to quickly deliver new insights, enhance customer experiences, and swiftly adapt to market changes is critical for maintaining a competitive edge.

As part of this evolution, some forward-thinking companies are experimenting with leveraging AI to modernize their data management systems. These companies are adopting composable data architectures, which offer flexibility and scalability, enabling businesses to adapt to changing needs while expediting the flow of insights. By integrating AI with modular, composable architectures, organizations can underpin their data programs with the speed and innovation required to stay ahead in today's dynamic market.

The Evolution of Data Architecture

Before we explore the future of data architecture, particularly composable architectures, it's important to understand how we arrived at this point. Over the past few decades, organizations have undergone

digital transformations to remain competitive and meet the rising expectations of both customers and key business stakeholders. The increasing demands of the business, combined with the overwhelming volume and variety of data, have underscored the need for a modern, agile data strategy—one that focuses on identifying and leveraging opportunities within data assets to drive value.

To provide context, we'll revisit the evolution of data architecture through two distinct phases: the Enterprise Data Warehouse (EDW) era and the Logical Data Warehouse era, as defined by Gartner. In this section, we'll examine the defining characteristics of each architecture and the specific challenges they presented, shedding light on how these past approaches have shaped the data strategies organizations are now adopting.

The data lakehouse represents an important advancement in data architecture, although it is not classified as an "era" by Gartner in the same way as EDW or Logical Data Warehouse. Unlike the earlier phases, which were characterized by widespread industry adoption and a distinct, overarching paradigm, the lakehouse concept is still emerging. It builds on the principles of the Logical Data Warehouse era while addressing its limitations, such as data movement and fragmentation. As a result, the data lakehouse is best viewed as an evolution within the broader shift toward more flexible and modular architectures, rather than as a separate era in itself.

Understanding these historical contexts is key to appreciating the shift toward architectures like the lakehouse, which promise to meet today's complex data needs by combining the strengths of past approaches with new capabilities.

The Enterprise Data Warehouse Era

When we think about traditional monolithic architectures, we often picture on-premises data warehouses where a single, centralized system processes and stores all the organization's data. While this architecture allowed for consolidated data management, it often lacked flexibility, making it difficult to scale or adapt as data volumes grew and the needs of the business evolved. This centralized model, though reliable in its day, became increasingly strained under the pressure of modern data demands. These traditional architectures typically consisted of two main components:

Data warehouse
: Fast, efficient access to data leveraging SQL for data consumers who need to generate reports and insights for decision making.

Data marts
: A focused subset of data within the data warehouse required by a single team or a business unit.

The primary goal of traditional monolithic architectures was to offload data from business-critical, customer-facing operational systems into an "offline" data store, where integrated business data could be used for reporting. This separation was designed to prevent overloading the source systems and reduce contention for shared resources. However, the on-premises data warehouse systems used in this approach had some notable limitations.

In these systems, compute and storage were tightly coupled, which often led to either under-provisioning or over-provisioning; both are scenarios you'd ideally want to avoid. Installing, configuring, and scaling these warehouses required the procurement of expensive, dedicated physical hardware, a time-consuming and costly process. Before these systems could become operational for business use, significant resources had to be invested.

In this architecture, data was typically processed through a rigid ETL (extract, transform, load) pipeline. ETL tools followed a predefined schema and structure to transform data into an optimized format for high-performance queries and ensure data integrity before loading. While effective in delivering reliable reports, this process significantly limited flexibility. If data needed to be moved or repurposed later, it incurred additional costs and risk. Any changes to the data structures often required manual rework and extensive testing, with the potential to break existing data pipelines.

Moreover, traditional reporting tools were designed to meet enterprise-wide reporting needs, relying heavily on structured data sources. This focus limited the integration of diverse data types and sources, making it difficult for businesses to adapt quickly or innovate with new types of data. As a result, the slow pace of change in these systems drove business users to bypass the central data teams, creating "shadow IT" silos. These silos led to fragmented data analysis and inconsistent insights across the organization, making it difficult to establish a single source of truth.

At the same time, cloud-based solutions began to emerge, offering greater scalability and access to on-demand compute resources. Businesses were also grappling with an explosion of data volume and variety, which pushed traditional architectures to their limits. This combination of factors led to the development of the Logical Data Warehouse era—a more flexible approach that aimed to address the shortcomings of monolithic architectures while enabling faster innovation and responsiveness to business needs.

The Logical Data Warehouse Era

This phase, known as the Logical Data Warehouse era, focuses on providing data teams with a diverse set of capabilities to handle the scale and variety of data required by modern businesses. By considering the specific needs of different data types and use cases, architects can determine the most appropriate "home" for the data, ensuring it resides in the optimal environment for access, performance, and cost-effectiveness.

In this architecture, the traditional EDW and its associated data marts remain a core component. However, they are complemented by three additional primary elements, which provide flexibility and scalability to address the growing complexity of data management:

Data lake
: A data lake is a way to store structured or unstructured raw data as object blobs or files at scale. Unlike traditional approaches that require structuring data before storage (known as "schema-on-write"), data lakes use a "schema-on-read" approach. This means you don't need to define the structure of the data up front; instead, the schema is applied dynamically when the data is accessed for analytics, dashboards, visualizations, or machine learning (ML) models. This approach allows for faster ingestion of data, as it eliminates the need for preprocessing, making it easier to get data into the hands of consumers to generate new insights. In contrast, "schema-on-write" involves defining the structure of the data (e.g., column names, data types, relationships) before it is stored, as is common in data warehouses. While this ensures data consistency and optimizes performance for predefined queries, it can limit flexibility and slow down the process of ingesting diverse data sources. The "schema-on-read" approach of data lakes provides greater agility, particularly when dealing with varied and rapidly changing datasets.

Operational data store (ODS)
: This is a storage layer that sits between the data sources and the data warehouse. It provides data consumers with a way to access near-real-time reporting related to transactional data. This aims to both reduce the need to access live operational systems—as we saw with the EDW era—and provide faster access to data than a data warehouse could provide.

Semantic layer
: The introduction of these additional components means that data for different purposes can be served from multiple places. To simplify access, a semantic layer provides a common interface for applications, tools, and users to access data in a consistent and unified way.

Initially, this approach involved maintaining separate platforms for the data lake and the data warehouse. While data lakes were ideal for storing raw, unstructured, and semi-structured data, data warehouses remained the go-to solution for structured data, analytics, and reporting. However, when businesses needed to combine data from both systems—such as running analytics that required unstructured data from the lake and structured data from the warehouse—the data had to be shifted between the two platforms. This movement of data added complexity, increased costs, and created challenges related to data freshness, duplication, and consistency.

The Data Lakehouse

To address these inefficiencies, the next stage of evolution emerged: the data lakehouse. This architectural design aims to unify the capabilities of both data lakes and data warehouses into a single platform. A data lakehouse aims to offer the flexibility and low-cost storage of a data lake for handling raw and diverse data formats while also delivering the high-performance analytics capabilities of a data warehouse.

By eliminating the need to move data between systems, the lakehouse reduces complexity, ensures data consistency, and enhances data freshness for real-time insights. Additionally, this unified approach lowers operational costs and simplifies data management, offering organizations a more streamlined, scalable solution to meet their growing data needs.

There are a few key technology advancements that have enabled the data lakehouse approach.

Metadata layers for data lakes

The metadata layer is a unified catalog that describes all data within the data lake. Catalogs are used to define:

- What data is available within the data lake
- Where the different datasets are located
- The format and structure of the datasets, i.e., columns, names, data types, etc.

Beyond simply cataloging what data is available, the metadata layer plays a crucial role in managing schema evolution, ensuring that changes to the structure of datasets—such as adding, modifying, or removing fields—are handled seamlessly without disrupting downstream processes. Additionally, the metadata layer supports advanced querying capabilities, enabling users to access and interact with data efficiently. By combining database-like functionality, such as schema management and indexing, with the flexibility of a data lake, the metadata layer ensures data consistency, enables transactional operations, and simplifies access for analytics and ML workflows.

With potentially multiple applications needing to read and write to and from this layer, guaranteeing transactional consistency between applications is important. For example, if an ETL process is updating data in the data lake and at the same time an application is trying to read the same data, there's a risk that the application may not get an accurate or complete view of the dataset. The metadata layer mitigates this issue by supporting open file formats such as Parquet and providing database-like features such as the ones mentioned earlier in this section, which ensures that applications accessing the data always retrieve a consistent and accurate view, even during updates.

High-performance SQL execution

In the past, while storing data in data lakes was affordable, it came at the cost of slow access times when trying to retrieve and analyze the data. Today, new technologies have emerged that make querying data from these lakes much faster. This is achieved by storing frequently used data in faster memory (like random-access memory or solid-state drives), through caching hot data, organizing the data in a way that speeds up access, and deploying techniques that help locate the information more efficiently. Additionally, modern processors can run queries faster through optimized methods. By combining all these advancements, data lakehouses can now handle large amounts of data with speeds comparable to traditional data warehouses, as shown in industry performance tests.

Ease of access for data science and machine learning tools

Data lakehouses use open data formats like Parquet, which makes it simple for data scientists and ML engineers to access and work with the data. This means they can easily use popular tools in their field, such as pandas, TensorFlow, and PyTorch, without needing to worry about compatibility issues. These tools are already set up to work with formats like Parquet and Optimized Row Columnar (ORC), making it straightforward for them to interact with the data in the lakehouse.

As shown in Figure 1-1, the evolution of data architecture into a flexible, open system allows for greater accessibility and compatibility with a variety of tools, enabling seamless integration of advanced analytics and AI workloads.

As organizations embrace cloud-based solutions to power their data and AI initiatives, they must carefully evaluate the ecosystem in which these technologies operate. Choosing between a closed or open ecosystem is a pivotal decision that affects flexibility, scalability, and long-term adaptability. Understanding the implications of this choice is critical, as it influences how seamlessly your architecture can integrate with new technologies, adapt to emerging needs, and avoid vendor lock-in.

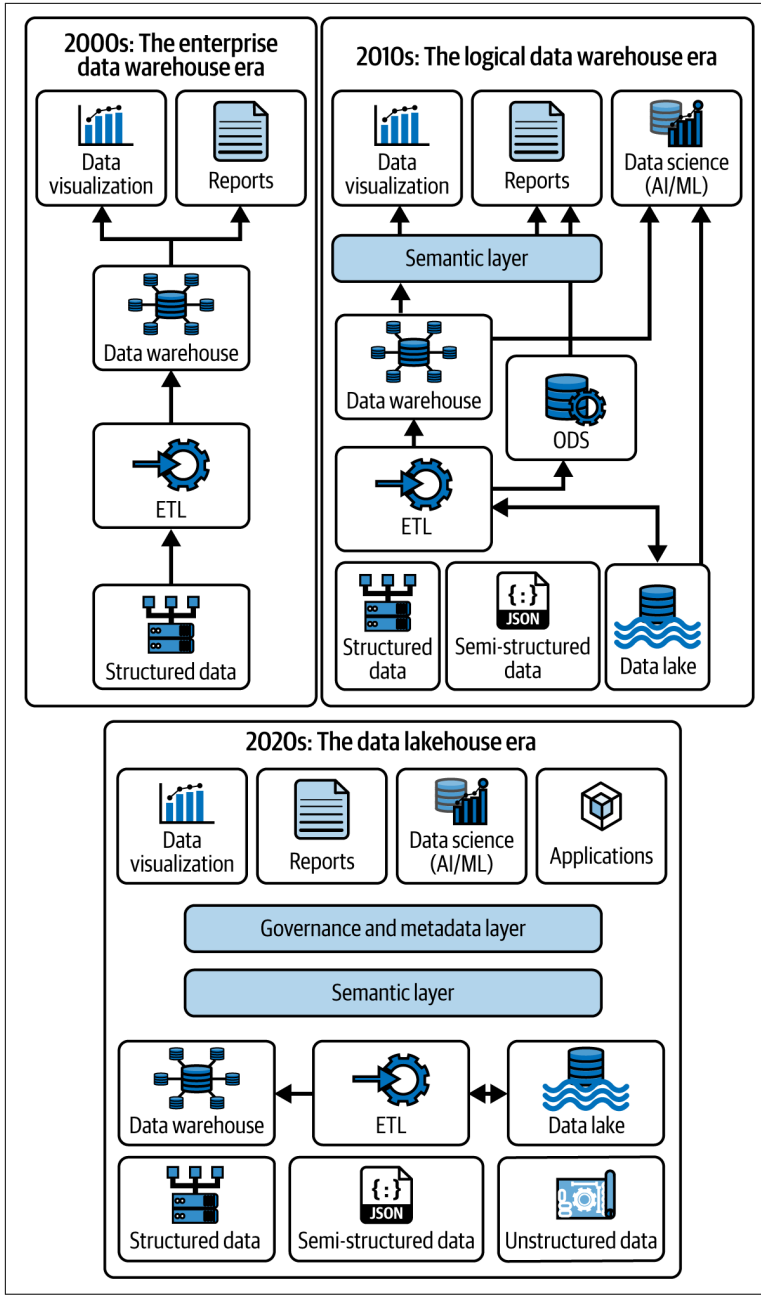


Figure 1-1. The three phases of data architecture

Closed Versus Open Ecosystems

Cloud computing offers significant advantages to companies of all sizes and industries, providing the agility, scalability, and cost-efficiency needed to remain competitive. As a data leader, it's essential to balance factors like agility, total cost of ownership, productivity, and the operational management of a cloud-based data architecture.

However, if not managed carefully, cloud adoption presents its own set of challenges. One key consideration of cloud adoption is making the choice between a closed and open ecosystem. Closed ecosystems, such as Apple's, offer deep integration and optimized experiences within a single vendor's stack, but they may also limit flexibility. For example, Apple customers often stay within the ecosystem because of its seamless integration, but this can also create dependency on Apple's services and devices.

In cloud computing, similar dynamics are at play. The global market for integrated cloud infrastructure as a service (IaaS) and platform as a service (PaaS) is currently dominated by the "big three" vendors: Amazon Web Services (AWS), Microsoft Azure, and Google Cloud. These cloud providers have built extensive IaaS and PaaS offerings, which form the foundation for their software-as-a-service (SaaS) applications. While this integrated stack delivers considerable value, it also deepens the relationship between the customer and vendor, much like a closed ecosystem.

Closed ecosystems, such as Oracle or Teradata, provide robust solutions for advanced use cases like data science and machine learning (DSML), but they may be more difficult to scale to noncore use cases. On the other hand, open ecosystems, such as Snowflake's partner ecosystem, prioritize broader integrations and collaboration across different platforms, providing more flexibility for growing businesses.

For smaller organizations, a closed ecosystem may simplify operations and reduce the need for extensive vendor management. But for larger enterprises or those with evolving needs, an open ecosystem or multicloud strategy offers flexibility to adopt emerging technologies like AI models.

However, this flexibility comes with challenges such as higher data transfer costs, latency issues, and complex security management

across platforms. Ensuring consistent encryption, access controls, and monitoring requires robust tools and expertise. Despite these complexities, careful planning and unified security frameworks can help mitigate risks, making multicloud strategies viable for scalable and adaptable architectures.

Larger organizations, on the other hand, may initially benefit from vendor lock-in due to group discounts, the ability to pool resources, and the opportunity to share knowledge and data internally. This can create efficiencies, especially when departments or subsidiaries need to access shared datasets across a common platform.

However, as organizations grow and evolve, their needs may expand beyond what a single vendor can offer. This is where challenges emerge. The existing vendor may lack the capabilities to meet new demands, prompting the company to explore new technologies. The problem arises when the existing tech stack lacks interoperability with newer systems. Migrating away from an entrenched vendor comes with significant risks, including system downtime, lengthy timelines, and high project costs.

The key to long-term success in cloud architecture lies in preserving flexibility. By designing a cloud architecture that balances single-vendor efficiency with the agility to integrate multiple platforms, companies can aim to get the best of both worlds. This gives organizations the freedom to navigate between ecosystems, allowing them to adopt new technologies as they emerge without being constrained by the limitations of a single vendor's stack.

What Are the Key Risks with a Closed Ecosystem?

There are several key risks associated with a closed ecosystem (Figure 1-2) that are important to address before exploring strategies to mitigate them:

Proprietary technologies
: Many vendors design their platforms using proprietary technologies that aren't compatible with other systems. For example, some cloud data platforms store data in a proprietary format to enhance performance and optimize storage. While this can be beneficial for certain use cases, the trade-off comes when you need to transform or consume the data—only the vendor's software can access or manipulate it, limiting flexibility and creating dependencies.

Existing integrations
: Over time, systems within the current architecture can become tightly coupled with the existing vendor's technology. These deep-rooted integrations make it difficult to introduce new technologies, as they may not seamlessly interoperate with the existing stack. This lack of interoperability presents significant challenges when onboarding new vendors or expanding capabilities.

Skills and experience
: Teams naturally develop expertise in the systems they work with, which, over time, creates a reliance on the current vendor's technology. Transitioning to new technologies requires retraining and upskilling employees, which can be both time-consuming and costly. This learning curve may slow down the adoption of new platforms, further entrenching the organization in the existing ecosystem.

Security and compliance
: Bringing new technologies into the organization requires rigorous due diligence from cloud security and compliance teams. Every new system must be vetted to uncover potential security vulnerabilities, ensure secure integration and authentication with existing systems, and comply with company policies and regulatory requirements. This added layer of scrutiny can complicate the process of adopting new vendors.

Lengthy contracts
: Vendors often lock customers into multiyear contracts by offering significant discounts at the start of the agreement. While these discounts can be appealing, they can also discourage switching vendors midcontract due to hefty penalties. This makes it financially challenging for businesses to adopt new technologies, even when a better option becomes available.

While closed ecosystems pose these risks, open ecosystems are not without their challenges. Managing integration complexities, maintaining data consistency, and ensuring security across multiple platforms can increase operational overhead. However, open ecosystems offer greater flexibility, interoperability, and the ability to adopt new technologies, making them a compelling choice for organizations seeking long-term adaptability.

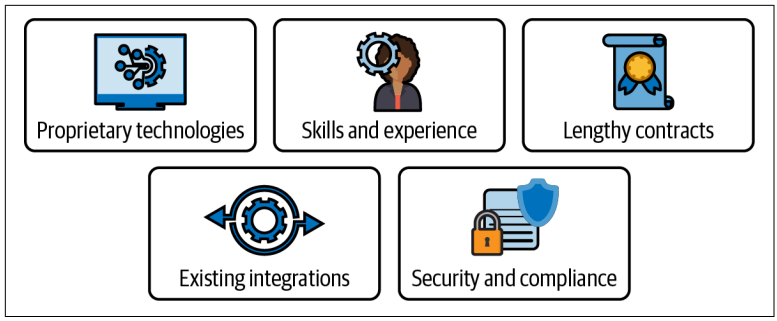


Figure 1-2. Key risks of a closed ecosystem

As a data leader, accepting a certain level of vendor lock-in is inevitable, as it often comes with some advantages, such as streamlined operations, specialized services, or cost efficiencies. However, the real challenge is determining how much lock-in is acceptable and identifying strategies to mitigate the associated risks. By balancing the benefits with these potential drawbacks, you can ensure your data strategy remains agile, adaptable, and capable of evolving with your organization's needs.

How Can You Reduce the Risk of Ending Up in a Closed Ecosystem?

If you're looking to avoid being locked into a closed ecosystem, there are several critical factors to consider when selecting a SaaS provider.

Interoperability and open standards

Open standards are reusable agreements designed to make data publishing, access, sharing, and usage more straightforward and of higher quality. Choosing a vendor that prioritizes interoperability and commits to incorporating open standards in their products and future roadmaps is key to avoiding lock-in.

Since open standards aren't owned by any individual, business, or entity, they are a vital tool for organizations looking to mitigate the risks of vendor lock-in. However, it's important to carefully examine a provider's commitment to open standards, as vendors often integrate these standards only at the basic data access points, leaving other areas proprietary.

Within the data space, open table formats are a direct application of the open standards philosophy. Data tables are a fundamental building block of data services and products, and using open formats ensures compatibility across platforms. The advantage of open table formats is that they allow data to be stored in one format and place yet still be accessible by any software, vendor, application, or partner in the same way. This significantly simplifies architecture, speeds up development, and reduces costs, providing flexibility and freedom to move between vendors or add new services without major disruptions.

Notable open table formats include Apache Iceberg and Delta Lake, both of which offer robust solutions for managing large-scale data lakes while maintaining data integrity. These formats help break down vendor-specific silos by allowing seamless data interoperability, making them a strong option for organizations seeking to future-proof their data architecture. We'll delve into more detail about these formats later in this guide.

Cloud agnostic

To reduce dependency on a single cloud platform, it's essential to work with vendors that can operate on multiple cloud platforms. This flexibility allows you to incorporate a multicloud design, spreading your reliance across multiple providers. With a presence on more than one platform, you can leverage the unique services offered by different cloud vendors and reduce the risks associated with being locked into a single provider. This multicloud strategy also simplifies future migration efforts between applications, giving your organization the ability to pivot as necessary.

Data ownership and access

Ensuring that vendors provide easy options to export or unload data in a common, nonproprietary format is vital. This guarantees that your organization maintains control over its data and can migrate it if needed. Retaining ownership of your data extends beyond day-to-day access—it also applies to backups and disaster recovery.

A real-world example of this is the case of Unisuper, an Australian pension company, which faced significant downtime in May 2024 when Google mistakenly deleted its customer account, jeopardizing critical business data worth $125M. Unisuper had data replicated across two geographies, but the deletion impacted both locations.

Fortunately, it had an additional backup with another service provider, which saved it from a potential disaster. This underscores the importance of a multicloud strategy and solid backup policies that ensure full data access and control, regardless of the vendor's platform.

One of the most widely adopted techniques to ensure data availability and reduce data loss is to adopt the 3-2-1 rule to counter these scenarios. Following these three principles can help:

- Make 3 copies of all data you want to protect so that you'll have 2 backups and 1 primary file.
- Store these files in 2 different locations or media types, such as with different cloud service providers, to have immunity against a broader spectrum of attacks.
- Keep 1 copy off-site and/or in on-premises storage.

Data security and compliance

In the event of a data breach, your organization may need to act quickly to minimize the impact. Ensuring that you have a well-defined plan, including the option to switch to another cloud provider while maintaining business continuity, is essential. During the vendor selection process, assess the portability of your data to avoid vendors that offer limited flexibility.

Additionally, to stay competitive and secure, organizations need access to the latest security innovations. A multicloud strategy enables you to use the best security tools available across different platforms, ensuring a more robust defense against evolving threats. Relying solely on one vendor's security features could introduce third-party risks, especially if the vendor experiences vulnerabilities that are outside your control. By being cloud agnostic, your organization gains the agility to move toward vendors that can adapt more quickly to security challenges.

Key Takeaways

Vendor lock-in and interoperability are crucial considerations for any data leader adopting cloud-based solutions. While cloud services offer enormous benefits, overreliance on a single vendor can limit your flexibility and result in challenges, including difficulty integrating new technologies, handling proprietary data formats,

and facing expensive migration processes. To reduce these risks, organizations should prioritize:

- Interoperability through open standards
- Multicloud strategies to leverage various cloud services and reduce vendor dependency
- Data control, ensuring easy export options and comprehensive backup protocols

By carefully weighing these factors and balancing the trade-offs, businesses can enjoy the advantages of cloud services while minimizing the risks associated with vendor lock-in.

The Emergence of Generative AI

Generative AI (GenAI) is no longer a novelty. More organizations than ever are regularly using it, with 4 in 10 deploying GenAI in more than two business functions. The technology's potential is no longer up for debate.

After the initial wave of excitement around GenAI, companies are beginning to gain a more nuanced understanding of what truly works and what doesn't in driving business value from this technology. The focus has shifted from isolated pilots and proofs of concept (POCs) toward the development of enterprise-scale solutions that can be deployed across the organization.

Today's consumers expect to interact with your business on any channel they choose, at any time, and GenAI and large language models (LLMs) hold the keys to unlocking this challenge at an unprecedented scale. The disruption caused by AI has the potential to fundamentally reshape how businesses build applications to serve their customers. The growing demand for these technologies is spurring a race among companies to unlock the full business value they offer. To remain competitive, your organization needs the speed and agility to deliver seamless customer experiences across all these channels.

As a result, many organizations are seeking ways to deliver more value as efficiently as possible. We believe it's crucial to consider a composable data architecture and determine whether it's a strategy that aligns with your business goals.

While many organizations have been working with AI solutions for over a decade, the release of ChatGPT marked a paradigm shift in how AI is approached. GenAI is now being used not only to enhance existing AI services but also to drive entirely new initiatives and data products. This transition represents a key turning point as businesses move beyond experimentation and begin integrating GenAI into their core operations, positioning themselves to capture long-term value from these tools.

To support this wave of innovation, technology vendors are offering AI-powered development tools that augment existing data and analytics roles and boost productivity. This shift isn't about replacing technical roles, but empowering them. In fact, a recent survey of IT professionals by Salesforce claims 86% of IT professionals say their jobs have become more important since the introduction of GenAI.

Before initiating any GenAI project, it's essential to establish a clear, tangible business goal rooted in business value. Leaders should avoid the "shiny toy" syndrome—experimenting with technology for its own sake. Organizations that succeed will focus intently on business value, not merely embedding tools into existing processes but reimagining data pipelines and the surrounding end-to-end workflows.

Organizations are beginning to understand the resources needed to achieve these goals and create significant impact. In the latest McKinsey State of AI survey of enterprise customers, 67% of respondents said they expect their organizations to allocate more of their technology budgets to AI and GenAI over the next three years.

For this to be impactful, data leaders must be prepared to manage change and navigate ambiguity and uncertainty as the technology evolves faster than ever. A flexible yet resilient data analytics architecture, one that is cost-effective and high-performing and provides a foundation for future pivots, will be essential in the years ahead.

Integrating What Works Now with Future Trends

If you ask a GenAI-based chatbot to "write an email" without providing context or necessary inputs, the resulting email would likely be irrelevant, lacking value, and ultimately unhelpful in improving

productivity. As users, we need to carefully engineer the prompt by specifying key details such as the recipient, the email's subject, the desired tone, and the overall objective. While this is essential for getting a useful response, gathering and inputting this information each time can become a tedious and time-consuming process.

LLMs are pretrained on massive datasets from various public sources like news articles, books, social media, and code. This extensive training enables them to adapt to a wide range of tasks, which is beneficial when the questions or tasks we need answers for aren't predetermined—like when using ChatGPT. However, in a business setting, where the need for accuracy often outweighs the need for broad flexibility, this generality can become a drawback.

In these cases, purpose-built chatbots, designed with a narrower focus, are more effective. These models can be fine-tuned for specific business applications, ensuring that they provide precise, reliable information. By integrating context-specific data and ensuring that the chatbot is aligned with business objectives, companies can build AI tools that prioritize accuracy and relevance over the general-purpose adaptability of larger LLMs.

The Benefits of Retrieval-Augmented Generation

Retrieval-augmented generation (RAG) is designed to enhance the performance of LLMs by retrieving relevant information from an external database, which acts as a knowledge base, before generating a response. For example, if the chatbot has access to your previous emails, it can analyze past interactions with a recipient, understand the context of your relationship, recognize your writing style, and anticipate your likely objectives. This not only saves time but also ensures that responses are more accurate and contextually relevant. The RAG approach helps make generic LLMs more precise, reduces the likelihood of hallucinations, and enhances relevance by grounding the output in real data.

In addition to RAG, it is also possible to fine-tune LLMs for specific tasks. Fine-tuning works particularly well when you have a labeled dataset of historical information to guide the model's training. For instance, you could train an LLM using historical maintenance reports from engineers, which may be stored as unstructured PDFs. By labeling specific reports that indicate system failures, the LLM can learn to recognize patterns associated with potential break-

downs. This fine-tuned model could then be used to spot early warning signs of infrastructure failure, enabling proactive maintenance and reducing the risk of downtime or outages. Both RAG and fine-tuning techniques offer significant benefits for improving accuracy and tailoring LLMs to specific business needs, ultimately driving greater value from AI implementations.

Applying Intelligence to Data Management

This is where technology vendors can play a pivotal role. Instead of developing generic functions that offer limited value, they can take on the responsibility of gathering and organizing metadata and contextual information on behalf of the user. By abstracting this complexity away from the customer and embedding it into the software they're already familiar with, vendors can significantly improve productivity. This approach moves us closer to a world where AI copilots are seamlessly integrated into development workflows, enabling users to focus on higher-level tasks without being bogged down by routine processes.

The integration of LLM-powered copilots into applications should feel natural and intuitive, blending smoothly into the user's workflow and taking meaningful actions on their behalf. This goes beyond just generating code; it extends to building data pipelines using natural language, transforming and preparing data quickly and efficiently. With built-in AI capabilities powered by dedicated LLMs, users can create anything from simple tasks to complex transformations in seconds, streamlining development processes.

Data engineering often involves solving problems at the intersection of people and data. AI-enabled workflows are all geared toward helping to shift the focus to strategic tasks, meaning continuously deploying technology that automates repetitive tasks—such as building common processes and pipelines and generating documentation. By automating these routine aspects, we can shift our focus to more strategic, high-value activities that directly contribute to the business's success.

Incorporating AI-enabled workflows into your development efforts offers benefits across various stages of the development cycle, including:

Natural language querying
: Ask data-related questions in plain English, and the AI will generate the corresponding SQL queries for you, simplifying data access.

Data exploration
: Quickly explore and understand new datasets by asking questions in your native language, making it easier to get insights without needing deep technical knowledge.

Query optimization
: Leverage the AI copilot to automatically optimize and improve the performance and readability of your SQL queries, ensuring cleaner and more efficient code.

Instant documentation access
: Got a question about how a service or feature works? Ask the copilot and it will direct you to the latest vendor documentation, saving time searching for the right information.

Visual pipeline generation
: Easily generate visual data pipelines using natural language prompts, speeding up workflow creation and simplifying complex data processing tasks.

Rapid data preparation and transformation
: Perform data preparation and transformation tasks quickly, scaling efficiently to meet the needs of any project.

Empowering all users
: Make GenAI accessible to all types of users, from technical experts to business users, by simplifying complex processes.

Best practices embedded
: The system can bake in best practices, ensuring that any code or pipelines generated automatically leverage the latest features and functionality available.

Context-aware assistance
: With access to project metadata, the system is context-sensitive, providing highly relevant and up-to-date information, tailored to your specific environment and needs.

By using AI to automate and streamline these areas, your development process becomes more efficient, allowing teams to focus on

innovation and delivering value. One key characteristic of composable architecture is its ability to integrate various technologies into a flexible and scalable solution. However, building and operating such a solution requires a team with diverse expertise across all the technologies involved.

Adopting a copilot-based approach in development not only enhances efficiency for experienced pipeline engineers but also democratizes data engineering. This approach lowers the barrier to entry, making data engineering accessible to less experienced individuals by providing a shallow learning curve. AI copilots can guide users through complex processes, empowering a broader range of people to contribute to development efforts while ensuring best practices are followed. This allows organizations to leverage talent more effectively, building resilient and adaptable solutions in a composable architecture framework.

The Role of the Data Leader in AI Strategy

A *data leader* is a strategic professional responsible for guiding an organization's data strategy, ensuring that data is effectively managed, governed, and leveraged to drive business value. They are often tasked with bridging the gap between technical teams and business stakeholders, translating complex data capabilities into actionable insights that align with organizational goals. This role demands a combination of technical expertise, leadership skills, and a deep understanding of how data impacts decision making, innovation, and competitive advantage.

The responsibilities of data leaders extend beyond managing data assets—they must also anticipate trends, address regulatory requirements, and align data strategies with emerging technologies such as GenAI. With this foundation, a data leader's role in the context of AI can be broken down into three critical responsibilities:

Building a strong data foundation
: To power any AI application effectively, data leaders must ensure a robust foundation that integrates business data (structured, semi-structured) and unstructured data. RAG can play a pivotal role in connecting this data to external knowledge sources, ensuring AI models have access to relevant, real-time information. This foundation allows businesses to maintain control

over the data while ensuring the flexibility to power advanced AI applications.

Real-world use cases beyond chatbots
: While chatbots are a popular entry point, data leaders must think more broadly about AI use cases. The real value comes from integrating unstructured data into analytics, embedding AI into business processes, and enabling AI copilots and agentic workflows. These advancements go beyond simple automation, allowing organizations to develop AI agents that can autonomously execute complex tasks, from customer interactions to predictive maintenance and supply chain optimization.

Governance and risk management
: Just as employees can expose organizations to risks, so can AI applications. Data leaders must ensure that bias, security, and governance are built into AI systems from the ground up. This includes managing sensitive data, preventing model bias, and ensuring robust security protocols are in place. Given the potential for AI systems to function as "virtual employees," data governance becomes even more critical as organizations scale their AI initiatives. AI governance frameworks must be comprehensive, ensuring that these "virtual humans" adhere to the same ethical and operational standards as the human workforce.

Summary

The new wave of AI innovation is already reshaping data management in transformative ways. AI is increasingly integrated into applications, enhancing user experiences by delivering personalized insights and automating tasks directly within the interface. For data professionals, the challenge is not just extracting business value from these advancements but also ensuring proper governance and safeguarding data integrity.

In the near future, AI-powered application development will evolve in tandem with composable architectures, enabling businesses to construct systems from modular components. These architectures offer flexibility, allowing organizations to quickly integrate new AI capabilities without the need for full system overhauls. Platforms built to support AI-driven development are democratizing the process, making it easier for more people to contribute to building

applications. These tools will improve productivity by enhancing support, improving testing processes, and generating higher-quality code. With composable architectures, developers can seamlessly mix and match prebuilt components, incorporating AI to create agile and scalable solutions.

The result? Developers, administrators, and architects will spend more time on creative problem-solving rather than on routine tasks like debugging or writing boilerplate code. Composable architectures allow organizations to future-proof their tech stacks, fostering ongoing innovation through the use of AI-driven components as building blocks. This shift not only streamlines workflows but also makes technology development more engaging and adaptable to change.

Looking ahead, AI's role in app development will continue to expand, influencing not just functionality but also enhancing security and collaboration. The fusion of human expertise with AI tools, supported by the modularity of composable architectures, will democratize development, making it accessible to a broader range of users. Low-code platforms integrated with AI will further simplify development while maintaining high standards. As this landscape rapidly evolves, data management must also adapt, balancing innovation with responsibility. Composable architectures will be central to this transformation, enabling organizations to harness the full potential of AI without compromising control or agility.

In this chapter, we've established how AI is reshaping data management, integrating automation, and enabling organizations to handle data at scale. But AI's true potential can only be unlocked when paired with a data architecture that is as flexible and dynamic as the technology itself. This is where composable architectures come into play. In the next chapter, we'll explore how composable data architectures provide the framework necessary to fully leverage AI's capabilities, empowering businesses to build systems that are not only scalable but also adaptable to future innovations. By diving deeper into the core principles and design of composable architectures, we'll see how these systems are key to driving AI-powered insights and operations.

CHAPTER 2
Understanding Composable Data Architecture

In Chapter 1, we explored the evolution of data architectures, from monolithic systems to flexible, modular designs. We also examined how AI is reshaping data management by automating tasks such as integration, preparation, and quality assurance while enabling real-time analytics and decision making. As we move into the discussion of composable data architectures, we build on this foundation by diving deeper into how modular, AI-driven systems enable continuous innovation, adaptability, and efficiency.

Our goal is to help you understand the core concepts and functionalities of composable data architectures, along with their capabilities and constraints. We'll guide you in harnessing the full potential of your data, enabling seamless integration, processing, and powering AI and analytics across diverse platforms and environments. Additionally, we'll explore how these components work together to create a robust, modular foundation that supports continuous AI innovation throughout your organization.

This chapter introduces the concept of a composable data architecture. We highlight the key components that form its backbone, examining how each element contributes to a cohesive, adaptable, and efficient data ecosystem. We'll also compare traditional monolithic architectures with composable architectures, discussing the pros and cons of each. By understanding these distinctions, we aim

to empower data leaders to make informed decisions on the best approach for their specific needs and requirements.

What Is a Composable Architecture?

The composable architecture marks a shift from the traditional model. It offers a flexible and scalable framework that enables organizations to optimize their data management processes. This approach involves building systems in a modular fashion, assembling independent, self-contained, and interchangeable components.

In many ways, the composable architecture shares many parallels with LEGO building blocks. Each brick in a LEGO set serves a specific purpose and can be combined in countless ways to create different structures. Similarly, a composable architecture breaks down components into packaged business capabilities or services that can be "composed" together. Each component has a distinct purpose and clearly defined boundaries.

In Chapter 1, we introduced how AI is transforming data management by automating tasks such as data integration and maintenance. While this shift complements composable architectures, it's important to note that AI is not an inherent feature of composable architectures. Instead, AI serves as an enabler, enhancing their capabilities by embedding intelligence and automation, thereby making these architectures more adaptive and efficient. Composable architectures stand out by layering AI on top of modular components, creating systems that can self-organize and adapt to changing business needs in real time. This is akin to a "self-driving" system, where the architecture can scale infinitely to meet new use cases with minimal friction or manual intervention. This transformation means businesses can adapt to changes faster, deploy new services at scale, and reduce operational overhead.

This approach contrasts with a single, integrated system—often referred to as a "monolithic" architecture—where the entire system is built on a single platform offering all capabilities. In a monolithic architecture, a change to just one component can ripple through the system, increasing risk and reducing flexibility. As discussed in Chapter 1, traditional monolithic systems, while once the norm, lack the flexibility to adapt quickly to the growing data needs of modern businesses. Composable architectures, with AI-enabled intelligence, offer the scalability and flexibility that today's AI-driven businesses

demand. By integrating AI at the core, composable architectures can handle rapid change, reducing manual overhead and increasing operational efficiency.

Comparing Traditional Versus Composable Differences

The most apparent difference between these two approaches lies in their architectural design, as illustrated in Figure 2-1. Composable architectures are built on a modular and decoupled framework, where each component of the data stack is treated as an independent service. In contrast, traditional monolithic architectures employ a unified design, where ingestion, transformation, storage, compute, and other components are tightly integrated into a single platform, which locks you into one architecture and offers little to no flexibility to adapt or scale individual components independently.

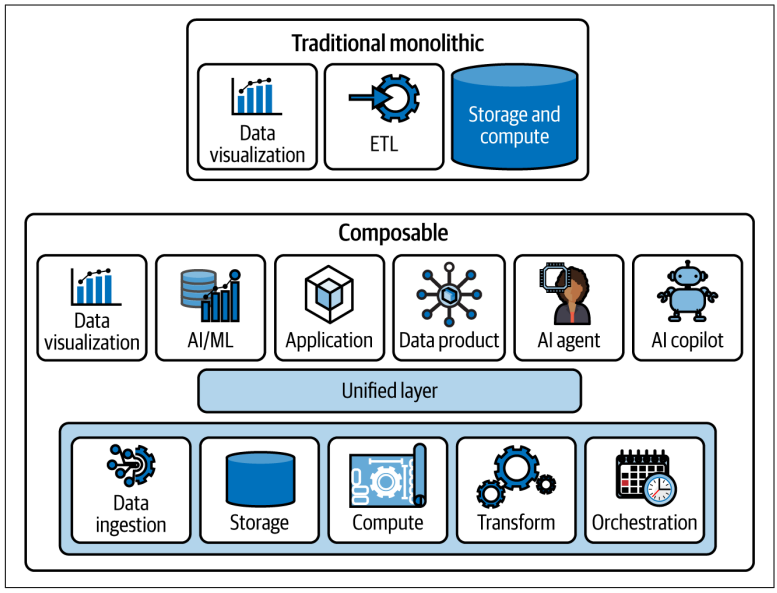


Figure 2-1. Traditional monolithic versus composable architecture

The choice between a monolithic or composable architecture can significantly impact your organization's ability to adapt to evolving business needs, scale efficiently, and integrate new technologies. By examining the pros and cons of composable architecture, you can better assess how it may cater to the diverse needs of your users and applications, ensuring that your data systems are not only effective today but also future-proof. Let's explore the key advantages and

potential challenges associated with a composable architecture to help you make informed decisions for your data strategy.

Pros of composable architecture

A composable architecture offers several key advantages that make it particularly valuable for modern data systems. Let's examine each benefit in detail.

Time to value. The modular nature of composable architecture simplifies the integration of new data sources and formats, enabling quick adaptation to changing business requirements. Developers can build a library of reusable components, which reduces development time and costs while enhancing code consistency and quality.

Futures optionality. One of the key benefits of a composable architecture is the ability to easily apply intelligence to its modular components. This flexibility allows organizations to dynamically adjust their architecture in real time as business needs evolve. In Chapter 1, we highlighted how organizations are using AI to drive new data applications and processes. Composable architectures take this concept further by offering futures optionality, where AI enhances each component's ability to autonomously adjust to new business needs. This adaptability gives organizations the freedom to respond quickly to market changes or new technological advancements without having to overhaul their entire infrastructure.

Reduced risk. A composable architecture provides the flexibility to select the best tools for each task, rather than being confined to a single vendor. The system's modularity allows components to be developed, tested, and deployed independently, simplifying management and maintenance. Organizations pay only for what they need, avoiding the costs associated with all-in-one platforms where only a portion of the capabilities are utilized.

Optimization of resources. Composable architecture enables the independent scaling of compute and storage, optimizing both cost and performance. This scalability extends to other system capabilities, allowing specific areas to be optimized as needed. Changes can be made to the ecosystem without the need for a complete replatforming, facilitating continuous improvement and adaptation to new technologies and evolving customer demands.

Cons of composable architecture

While the benefits of composable architecture are compelling, it's also important to consider the potential challenges that come with this approach. Like any technology decision, there are trade-offs that could affect your organization's operations, security, and overall complexity. We aim to provide you with a balanced perspective to help highlight the considerations that might influence your implementation strategy.

Increased operational overhead. Managing a distributed system of independent components requires robust monitoring, orchestration, and troubleshooting capabilities. Organizations must invest in tools and processes to ensure these components work seamlessly together, and failure to do so can result in fragmented workflows, inefficiencies, and an increased risk of downtime. This complexity often necessitates a greater investment in DevOps expertise and operational tooling, which can stretch resources and budgets, especially for smaller teams.

Security and data privacy. While composable architecture offers flexibility, it also introduces complexity in managing security and data privacy. With multiple integrated components and services, specific threats such as misconfigured APIs, which can expose sensitive data, and data breaches across interconnected services become significant risks. To mitigate these, adopting a zero-trust architecture ensures that every interaction between components is verified. Additionally, implementing strong encryption standards for data in transit and at rest safeguards sensitive information. Regular audits, automated vulnerability scans, and adherence to compliance frameworks like the General Data Protection Regulation (GDPR) or California Consumer Privacy Act (CCPA) further bolster the security posture while ensuring regulatory compliance.

Specialized skills. With a monolithic architecture, organizations could rely on teams with one set of skills based on one central technology platform. However, with a composable architecture, a diverse and blended skill set is required to design, build, test, and deliver data services and products to customers. This creates a challenge for organizations, which either have to invest in their own employees to train them or to source talent with these specialized skills on the open market, who are harder to find and can be expensive to

hire. Companies that fail to adequately train their teams responsible for building a composable architecture can run into a number of pitfalls, resulting in a delayed time to market and additional costs to bring in outside experts.

Lack of clear ownership. A composable architecture is made up of several different technology components. Some of these capabilities can span multiple business processes and teams, which can create gray areas of who is responsible for what. As more services are introduced, the number of teams also increases. Over time, it becomes difficult to know the available services a team can leverage and who to contact for support. Teams will also require ways to collaborate and work together as efficiently as possible, often requiring another level of communication to coordinate development work.

Key Principles

Modern composable systems are built on five key principles: modularity, flexibility, scalability, interoperability, and intelligence (AI). These principles work together to create adaptable, efficient data architectures that can operate at scale (Figure 2-2).

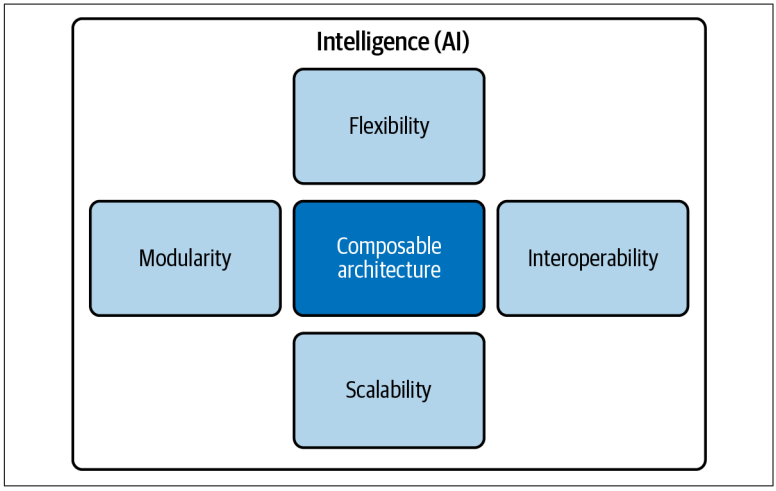


Figure 2-2. The five key principles of composable architectures

These principles provide the guardrails to build systems that can dynamically respond to changing business needs, integrate diverse data sources, and scale seamlessly. Let's break down each of these

principles, exploring their significance and how they contribute to building a robust composable data architecture.

Modularity

Modularity refers to the design of data architecture systems as a set of independent, interchangeable components that can be combined in various ways to meet different needs. Just like each LEGO brick has a specific role to play, each component serves a specific function, making it easier to maintain and manage. Here are some of the benefits:

Customization
: Easily tailor solutions to specific business requirements by selecting and combining different modules.

Ease of maintenance
: Individual components can be updated or replaced without affecting the entire system.

Innovation
: Encourages experimentation and rapid deployment of new technologies by isolating changes to specific modules.

A modular data pipeline might include separate modules for data ingestion, transformation, and storage, allowing organizations to swap out high code with low code transformations easily without disrupting the entire pipeline.

Flexibility

In the same way we can pick up and use a brick from any LEGO set, a composite architecture allows us to adapt to changing requirements, select the best technologies, and work with a diverse range of data sources without extensive reengineering. Here are some of the benefits:

Responsive to change
: Quickly accommodate new business needs, regulations, or technological advancements.

Versatility
: Support a wide range of data types and sources, enabling broader analytical capabilities.

Reduced downtime
: Changes and updates can be implemented with minimal disruption to operations.

A flexible data architecture might support both structured and unstructured data, allowing an organization to integrate new data sources, such as Internet of Things (IoT) devices, with ease.

Scalability

We can always decide to add more LEGO blocks based on the finished product we're aiming for; in the same way, our data architecture needs to handle increasing data volumes and processing demands efficiently as the organization grows. Here are some of the benefits:

Performance
: Maintain high performance levels even as data volumes and user demands increase.

Cost-effective
: Scale resources up or down based on demand, optimizing costs.

Future-proofing
: Prepare the organization for future growth and data challenges.

This ensures that the system can handle sudden spikes in workload without performance degradation, improving user experience and maintaining operational reliability, all while optimizing costs by scaling down resources during off-peak periods.

Interoperability

Just like each LEGO block connects to any other LEGO block in the same way, we want different systems and components within a data architecture to communicate with each other through clear, well-defined connections. This can be achieved through the use of standardized APIs, effective API management practices, and industry standard protocols, to allow seamless integration between components. These technical elements ensure consistent communication, reduce interoperability challenges, and enable secure, efficient data exchange across the architecture. Here are some of the benefits:

Integration
: Easily integrate with existing systems and third-party applications, enhancing data flow and collaboration.

Efficiency
: Reduce data silos and redundancies by enabling different systems to share and access data.

Collaboration
: Foster collaboration across departments by providing a unified view of data.

Intelligence (AI)

Intelligence refers to the integration of AI and ML technologies within the data architecture to enable automation, optimization, and enhanced decision making. Much like a self-driving system that adapts to changes in real time, AI-powered intelligence allows for continuous learning and improvement, ensuring the architecture can anticipate and respond to business needs dynamically. Here are some of the benefits:

Automation
: Automate routine tasks such as data preparation, transformation, and integration, freeing up human resources for more strategic activities.

Optimization
: AI can continuously optimize the architecture by dynamically allocating resources based on usage patterns, ensuring both performance and cost efficiency.

Predictive adaptation
: Leverage predictive analytics to forecast future business needs and proactively adjust data processes, ensuring the system remains agile and responsive to change.

In an AI-augmented data architecture, AI can automatically monitor performance, usage trends, and data quality. It can dynamically reallocate compute resources, predict future data demands, and optimize data storage, ensuring scalability without manual intervention. Additionally, AI can identify opportunities for performance improvements, suggesting new ways to streamline data processes or optimize query execution across the system.

Data Architecture Autonomous Framework and Levels

The parallels between advancements in autonomous vehicle technology and the development of data architectures are strikingly similar. Just as the automotive industry has progressed from manual driving to fully autonomous vehicles, data architectures have transitioned from rigid, monolithic systems to flexible, composable structures that operate with increasing levels of automation and efficiency. Both journeys reflect a shift toward greater flexibility, adaptability, and self-sufficiency, making the comparison a compelling way to understand this evolution.

We have developed a framework that helps us define the concepts of a composable architecture, highlighting how the key principles of modularity, flexibility, scalability, and interoperability—which we'll cover in more detail later in this chapter—are essential to achieving the great levels of autonomy and performance in data systems.

Table 2-1 maps the six stages of data architecture evolution to autonomous levels, illustrating a journey from rigid, manual systems to fully composable, autonomous solutions. Each organization will find itself at a different stage of this journey, from basic automation to advanced composable systems.

Table 2-1. Autonomous data architecture evolution levels

Autonomous level	Description
Level 0: No automation (traditional monolithic architecture)	Traditional data architectures are rigid and centralized, requiring manual processes for data handling and lacking flexibility. Many data tasks, such as failures, dependency management, and orchestration, must be managed individually without automation.
Level 1: Assisted automation (basic tasks automated)	Initial automation begins with basic ETL processes and reporting functionality. Automation handles specific repetitive tasks, but the system lacks the ability to adapt dynamically in response to workloads.
Level 2: Partial automation (modular components)	Modular components are introduced that automate more tasks independently, such as data ingestion and transformation, but that still require significant oversight and integration.
Level 3: Augmented automation (composable architecture)	Data architectures become more composable, allowing dynamic assembly of data pipelines. Systems can operate semiautonomously with modular, reusable components but need human oversight for complex decision making.

Autonomous level	Description
Level 4: Agentic automation (advanced composable systems)	Highly composable systems capable of managing diverse workloads with minimal human input. Data teams can quickly assemble, reconfigure, and deploy data solutions, enhancing adaptability and efficiency.
Level 5: Full automation (fully composable and autonomous)	The ultimate goal of data architecture evolution, where systems are entirely flexible, scalable, and autonomous. They integrate seamlessly with new technologies and adapt automatically to changes in business requirements, driving continuous innovation.

Understanding your current level of data architecture maturity is important, as it helps direct your resources accordingly to reach the next step of evolution. While reaching levels 4 or 5 requires significant investment and development, it highlights the fact that there are tangible rewards to be reaped for those companies who invest in these improvements.

We have identified four distinct areas or phases that span the development lifecycle, including design, execution, and optimization. Tasks associated with these phases can be rolled up and sit beneath these broad headings, as shown in Figure 2-3.

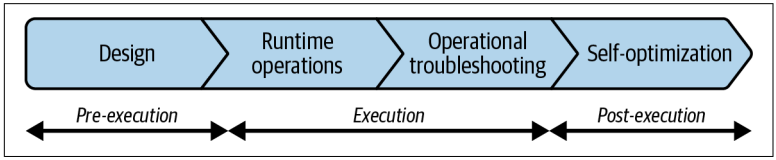


Figure 2-3. Development lifecycle phases covering pre-execution, execution, and post-execution phases

We define these four phases as follows:

Design
: This is a pre-execution phase, building code assets in response to customer requirements. This includes building data pipelines to ingest and transform the data into a format optimized for the method of data consumption, i.e., data visualization, artificial intelligence/machine learning (AI/ML), applications, and ad hoc analytics. Additionally, this phase involves the evaluation of whether existing LLMs can be leveraged or if new models need to be created and trained based on the specific needs of the business or use case.

This phase also includes the creation of environments to store the data, such as compute resource configuration, roles and user management, and the design and creation of database objects, including Data Definition Language (DDL) and Data Manipulation Language (DML) code such as tables, views, and stored procedures. A continuous integration/continuous deployment (CI/CD) and DataOps approach to support code versioning, multiple code branches, and automated deployments and testing also fall into this phase.

Runtime operations
: This phase is all about orchestrating, executing, and running code. This could be as part of development, testing, or production workloads. We need to schedule and trigger jobs in an efficient way to deliver clean, usable data to our data consumers and applications.

Operational troubleshooting
: When executing code, it's crucial to monitor for any warnings or errors and respond to them quickly. Leveraging GenAI and AI-powered autopilots can help automate the identification of errors, resolve common issues, and execute necessary actions, such as backing out erroneous data and restarting pipelines. This automation reduces human intervention and speeds up the troubleshooting process.

  If any erroneous data enters the pipelines, AI can automatically detect the issue, roll back the affected data, and correct the problem without disrupting downstream environments. Additionally, autopilots can restart the pipelines from the point of failure, minimizing downtime and eliminating unnecessary rework. By automating these processes, operational efficiency is significantly enhanced, allowing teams to focus on higher-level tasks.

Self-optimization
: This is a post-execution phase. In this phase, we want to be proactive in spotting potential risks, such as degradation of performance over time, so that we can take action before they become issues. We want our pipelines and data delivery processes to become self-healing and be intelligent enough to reroute queries dynamically for maximum performance and efficiency. We want to avoid underused resources wherever possible.

Breaking down common phases of work associated with developing and operating a data architecture allows us to drill into the role that automation plays at various levels. In Table 2-2, we explore how the different phases of development align with varying degrees of autonomy, illustrating the increasing importance of automation throughout the data lifecycle.

Table 2-2. Development phases mapped to autonomous levels

Autonomous levels	L0: None	L1: Assisted	L2: Partial	L3: Conditional	L4: Advanced	L5: Full
Design	P	P/S	S	S	S	S
Runtime operations	P	P	P/S	S	S	S
Operational troubleshooting	P	P	P	P/S	S	S
Self-optimization	P	P	P	P	P	S

P = People (manual), S = System (autonomous)

Table 2-2 provides a clear framework for understanding how automation can be incrementally applied across the development process. The design phase is an area we believe has the biggest immediate potential to exploit significant value from adopting automation. There are several reasons for this:

- Design and development work is typically the longest part of delivering any new data product or service to the business or to your customers. Many tasks can be standardized with common patterns, which makes them ideal candidates to automate in a repeatable way.
- Many modern tools, such as GitHub Actions, Jenkins, Terraform, dbt Cloud, and Azure DevOps, offer automation capabilities out of the box. These enable CI/CD workflows, including automated environment creation, code deployment, and test execution, streamlining processes and reducing manual effort.
- The design and development phase is not directly customer facing and so the risks are reduced compared with later stages in the development lifecycle that can have a direct impact on the customer and/or business-critical operations. At this early stage in the lifecycle and as teams build their maturity, we would anticipate quality assurance checks and balances by people to form part of the process. Importantly, these checkpoints would

occur *before* any code is pushed to production environments, significantly de-risking the initial adoption of automation.

Moving into levels 3 and above, the journey toward higher levels of automation becomes increasingly impactful. At this stage, automation is not just a tool for efficiency but becomes integral to ensuring consistency, reliability, and scalability across the entire data architecture. The confidence gained from successful automation in earlier phases allows organizations to expand its application into more complex and mission-critical areas, such as runtime operations and operational troubleshooting.

Composable architecture plays a central role in these transformation stages. By breaking down the data architecture into modular, interchangeable components, organizations can more easily integrate advanced automation at each phase of the development lifecycle. This modularity ensures that, as automation is implemented, it can be seamlessly scaled and adapted to changing business requirements. Composable architecture allows for greater flexibility, enabling organizations to assemble and reassemble their data architecture components quickly, optimizing them for specific tasks or levels of automation.

As the automation within the architecture takes over more of the operational activities, it begins to play a crucial role in self-optimization. Systems at these higher levels of autonomy are capable of proactive decision making, identifying potential issues before they escalate, and dynamically adjusting processes to maintain optimal performance without the need for human input. This shift from reactive to proactive operations represents a step change in how data architectures are managed, allowing organizations to minimize downtime, reduce manual intervention, and focus on innovation rather than maintenance. The ultimate goal of these advanced autonomous levels is to create a self-sustaining ecosystem where data flows seamlessly, systems self-heal, and the organization can respond rapidly to changing business needs, all while maintaining the highest levels of data integrity and efficiency.

Exploring Applications and Use Cases

In Chapter 1, we discussed how AI is reshaping traditional use cases across industries, from fraud detection to predictive maintenance.

Composable architectures expand on this idea by supporting a wide array of workloads, from AI-powered applications to data engineering and beyond. By integrating AI into the core of these architectures, organizations can automate and optimize these workloads in real time, delivering faster, more accurate results.

Traditional architectures often fell short due to the need to physically move data between systems before it could be served, causing latency, inefficiencies, and accumulating data debt. For example, a retail company using a monolithic data warehouse might face delays in generating sales reports because real-time IoT data from stores must first be transformed and loaded into the warehouse. These systems typically excelled at structured data and analytics but struggled to handle unstructured or semi-structured data, such as social media sentiment or IoT telemetry, leaving businesses reliant on additional tools to fill the gaps. Moreover, integrating AI in monolithic architectures is challenging due to the rigidity of their pipelines and the lack of support for dynamic, real-time processing needed for AI models, requiring significant custom engineering for even basic AI-driven workloads.

Figure 2-4 illustrates the different workloads a composable architecture needs to support and why flexibility and agility are crucial when it comes to supporting a wide range of workloads.

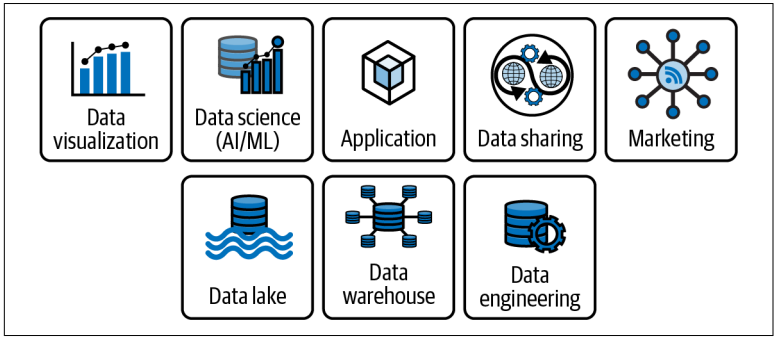


Figure 2-4. The range of data workloads a modern architecture needs to support is vast and diverse

Within these workloads we can identify several use cases, and it's common for use cases to span multiple workloads. This is why we find the need to move data between systems within a monolithic architecture to activate a use case.

Here are some example use cases within the applications workload:

Ecommerce
: An online retailer uses an ecommerce platform, integrating payment processing, inventory management, and customer support systems. This setup allows for seamless updates and the integration of new services without disrupting the entire platform.

Fraud detection (classic AI)
: Leveraging data from transactions, user behavior, and external sources to identify and flag potential fraudulent activities in real time, enhancing security and reducing financial loss.

Agentic workflows (GenAI)
: This refers to AI-powered workflows where autonomous agents, often driven by GenAI or other advanced AI models, take on complex tasks without constant human intervention. These agents can independently analyze data, make decisions, and execute actions within predefined boundaries or objectives. Unlike traditional automation, which relies on static rules or triggers, agentic workflows dynamically adapt to changing inputs and environments, learning and improving over time.

  For example, in customer support, an agentic workflow might involve an AI system autonomously resolving common queries, escalating more complex issues to human agents, and continuously learning from interactions to improve future responses. This approach not only enhances efficiency but also introduces a higher level of intelligence and adaptability, enabling businesses to scale operations and meet diverse needs seamlessly.

Here are example use cases within the data engineering workload:

ETL/ELT data pipelines
: A financial institution uses data engineering pipelines to extract, transform, and load (ETL/ELT) transaction data into its analytics platform. They can easily integrate data quality checks and balances and automate testing on key performance indicators (KPIs) to ensure data integrity.

GenAI for automated data pipelines
: A GenAI-powered system can autonomously build and optimize data engineering pipelines, identifying potential performance bottlenecks and suggesting improvements without

manual intervention. The model continuously adapts to changes in data structure, ensuring optimal performance and minimal downtime.

The following example use cases fall within the data lake workload:

IoT devices for infrastructure monitoring
: IoT devices generate enormous amounts of data in semi-structured or unstructured formats. Energy companies deploy IoT sensors in the field to monitor remote infrastructure, which is critical for delivering energy to businesses and households. When extreme weather events occur, these data streams carry crucial, business-critical information, alerting field engineers to outages and damage to key assets. A data lake ingests and stores the vast volumes of data from IoT devices, enabling rapid identification of potential issues so that field engineers can be notified with minimal latency.

Securely store and access unstructured data at scale
: A healthcare organization stores vast amounts of unstructured data, such as patient records and imaging data, securely in a data lake. This allows them to centralize and manage data for compliance and research purposes, with the flexibility to scale storage as needed. Data can be accessed directly without the cost of modeling data into a predefined schema using a "schema-on-read" approach.

Near-real-time analysis of semi-structured data
: A data lake that integrates GenAI models can analyze semi-structured data from IoT devices in real time, predicting infrastructure failures before they occur. By using historical patterns and ML algorithms, GenAI enhances the ability to prevent outages and deliver proactive maintenance solutions.

Here are some use cases within the data warehouse workload:

Monitoring business performance for strategic decision making
: A retail company leverages a data warehouse to consolidate sales data from multiple stores. The composable architecture allows them to run complex queries and generate reports, helping management make data-driven decisions.

Agentic data warehousing (GenAI)
: In this scenario, GenAI-enhanced systems automatically optimize query performance, forecast future sales trends, and even suggest new insights based on historical data patterns. This empowers decision makers with predictive analytics that adapt to changing market conditions without manual recalibration.

Now let's look at example use cases within the business intelligence (BI) and data visualization workload:

Enterprise reporting and insights
: An enterprise creates executive dashboards that pull data from various sources, including customer relationship management, finance, and operations. The composable architecture supports real-time data visualization, providing leadership with up-to-date insights.

GenAI-driven visualizations
: AI agents autonomously generate and customize data visualizations based on user preferences and trends, automatically adjusting the presentation as new data flows in. These agents can also identify key insights that may not be immediately apparent to human users, suggesting actions or areas of concern.

Here are example use cases within the AI and data science workload:

Classic AI
: A manufacturing company employs AI models to predict equipment failures. The data science workload processes historical data and sensor inputs to generate maintenance schedules, reducing downtime and maintenance costs.

GenAI for predictive maintenance
: With GenAI models, the system not only predicts failures but also autonomously generates repair schedules, orders replacement parts, and allocates resources without human intervention. This moves beyond predictions, enabling fully automated workflows.

These next example use cases are within the data sharing workload:

Secure data sharing for collaboration
: A pharmaceutical company shares clinical trial data with partner organizations. The composable architecture enables secure data sharing, allowing researchers to collaborate while ensuring data privacy and compliance.

GenAI-enhanced data sharing
: GenAI models ensure secure and compliant sharing by automatically encrypting sensitive data, redacting personal identifiers, and ensuring the appropriate access control for different stakeholders. GenAI can also track how shared data is being used, providing insights into collaboration outcomes.

The following example use cases fall within the marketing workload:

Integration of omnichannel customer-centric data
: A marketing team uses a composable architecture to integrate customer data from various sources, including web analytics and social media. This setup allows them to create personalized marketing campaigns that target specific customer segments based on real-time data.

GenAI-powered campaigns
: With GenAI, marketing campaigns become more dynamic. GenAI analyzes customer behavior in real time, adjusting campaign messaging, offers, and channels to increase engagement automatically. The system can autonomously learn and optimize these campaigns without requiring manual input, leading to higher conversion rates.

Use Cases Mapped to Autonomous Levels

In this section, we map the specific use cases described in the preceding section to our data architecture autonomous framework. This helps to define the incremental benefits that can be realized at each level as you transition from a monolithic to a composite architecture leveraging great efficiencies and automation at scale.

For an ecommerce application, the autonomous levels might be as follows:

L0 / no automation
: Integrated systems make updates or the addition of new services cumbersome and prone to disruptions.

L1 / assisted automation
: Basic automation handles repetitive tasks like stock level updates, but changes still cause interruptions.

L2 / partial automation
: Independent modular components enable smoother updates to inventory or payment systems, but with some oversight needed.

L3 / augmented automation
: Ecommerce platform is more flexible, allowing automatic updates across services based on predefined conditions with minimal human oversight.

L4 / agentic automation
: The platform can autonomously manage new service integrations, respond to dynamic changes like flash sales, and reconfigure systems for optimal performance.

L5 / full automation
: Entire platform dynamically adapts and optimizes based on real-time customer behavior, product availability, and business rules with no manual intervention required.

For ETL pipelines, the autonomous levels might be as follows:

L0 / no automation
: Manual management of ETL processes leads to inefficiencies and a high risk of human error.

L1 / assisted automation
: Simple, repetitive ETL tasks are automated, but custom workflows and oversight are still required.

L2 / partial automation
: Modular ETL components enable partial automation of data pipelines, reducing manual intervention but still requiring checks for complex data sources.

L3 / augmented automation
: ETL pipelines can dynamically adapt to data changes, reducing manual reconfiguration and handling varied workloads.

L4 / agentic automation
: Data pipelines autonomously scale and optimize, handling complex data integrations with little human involvement.

L5 / full automation
: Fully autonomous ETL pipelines can self-heal, adjust performance, and optimize data workflows based on real-time conditions without any manual oversight.

For an IoT application, the autonomous levels might be as follows:

L0 / no automation
: Unstructured IoT data requires manual processes for ingestion and storage, leading to high latency in data processing.

L1 / assisted automation
: Some basic automation helps ingest IoT data into the data lake, but significant manual effort is still required.

L2 / partial automation
: Modular ingestion systems automate parts of the IoT data flow, but management of unstructured data still needs human oversight.

L3 / augmented automation
: Automated, flexible pipelines adapt to incoming IoT data streams, with automated alerts for potential issues.

L4 / agentic automation
: IoT data flows are fully optimized in near real time, automatically predicting potential data issues and adjusting pipelines.

L5 / full automation
: IoT systems autonomously manage data, predicting failures, optimizing pipelines, and providing proactive insights without human involvement.

For sales reporting, the autonomous levels might be as follows:

L0 / no automation
: Sales data is consolidated manually, leading to delays in reporting and limited ability to react to real-time trends.

L1 / assisted automation
: Simple automations are in place for sales data consolidation, but reporting still involves manual work.

L2 / partial automation
: Modular components automate most data consolidation, with periodic human intervention required to validate reports.

L3 / augmented automation
: Sales data pipelines autonomously generate reports, reacting to data trends in real time with minimal human oversight.

L4 / agentic automation
: Reports are generated and optimized automatically, providing predictive insights to sales teams with real-time updates.

L5 / full automation
: Sales reporting is entirely automated, with AI-driven insights and recommendations based on real-time data trends, requiring no human intervention.

The autonomous levels for executive dashboards might be as follows:

L0 / no automation
: Dashboards are updated infrequently, and data is pulled manually from various sources, resulting in outdated insights.

L1 / assisted automation
: Basic automation enables periodic updates of dashboard data, but manual intervention is still required for new insights.

L2 / partial automation
: Modular data sources automatically refresh dashboards, though some oversight is needed for complex visualizations.

L3 / augmented automation
: Dashboards are dynamically updated with real-time data, providing executives with actionable insights with minimal manual involvement.

L4 / agentic automation
: Dashboards autonomously visualize and highlight trends, adapting the presentation based on the needs of the business.

L5 / full automation
: Executive dashboards are fully autonomous, providing real-time, AI-driven insights that adapt dynamically to changing business conditions.

For an predictive maintainance application, the autonomous levels might be as follows:

L0 / no automation
: Maintenance schedules are generated manually with little to no automation, leading to inefficiencies.

L1 / assisted automation
: Basic AI models assist in predicting equipment failures, but manual data handling and analysis are still required.

L2 / partial automation
: Automated models generate predictive maintenance schedules, though human oversight is needed for adjustments.

L3 / augmented automation
: Predictive maintenance workflows adapt dynamically based on real-time data, adjusting schedules with minimal human input.

L4 / agentic automation
: The system autonomously optimizes maintenance schedules, ordering parts and assigning tasks based on AI-driven insights.

L5 / full automation
: Entire maintenance processes are autonomously managed by AI, from predicting failures to scheduling repairs and managing inventory without human involvement.

The autonomous levels or collaborative research might be as follows:

L0 / no automation
: Data sharing requires manual processes, with a high risk of compliance issues and inefficiencies.

L1 / assisted automation
: Basic automations support data sharing, but data privacy and compliance checks are largely manual.

L2 / partial automation
: Automated systems manage the data sharing process, but human oversight is needed for privacy and compliance.

L3 / augmented automation
: Data sharing workflows automatically handle privacy and compliance, securely distributing data based on preset conditions.

L4 / agentic automation
: The system autonomously optimizes data sharing, automatically managing privacy, compliance, and access control.

L5 / full automation
: Fully autonomous data sharing system with AI-driven controls that dynamically adjust permissions, ensuring secure and compliant data collaboration across partners.

For personalized campaigns, the autonomous levels might be as follows:

L0 / no automation
: Marketing campaigns are based on static data, with little flexibility to personalize offers in real time.

L1 / assisted automation
: Basic automation enables simple personalizations, but manual effort is required for campaign optimization.

L2 / partial automation
: Modular systems automate more personalization tasks, though manual oversight is needed to fine-tune campaigns.

L3 / augmented automation
: AI models personalize campaigns dynamically, optimizing offers and content in real time with minimal input.

L4 / agentic automation
: Campaigns are autonomously managed by AI, continuously optimizing content and offers based on real-time data insights.

L5 / full automation
: Fully automated marketing campaigns, where AI autonomously adjusts messaging, channels, and offers to optimize engagement and conversion without human involvement.

Summary

These use cases demonstrate how a composable data analytics architecture enables organizations to rapidly integrate and analyze diverse data sources, leading to more informed decision making and greater agility in responding to business needs.

In a composable architecture, the delivery of such a diverse set of workloads and use cases can be handled efficiently and securely. The inclusion of AI, GenAI, and LLMs further enhances this architecture by introducing intelligent automation and adaptability. AI-driven components can dynamically adjust to evolving data needs, streamline processes like data integration, and even optimize resource allocation in real time. Incorporating a unified layer across the set of components allows users to connect using the best tool for their requirements, from notebooks for programming to dashboards, AI/ML tools, applications, and data sharing with third parties.

This intelligent, AI-powered approach ensures a consistent and seamless experience throughout the organization while catering to a diverse range of users. From C-level executives running BI dashboards to monitor the latest sales performance, to field engineers conducting emergency maintenance on critical energy infrastructure, to data scientists building predictive models using Python, the AI-enabled architecture adapts to varying needs and maximizes efficiency.

Armed with this shared understanding of how data architectures are evolving, and the growing role of intelligence within a composable architecture, we've reached the point where we can delve deeper into the specific components. In the next chapter, we will explore how AI-driven autonomy within a composable data architecture leads to improved performance, agility, and scalability.

CHAPTER 3

Designing and Configuring Infrastructure for Composable Architectures

In Chapter 1 we looked at the evolution of data architecture over the preceding years. One observation is that the primary objective has remained consistent up until recently: to consolidate diverse data sources into a central, unified repository for reporting and analytical use cases. Today, some organizations have recognized that centralizing their data assets, which involves integrating many diverse data sources into a single data repository, is either too time-consuming or cost-prohibitive. This significant shift in focus has resulted in the emergence of the data lakehouse, and now composable architectures to support the pressing need to accelerate the adoption and time to value of AI-driven products and data applications.

The key to unlocking GenAI initiatives at scale lies in high-quality, trusted, and curated data and metadata, which form the foundation for activating new AI use cases through a robust data and analytics infrastructure. This means that data leaders must not only critically evaluate their existing architectures, aiming to address challenges such as data silos, data governance and security, inconsistent data quality, and change management, but also consider how the latest wave of GenAI can be leveraged to expedite this shift in focus.

In this chapter, we delve deeper into the specifics of composable architecture by examining the required capabilities, the core components and their roles, how to secure and govern the platform, and effective strategies for managing change.

Centralized Versus Federated

When designing architectural frameworks, data leaders often face a key decision: adopting a centralized or federated approach. In a centralized model—often represented by the data fabric approach—data management and governance are managed centrally, providing consistency, standardization, and a unified strategic direction. This can simplify compliance and streamline decision making but may lack the flexibility needed for diverse business needs with specific needs.

Alternatively, a federated model, such as the data mesh approach, distributes data ownership and responsibility across multiple domains, empowering teams to independently manage their data while aligning with shared governance standards. While federated models offer agility and domain-specific insights, they can introduce challenges around maintaining consistency and oversight as well as requiring a relatively high level of overall maturity across the entire organization.

Later in this section, we will explore why we think a center of excellence (CoE) approach can provide a balanced middle ground, combining the expertise and governance of a centralized model with the flexibility and innovation of a federated approach.

Data Fabric: A Centralized Approach

Before discussing more balanced models like data mesh and CoE, it's important to understand *data fabric* as a centralized, highly integrated approach to data management. Data fabric provides a unified architecture that connects disparate data sources across an organization, allowing data to flow seamlessly while maintaining strict governance, security, and quality standards.

In a data fabric model, data is managed centrally, often supported by advanced technologies such as metadata-driven data integration, knowledge graphs, and AI-based data orchestration. This approach offers strong data governance and consistency, making it ideal for

organizations prioritizing control, compliance, and a single source of truth across all business functions. However, its centralized nature can limit flexibility for individual business units and may slow down innovation at the domain level, as each request for data or analytics insights must pass through a central team.

Data fabric is highly effective for organizations needing centralized oversight and uniformity, especially those in regulated industries where data quality and security are paramount. Yet, as data needs diversify across business functions, the purely centralized approach can lead to bottlenecks and challenges in meeting the varied data demands of each domain.

Data Mesh: A Decentralized, Domain-Oriented Model

The data mesh, put forward by Zhamak Dehghani in 2019, takes a fundamentally different approach, abandoning the idea that "to satisfy analytical use cases, data must be extracted from domains, and consolidated and integrated under central repositories of a warehouse or a lake."[1]

The data mesh is an organizational and architectural approach for sharing, accessing, and managing analytics data in complex and large-scale environments within or across organizations, which consists of four common principles:

Domain-oriented data ownership and architecture
: Business domains are responsible for their own data, so ownership is decentralized. Data mesh promotes the idea that data should be owned and managed by the domains or teams that are closest to it, rather than being centralized in a single data team or warehouse. For instance, a product group could be a business domain; it would own its own data. Business domains manage the data quality, understandability, and interoperability of their data. Domain ownership is designed to help organizations scale the number of data sources that can be used throughout the company.

1 Zhamak Dehghani, *Data Mesh: Delivering Data-Driven Value at Scale* (O'Reilly, 2022), 57.

Data as a product
: This means that data should be managed and delivered with the same care and attention as any other product, focusing on user satisfaction, quality, and discoverability. This means each data product should be well-defined, documented, and easily accessible. The data product must be interoperable with—and able to be joined with—other data products owned by other domains.

Self-service data infrastructure as a platform
: A self-service platform represents a set of capabilities provided as a service to enable teams to access data. Multiple people, from data scientists and power users to executives requiring operational dashboards, all need to be catered to.

Federated governance
: This means that governance policies are applied consistently across all data domains but allow for some flexibility and autonomy at the domain level. It's about striking a balance between centralized standards (such as data privacy and security) and local autonomy for domain-specific needs.

Embracing data mesh necessitates a paradigm shift in how companies perceive and handle data and requires a similarly significant culture shift. We therefore recommend some key questions to ask before considering if it is the right fit for your organization:

Do the benefits versus cost stack up?
: For some organizations it may make sense to move from a centralized to a decentralized model if they have multiple business units and the available staff within each domain where the benefits and economies of scale might outweigh the costs.

What skills are required?
: Each domain will need to think about data as a product. The finance team, for example, may have never experienced a product-driven approach, which requires a new way of working.

What does the organizational model look like?

The company will need to be organized to execute. Although there will be domains, the scope of those domains will need to be determined. Additionally, some organizations won't have all of the skills needed in each domain.

Center of Excellence: Balancing Centralization with Flexibility

CoE is an approach that adopts a hub-and-spoke organizational model to balance centralized control with distributed innovation. At the core of this model is a centralized team that plays a pivotal role in setting data standards, governance policies, and best practices across the organization. This team ensures that data is high quality, clean, secure, and well governed, providing a solid foundation for any data-driven initiatives.

The CoE acts as the strategic brain of the data organization, developing a unified framework for data management that aligns with the overall business strategy. This includes defining data architecture blueprints, security protocols, data lineage tracking, and compliance frameworks to meet regulatory requirements. The goal is to establish a single source of truth that fosters trust in data among users across the organization.

In addition to setting standards, the CoE also provides guardrails and tools that enable self-service capabilities for individual business units, allowing them to leverage the organization's data assets independently while maintaining alignment with central governance. This approach ensures that business units with varying levels of data maturity can access the data they need, whether through self-service analytics or centralized support. The CoE equips them with data catalogs, automated data quality checks, and training resources, empowering them to build their own data products and insights without compromising data consistency or security.

By acting as the central point of expertise, the CoE ensures that business units can innovate at their own pace, leveraging their domain-specific knowledge while relying on the central team's expertise for complex data challenges, such as advanced analytics, AI/ML model deployment, or data integration projects. This model not only drives efficiency and reduces duplication of effort but also ensures that data governance is consistent across all business functions, resulting in a scalable, future-proof data architecture.

The hub-and-spoke structure of the CoE provides the flexibility of a federated model but with centralized oversight, allowing organizations to avoid the common pitfalls of fragmented data management, such as data silos and inconsistent quality standards. It enables business units to focus on innovation and agility, while the CoE maintains the infrastructure, tools, and standards that ensure a cohesive data strategy. This approach not only enhances data-driven decision making across the organization but also maximizes the value of data assets by ensuring they are readily accessible and reliable for all stakeholders.

Designing the Architecture for Composable Data Systems

We believe that composable data architecture represents the next evolution beyond the current data lakehouse approach. Integrating AI-powered applications will not only improve data accessibility for consumers but also serve as a catalyst for significant productivity gains in data engineering teams.

The convergence of data platforms combined with AI-driven intelligence is pivotal to the success of composable data architecture. From this combination, we see two primary types of applications emerging:

Core capability uplift
: Integrating LLMs into the core of the architecture significantly enhances key capabilities. These include code generation and optimization, data integration, the semantic layer, data governance, observability, logging and monitoring, data quality and testing, and metadata search. Each of these areas benefits from the intelligence and automation that LLMs bring, driving greater efficiency and effectiveness across the data platform.

LLM-powered apps
: New supporting capabilities as a result of harnessing the power of LLMs augment the core capabilities in new and exciting ways. AI agents and intelligent assistants form part of this category. These applications are able to harvest the metadata from within the data platform and provide meaningful insights to data consumers through natural language.

Core Components

In this section, we lay the groundwork for understanding the initial set of core components (see Figure 3-1) that form the foundation of a composable data architecture. These components focus on infrastructure and provide the building blocks necessary to support scalable, adaptable, and resilient data platforms. The core components we explore here are not standalone; rather, they are part of a larger ecosystem that enables data integration, processing, and analytics to operate seamlessly.

As we progress through subsequent chapters, we'll expand this list, integrating additional capabilities that address data management, orchestration, and AI-driven insights. By the end, we aim to build a comprehensive view of what's needed to architect a truly composable system. Each component we introduce plays a specific role in achieving this vision, offering unique functionalities while remaining interoperable with others.

Understanding how these components work together will help you navigate key design decisions, balance performance with flexibility, and ensure that your architecture can adapt as new needs arise.

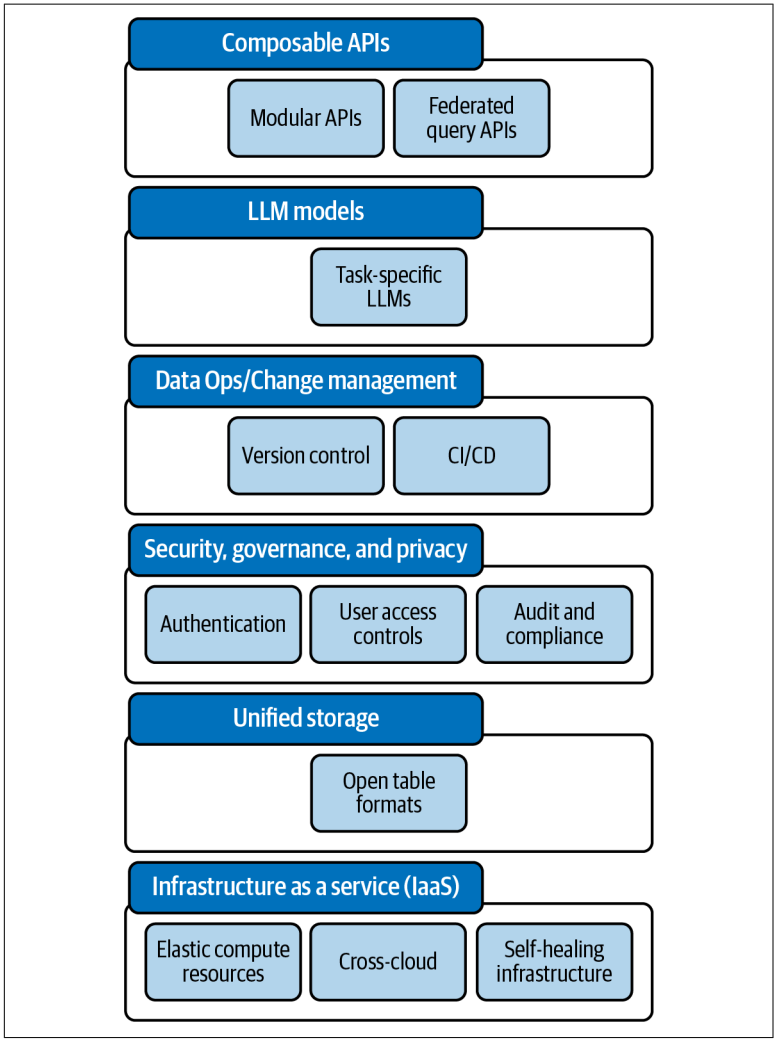


Figure 3-1. Core components (infrastructure-focused)

Infrastructure as a Service

In a composable data architecture, infrastructure as a service (IaaS) forms the backbone, providing flexible and scalable compute resources essential for handling dynamic workloads. IaaS allows

organizations to dynamically allocate resources based on real-time needs, ensuring cost-effective scaling and high-performance processing. As the demand for data-driven insights and AI applications grows, having a reliable IaaS foundation becomes critical, enabling teams to manage everything from basic data operations to complex ML workloads.

In the following sections, we explore how elastic compute and processing and the integration of AI-optimized compute services support the performance, scalability, and agility required in a modern, composable data environment.

Elastic Compute and Processing

Having access to a reliable, flexible, and scalable compute infrastructure that dynamically adjusts compute resources based on workload is of major importance to realizing the composable architecture. You can learn more about how elastic computing resources work on the Microsoft Azure website.

We are starting to see more examples of AI-optimized compute services being interwoven into the fabric of vendor SaaS products, which are using predictive analytics for optimal scaling. This includes GPU-accelerated, on-demand infrastructure to support ML workloads like training and inference, which we'll discuss next.

Dedicated AI and Generative Processing Power

As AI technologies like traditional ML and GenAI models (e.g., LLMs) advance, the demand for dedicated processing power has surged. Efficiently running these models at scale requires high-performance, purpose-built hardware, which has become a critical factor for organizations seeking to leverage AI's potential. This new wave of AI has made high-performance, purpose-built deep learning processors one of the hottest commodities on the market, driving the second quarter fiscal 2025 record revenues of $30.0 billion for NVIDIA—up 15% from Q1 and up 122% from a year ago.

Many SaaS data platforms such as Databricks and Snowflake have started to incorporate these options into their offerings, with Databricks recently striking a five-year deal with Amazon to use their Trainium AI chips—a cheaper alternative to NVIDIA's GPUs, providing more cost-effective options to its customers.

Autonomous Cross-Cloud Solutions

Cloud services initially started out as a way to move away from the tangled web of on-premises legacy technology and realize unparalleled scalability and access to compute resources at the click of a button. As their cloud environments have matured, and more services have been introduced, organizations find themselves struggling with similar issues, such as lack of transparency, visibility, security, and compliance.

Over the next few years, the myriad of on-premises, public, and private cross-cloud solutions will be managed autonomously by an AI-driven layer, making cloud services simple to use once again. This level of intelligence will be smart enough to identify, disable, and report malicious threats; spot areas of high demand and optimize or intelligently reroute workloads on the fly to compute resources with available capacity; and deploy and configure the relevant services to distribute and balance workloads effectively—all without human interaction.

This solution will comprise a suite of software and services that sit above the public cloud providers, creating a level of isolation between the constraints of the physical cloud infrastructure to facilitate ease of cloud deployments. This will help provide greater control, visibility, and observability for organizations. For data leaders, this is compelling, as we may wish to make managing the complexity of the underlying technology stack someone else's problem so that we can focus on adding business value.

Unified Storage Layer

Realizing the vision of the composable architecture requires interoperability and separation of resources such as compute and storage. This separation of resources allows you to store data once and use it many times. The great thing is if you need to scale your low-cost storage, you don't need to scale your more costly compute power at the same time. This provides you with a great deal of flexibility in your operations.

But how do you manage the huge variety of data formats while avoiding some of the common pitfalls of ending up in a closed ecosystem where data is locked into priority formats? In Chapter 1, we touched on some of the key risks of a closed ecosystem along with a range of approaches to help mitigate those potential pitfalls, which included interoperability and open standards. One of the key challenges that data leaders face when aiming to adopt an open composable architecture is selecting the optimal format for their data.

Apache Iceberg, Apache Hudi, and Linux Foundation Delta Lake are all strong choices for storage formats, providing open, flexible options that enhance compatibility and interoperability across systems. These open formats offer clear advantages over proprietary alternatives; however, deciding which format to standardize on can feel overwhelming due to the long-term impact of the decision, often leading to analysis paralysis and concerns about making an irreversible choice.

Companies such as Databricks have provided solutions to these challenges in the past with Delta UniForm. This is a metadata layer over the top of open table formats with the aim of providing better interoperability across all formats and placing the choice of where best to store the data in the customer's hands.

One of the most popular of all open table formats is Apache Iceberg. Developed originally by Netflix before being made open source, it provides a level of abstraction across different cloud data storage services such as Amazon S3, Google Cloud Storage, and Microsoft Azure Blob Storage. It offers similar functionality to SQL database tables, which enables data connectivity into applications that may need to collect data from multiple clouds. It offers database-like capabilities, such as transactional consistency, schema evolution, and time travel, as well as high performance when compared to native database tables.

It's no wonder that both Databricks and Snowflake ended up in a bidding war for Tabular—a start-up looking to commercialize Iceberg, doing $1 million in annual recurring revenue. Databricks eventually won out, acquiring Tabular for a reported $2 billion, signaling how strategically important this acquisition is in this highly competitive market.

Currently, Iceberg adoption is still in its infancy at many organizations, so as a data leader, how do you decide when to migrate? How do you identify the potential benefits and pitfalls to understand if the cost of change is worthwhile?

Often different query engines have their sweet spots; for example, Snowflake may cater better for one type of workload in your organization than Databricks, while for other workloads Databricks excels, as illustrated in Figure 3-2. This is why in larger businesses it's not uncommon to find both catering to different needs.

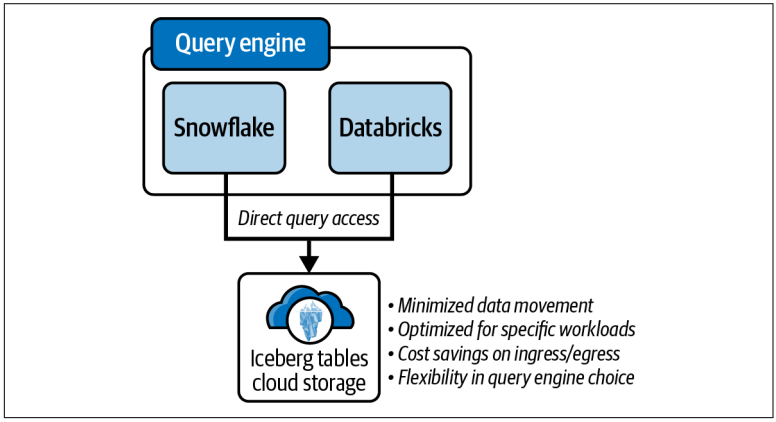


Figure 3-2. Iceberg allows you to pick the best query engine for your needs

Before the broader adoption by these vendors of open source table formats such as Iceberg, you had little choice but to import your data into the platform to get all the performance and features offered by the database, but this meant the data had to be copied away from your underlying cloud storage provider. With Iceberg, you can now bring your query processing engine to where the data lives. This saves on data ingress and egress fees, which, for near-real-time streaming data, can become significant.

With this high-level understanding of open table formats, focusing specifically on Iceberg and some of its general benefits, we'd like to provide some key points to help you in evaluating a move to Iceberg. If you are considering migrating, consider a proof-of-value approach dictated by high-value business use cases that are narrow and laser focused on definitively proving out a predefined set of success criteria:

Migration strategy
: It's important to plan the migration carefully, focusing on POCs for critical use cases and testing data accuracy throughout the process to avoid disruptions and data loss.

Performance optimization
: Iceberg is designed for large-scale datasets, but optimizing partitioning and schema evolution is crucial for reducing query times and maximizing performance. Proper partitioning based on query patterns can minimize data scans and improve efficiency.

Data ingestion and transformation
: Integrating Iceberg effectively requires compatibility with existing ELT (extract, load, transform) tools and workflows to ensure smooth data ingestion and transformation processes. The aim here is to enable seamless integration with Iceberg catalogs across platforms, providing flexibility for users to write data into Iceberg and access it through commonly used data environments.

  By supporting Iceberg catalog integration, users can maintain consistent data management across platforms while taking advantage of Iceberg's unique capabilities for versioning and partitioning. Ensuring transformations are accurately repointed and continue to perform as expected is essential, especially given that not all systems fully support Iceberg's advanced features. Comprehensive testing across your spectrum of workloads helps mitigate risks and achieve reliable, optimized performance with Iceberg.

Data governance and compliance
: Iceberg's versioning and rich metadata allow for better data tracking and compliance, but setting up proper governance policies is essential to maintain security and compliance during and after migration. You should also check if the role-based access control (RBAC) policies you have set up on your data platform, such as Snowflake or Databricks, can be "synced" with the Iceberg layer. If not, what you may save on data redundancy you may pay for by having to maintain your access controls in more than one place.

Integration with BI tools
: Many BI tools like Tableau and Power BI may require additional configuration to support querying data directly from Iceberg, especially when moving from traditional data warehouses. Commonly, we see organizations supporting a hybrid model, with a warehouse used for clean, curated, aggregated data at the presentation (or Gold) layer to provide a balanced solution.

By addressing these considerations, and with an unwavering focus on incrementally establishing business value, your organization can reap the full benefits of an open composable data architecture, leveraging Iceberg where appropriate for better scalability, reduced costs, and improved performance without losing control of its data.

Composable APIs

In Chapter 2, we mentioned that interoperability is one of the five key principles of a composable architecture. This essentially means the ability to bring your own tools to "plug and play" to extract new insights, interrogate and work with data, and leverage the power of the underlying tools and services, all with no changes to the underlying architecture.

Having the most efficient and elegant architecture is worth nothing if no one can access the services and data within it in a frictionless way. This is where the importance of our composable API layer comes in.

The composable API layer consists of two primary services:

Modular APIs
: Provides composable, modular APIs for accessing different components of the data architecture (e.g., storage, compute, LLMs, search), allowing easy integration of AI capabilities into other systems

Federated query APIs
: Supports querying data from multiple sources (e.g., across multiple databases or cloud platforms) through a unified API, simplifying data access for composable architectures

The modular API layer supports some key capabilities within our architecture such as intelligent agents and AI-driven insights, which leverage LLMs harnessing natural language processing for conversational AI. We'll go into these features in more detail in Chapter 5.

The modular nature of these APIs strikes another note within our key principles: modularity. This reduces the maintenance overhead, as changes can be made to individual components without the risk of impacting the entire application, further streamlining the development process.

Within the federated query API we see a common architectural pattern providing isolation over a range of complex, unrelated APIs. In this layer we present a uniform interface to data consumers across multiple domains or subgraphs. Known as a "supergraph," this significantly diminishes the need for manual intervention in data integration and API orchestration.

Adaptive Infrastructure

Building a resilient foundation for a composable data architecture requires the ability to detect and respond to issues automatically. Self-healing infrastructure addresses this need by using data-driven insights to take automated, intelligent actions. This capability becomes increasingly vital as organizations manage development, quality engineering, and production environments across hybrid cloud setups. In these environments, the ability to identify problems and resolve them before they disrupt operations can save time, reduce costs, and maintain trust in data systems.

The design of composable architectures makes this self-healing approach possible by decoupling components, allowing services to operate independently. This means that services don't run within the same process but instead communicate through event-based messaging, allowing services to react quickly to changes or failures. When components use events to communicate, they can isolate issues to prevent them from cascading through the entire system, ensuring that one failure doesn't bring down multiple services.

Kubernetes stands out as a powerful tool for self-healing in cloud native environments. However, it relies heavily on correct configurations and policies. Misconfigurations, such as improper resource limits or faulty health checks, can cause cascading failures. Proper setup and regular audits are essential to avoid these risks. Kubernetes automatically monitors the health of containerized workloads and takes corrective actions when something goes wrong. For instance, if a pod (a group of containers) crashes or stops responding, Kubernetes will automatically restart it or reschedule it to a

different node with available resources. This ensures that critical data services remain highly available without requiring human intervention. For data and analytics platforms that depend on containerized data pipelines, this capability can significantly reduce downtime and keep data flowing smoothly.

For organizations that rely on infrastructure as code (IaC), tools like Terraform can complement Kubernetes by managing the provisioning and configuration of underlying cloud resources. Terraform tracks the desired state of infrastructure, such as storage or compute configurations, and adjusts resources as needed. When combined with Kubernetes orchestration, this approach creates a self-healing system that automatically adapts to changes, ensuring that data workloads run consistently even as conditions change.

AI-driven auto-scaling further enhances self-healing by leveraging ML models to analyze infrastructure performance and predict resource needs. For example, AI models can monitor data ingestion rates or query patterns to adjust compute resources dynamically, ensuring that processing power scales up during high demand and scales down when workloads decrease. This not only improves cost efficiency but also ensures that data processing pipelines remain responsive and efficient.

Serverless functions, such as those offered by AWS Lambda, also contribute to self-healing in event-driven architectures. They automatically retry failed events or redirect errors to a dead-letter queue for further analysis, allowing the system to recover gracefully from transient failures. For example, if a data transformation task fails due to a temporary API timeout, the serverless function can retry until the process succeeds, preventing data loss or incomplete results from impacting downstream analytics. However, serverless functions often have retry limits and can fail silently if not properly monitored. Implementing robust monitoring and alerting is crucial to ensure failures are detected and addressed.

These adaptive capabilities become even more critical as data and analytics systems scale, where managing infrastructure manually becomes impractical. By automating recovery and adapting to real-time conditions, organizations can maintain reliable data pipelines and analytics platforms without the need for constant human oversight.

In the next chapter, we'll explore how AI-driven intelligence can take these concepts further by streamlining the management of data workloads and operational pipelines, allowing for faster response times, greater scalability, and a focus on delivering insights and business value rather than managing infrastructure.

Ensuring Security, Governance, and Compliance

The need for robust security, governance, and compliance mechanisms to ensure data integrity, protect against breaches, and meet regulatory requirements has never been more critical.

In a traditional monolithic data architecture, security controls are often tightly integrated into a centralized system, making it challenging to adapt to new security requirements or regulations. In contrast, a composable data architecture breaks down the data platform into modular components, each of which can have specific security controls applied independently.

As your data platform will physically store your corporate data, other tools will bring the processing to the data. It's natural, then, that the primary focus from a security perspective will be on the data platform itself; therefore, you should expect that your data platform infrastructure provides multiple levels of security controls, including networking, user access, and encryption controls.

The composite architecture dictates you bring several tools into your architecture, which creates a number of integration points to consider. There are several key considerations when designing your architecture here. You want to confirm that the connectivity between the tools that connect to your data platform don't physically transfer any data, and you also want to ensure that key-pair authentication options exist for application-to-application authentication.

When it comes to user authentication, introducing an identity provider (IdP) into your data architecture to manage digital identities brings significant benefits. This approach creates a federated identity environment and separates user authentication from access controls, which your data platform takes care of. Having a dedicated IdP brings with it additional benefits such as a centralized place to take care of user credentials and allows for single sign-on (SSO) as

well as multi-factor authentication (MFA), creating a secure, robust architecture while simplifying the overall maintenance.

Using RBAC, the organization can apply different access policies to various data sources or regions. The great thing about following an RBAC approach is that users have to be part of a role, and the role is granted privileges to perform operations on specific datasets or objects, making it easier to control and manage who has access to what. For example, data analysts in Europe might have access to aggregated customer data but not to raw data containing personally identifiable information, whereas a different set of access rules could apply to analysts in other regions. If regulations change to require more stringent access controls, privileges attached to an existing role can be easily revoked, with the changes applied immediately.

Audit and Compliance

You need to ensure your data architecture has the ability to monitor and log data access to the data assets within your data platform. Additionally, other technologies that form your composable architecture will have their own native logging mechanisms. Therefore, it is worthwhile considering if you wish to consolidate logs from all these components into a centralized enterprise logging tool such as Splunk, Grafana, or Datadog for improved observability. However, this approach requires robust log normalization to ensure consistency across sources and adequate storage capacity to handle the volume of collected logs effectively. These tools fall under the category of security information and event management (SIEM) and can leverage AI to help identify and nullify threats before they impact business operations.

AI-assisted services can help with a number of aspects, including:

Adaptive thresholding
: This analyzes historical performance benchmarking data from within your own organization and responds to the demands of the workloads. Not only does this approach automatically adapt to seasonal patterns caused by changes in user demand, but it also learns from actual usage patterns to optimize resource allocation and reduce costs.

Monitoring and alerting
: These identify if the number of alerts is higher or lower than usual and where they are originating, helping direct teams to prioritize their resources more effectively.

Security by Design

The most important part is ensuring security considerations are integral to the initial design stages, rather than an afterthought. Engaging with your security and governance teams to gather their requirements is a crucial input and will help you as a data leader design a robust, flexible, and scalable architecture blueprint with secure guardrails woven into the fabric of the design.

These governance frameworks should also align with your business's strategic objectives, ensuring that data is used responsibly and ethically while supporting data-driven decision making.

Change Management

As a data leader, it's important to consider how changes to the architecture can be managed with an eye toward maintaining security and compliance, ensuring that updates or new components don't introduce vulnerabilities or compliance risks or result in operational outages. In this section we look at how this architecture can help support you and your teams to deliver high-quality code while minimizing the risks.

Version Control

Use versioning tools like Git to track changes and iterations effectively. By maintaining a clear version history, you can revert to previous model versions if issues arise or performance degrades. Moreover, documenting changes and model metadata aids in transparency and collaboration among data scientists and engineers.

Tools like GitHub were originally built to support a manual, human-centric coding workflow. Developers would write code, collaborate with team members, and track changes through branches, all while relying on GitHub's version control to keep a record of their contributions. This model is still relevant today, offering a structured way for teams to collaborate on codebases. However, the rise of AI coding assistants is transforming this traditional approach. With

AI now generating and modifying code, the dynamics of version control are fundamentally shifting.

While GitHub has adapted over the years to embrace automation tools like CI/CD—more of that in the next section—it still assumes that most of the work comes from human developers. The introduction of AI-driven development challenges this assumption. Managing the interaction between human-written and AI-generated code introduces new complexities. For instance, traditional version control systems like GitHub are not optimized to clearly differentiate between code changes made by a developer and those made by an AI tool. This can complicate debugging, auditing, or rolling back changes, as it becomes harder to identify the origin of a specific modification. In an AI-driven workflow, the lines between human and machine contributions blur, creating challenges for GitHub's existing model, which focuses on tracking individual user contributions.

Despite these challenges, there are significant opportunities for enhancing version control systems to better accommodate AI-driven workflows:

Seamless integration with AI tools
: Future evolutions of version control platforms could serve as a central hub for managing AI-driven development, streamlining processes like pushing and pulling code between systems. By integrating AI tools directly into these platforms, developers could manage the entire workflow in one place without needing to manually transfer code. This would significantly increase efficiency and reduce the friction associated with multisystem workflows.

AI-aware version control
: A more advanced version control system could identify whether changes were made by a human or an AI tool, making it easier to trace modifications and ensure accountability. The ability for an intelligent service to add comments to code changes within the version control repository is an accessible way to provide traceability, and it would be particularly useful for debugging or conducting security audits, providing a clear view of how a codebase has evolved over time.

Natural language change requests
: As developers increasingly use natural language prompts to guide AI in generating code, we expect to see features that translate natural language instructions into code changes. This would streamline the development process, especially when collaborating with nontechnical stakeholders who are more comfortable describing changes in plain language.

Component-based workflows
: In a composable architecture, there are a range of tools and code assets that make up the entire solution. Most, if not all, tools that make up a cloud-based composable solution support version control systems such as GitHub. The consolidation of code provided by a multitude of services centrally is of enormous benefit, allowing developers to manage and version individual components separately while maintaining the integrity of the overall system. This would enable teams to track the evolution of specific features and prevent changes to one component from inadvertently breaking the entire application.

AI-enhanced code reviews
: As AI tools increasingly contribute to the codebase, the review process must adapt. AI-enhanced code reviews could assist human reviewers by analyzing AI-generated code, identifying potential issues, and suggesting improvements. This would help maintain the quality of the codebase while reducing the burden on human reviewers, ensuring that even AI-generated contributions meet the organization's standards.

Continuous Integration/Continuous Deployment

To deliver data products at scale, it's simply not enough to rely on people to validate, test, and deploy pipelines. You must leverage automation and intelligence to do more with less. A repeatable and proven CI/CD pipeline allows teams to build and release more changes, in less time, with confidence that the codebase is continuously tested, validated, and safely deployed into production environments.

Without a CI/CD pipeline in place, even the smallest of changes may contain undetected errors that, when deployed into production, lead to inaccuracies in reports and the costly process of backing out changes. The impact of this is far-reaching; it can lead to data

consumers losing trust in the data and the processes that deliver the data.

The key to avoiding this is to provide your data engineers with early feedback on their code. When code changes are checked into your codebase within your version control system, CI/CD pipelines are automatically triggered. This workflow is able to identify the changes and understand the downstream dependencies that rely on these changes. This allows the pipeline to deploy those objects that have been impacted to a temporary environment and execute the data pipelines and associated tests before reporting back the results and cleaning up the environment.

Integrating AI-driven intelligence into the CI/CD process allows it to provide detailed insights to data engineers about changes, such as rows or columns that have been added, modified, or removed; changes in column values; and alterations to column data types or order. It can even indicate the percentage of rows that have changed, been added, or been removed relative to the entire data model. This level of detail enables engineers to precisely assess the potential impact of their changes on data models and reports before merging them into production. For example, if column values shift unexpectedly or nulls appear, engineers can identify and address these issues before the data becomes accessible to end users or downstream systems. This granular visibility ensures that data remains accurate and reliable, building confidence before it is exposed to broader audiences.

This advanced, intelligence-enabled CI functionality provides data professionals with the confidence that every code change not only builds successfully but also implements the precise changes intended for the business. These processes help you and your teams deliver at scale by streamlining the development process and eliminating the need for manual testing and reviewing row-level differences. Catching data issues as early as possible in the development lifecycle is far less expensive than having to fix them in production. This means your teams can focus on adding value to the data by building data products and generating new, tangible insights instead of focusing on debugging or resolving issues after deployment.

In the future, we expect to see these improvements becoming wider-reaching across components and services that make up the composable architecture, extending to include downstream dashboard

impact analysis, enabling users to better understand where data is used and what the potential impact is for data consumers. We also see a world with AI powering code reviews to enhance code and data development processes at scale, and data quality monitoring that alerts data engineers to issues before they affect end users, ensuring that they are always the first to know about potential data problems.

Key Takeaways for Data Leaders

As organizations navigate the complexities of modern data architecture, several critical considerations emerge for successful implementation:

Choose the right model for governance
: A CoE offers a balanced approach, ensuring consistency and high standards while allowing business units to innovate with autonomy.

Leverage AI for efficiency
: Integrating LLMs and AI-powered tools into your data platform can drastically reduce manual work, enabling teams to focus on delivering business value.

Optimize storage and compute costs
: Separating storage from compute gives you the flexibility to scale economically, using open table formats like Apache Iceberg to avoid vendor lock-in.

Build security into the architecture
: Adopt modular security controls that adapt as your platform grows, ensuring that compliance and governance remain strong as new regulations emerge.

Adapt version control for AI
: Recognize that AI-generated code changes the game for version control. Prepare to adopt AI-aware tools that can track and review these changes effectively, ensuring a smooth integration of human and AI contributions.

With the foundation of infrastructure and core capabilities in place, we now turn to Chapter 4, where we'll explore how these elements come to life when running data workloads. We'll turn our attention to data management, AI-powered orchestration, natural language

querying, and task-specific LLM models. We'll see how these capabilities support efficient data processing, ingestion, and analytics, helping organizations extract insights faster and unlock the potential of AI-driven data applications.

Summary

In this chapter, we delved into the core elements that make up a composable data architecture, focusing on the infrastructure, security, and governance frameworks that enable scalable, adaptive data platforms. We explored the choice between centralized and federated models, assessing the benefits and trade-offs of each. While federated models like data mesh offer flexibility, we advocate for a CoE approach. A CoE strikes a balance by combining centralized expertise and governance with the freedom for business units to innovate. This model ensures that data remains high quality, secure, and well governed while still empowering teams to self-serve as they mature.

We also examined how LLMs and AI-powered applications can transform composable architectures, turning them into smarter, more efficient platforms. We discussed the importance of high-performance ML processing power and cross-cloud solutions, which are critical for handling the scale and flexibility needed for advanced AI workloads.

We emphasized the benefits of separating storage and compute resources. This approach allows for scaling storage without the need to scale compute, providing both cost-efficiency and operational flexibility. We highlighted the role of open table formats like Apache Iceberg in enabling interoperability and avoiding vendor lock-in, offering data leaders a way to keep their data accessible and adaptable.

Security, governance, and compliance play a pivotal role in a composable architecture. We explored how RBAC, automated audit logging, and security by design ensure data remains protected while adapting to evolving regulatory landscapes. Modular security controls provide the flexibility needed to address new threats and compliance requirements, giving data leaders confidence in their ability to secure diverse environments.

Lastly, we covered version control in the context of AI-driven development, noting how traditional tools like GitHub must evolve to keep pace with AI-generated code and natural language change requests. AI-enhanced workflows bring new opportunities for collaboration and efficiency, but they also require a rethink of how we track changes and maintain accountability.

CHAPTER 4

Running Data Workloads Using Composable Data Architecture

Building high-performing, scalable data pipelines is essential for modern organizations looking to unlock the full potential of their data. However, achieving this at scale can quickly become a complex and time-consuming endeavor. Traditional data transformation approaches often struggle to meet the demands of today's fast-evolving landscape, where agility and intelligent capabilities are critical. To stay ahead of the competition, we must design composable, agentic architectures that incorporate advanced tools such as copilots and LLMs from the outset—paving the way for innovative use cases, including but not limited to secure AI applications.

As a data leader, balancing cutting-edge AI-driven services with efficient pipeline management can streamline efforts, reducing the time and resources needed to deliver impactful data products. Automating repetitive tasks, enhancing data quality, and fostering collaboration can democratize data insights and improve efficiency. In this chapter, we'll expand on these benefits by introducing four key capabilities into our composable architecture:

- AI-powered orchestration
- Task-specific LLMs
- Copilots
- AI data management

We'll explore how the combination of copilots, task-specific LLMs, AI-powered orchestration, and AI data management enhances each phase of the data workload lifecycle we introduced in Chapter 2, including design, runtime operations, operational troubleshooting, and self-optimization. Each capability supports the goal of creating a flexible, scalable data architecture that adapts to changing business demands. We'll guide you through how these intelligent tools improve data quality, streamline processes, and empower teams at every stage of the development lifecycle, from design to self-optimization. By the end of the chapter you'll understand how each set of capabilities work together within our model (Figure 4-1), which you'll recall from Chapter 3.

Armed with this knowledge, our hope is that the valuable information we share over the next few pages helps you achieve this vision, bringing your composable architecture to life with greater ease and efficiency.

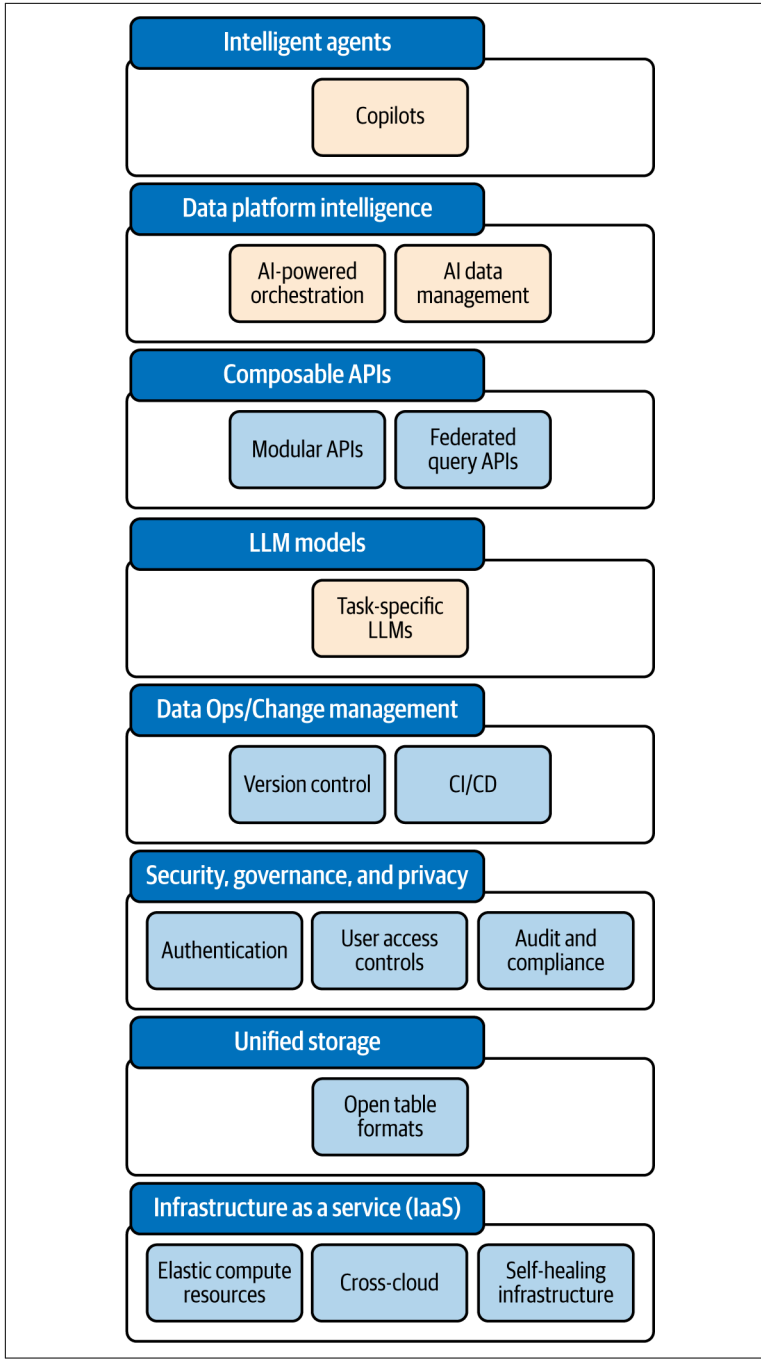


Figure 4-1. Capability model with data workload capabilities

Implementing Data Pipelines and Workflows

Data pipelines form the backbone of your data workloads, driving the movement and transformation of data from source to consumption. They support the design phase of the data lifecycle by creating a foundation for secure, reliable, and scalable data handling. With intelligent capabilities, the design and implementation of data pipelines can become more agile, adaptive, and efficient.

In a composable architecture, adding intelligent capabilities like task-specific LLMs and copilots enhances this foundational work, making pipeline design and implementation more agile, adaptive, and efficient. These tools automate repetitive tasks, enforce quality checks, and generate code, allowing data engineers to focus on high-value activities and speeding up the development process. Task-specific LLMs bring specialized expertise to distinct data processing functions, such as data cleaning, entity recognition, or data transformation, while copilots assist by managing metadata, creating efficient workflows, and ensuring alignment with business logic.

AI-Powered Orchestration

AI-powered orchestration dynamically manages task sequencing, resource allocation, and dependencies across data pipelines. This capability ensures efficient operations, adapts to workload demands in real time, and reduces the need for manual intervention. By integrating orchestration tools into composable architectures, organizations can streamline the runtime phase, enhance operational troubleshooting, and optimize analytics workflows, driving efficiency and scalability across the data lifecycle.

Task-Specific LLMs

General LLMs bring high efficiency, but it's a misconception to think they always have the right answers. Unfortunately, they are prone to providing incorrect responses—known as "hallucinations"—which are more likely to occur in scenarios involving open-ended questions, poorly defined prompts, or when the model lacks sufficient context or domain-specific knowledge. These hallucinations can occur anywhere from 1% to nearly 30% of the time (Figure 4-2). This creates a risk of misleading insights or flawed decisions if these inaccuracies go uncorrected. The trade-off, as some argue, is that AI models would lose their "magic" if they

simply withheld responses when uncertain. However, as LLMs become more widely adopted, we expect that organizations will focus increasingly on reducing hallucinations, with the ultimate objective of achieving fully accurate and reliable AI models.

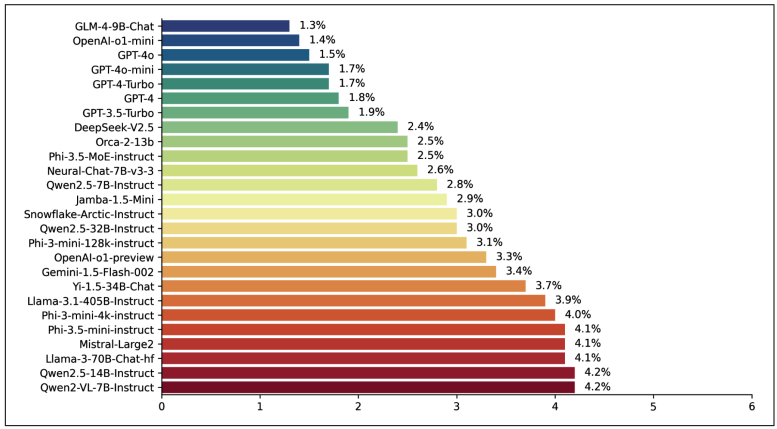


Figure 4-2. Hallucination rate for the top 25% LLMs

We must acknowledge this trade-off and determine the best way to manage these risks. The use case plays a significant role; in some scenarios, accepting this trade-off may be reasonable, but in others, such as analyzing financial data for strategic decisions, we may require complete accuracy. The challenge lies in not knowing if your answer falls within the 5% margin for hallucinations.

There are ways to minimize these risks. A strong starting point is prompt engineering—using more precise and context-rich prompts to improve response accuracy. The more context we provide, the better the response will be. Additionally, setting guardrails that guide the AI model's output can help restrict responses and guard against biases.

Enterprises can use foundational models like OpenAI's GPT-4 to explore GenAI applications, but these models have limitations. Privacy and security of corporate data, transparency of training data, fine-tuning for specific requirements, task-specific accuracy, and cost-value considerations all pose challenges.

Task-specific LLMs offer targeted capabilities but come with limitations, including susceptibility to hallucinations, limited adaptability outside their training data, and the cost and complexity of fine-tuning models for highly niche tasks. These constraints under-

line the need for careful evaluation of use cases and consistent monitoring to ensure model accuracy.

In Chapter 1, we touched on the differences between RAG and fine-tuning. While prompt engineering and RAG are powerful techniques to enhance LLM performance, they have their limitations. Prompt engineering relies heavily on the user's ability to craft precise, context-rich instructions, which may not always be feasible for complex or nuanced tasks. Poorly crafted prompts can lead to ambiguity or incomplete outputs, reducing the reliability of results.

Similarly, while RAG can significantly improve accuracy by grounding responses in relevant external documents, it may fall short when the required knowledge is not present in the provided inputs, or when real-time retrieval fails to supply sufficient or high-quality context. Additionally, RAG implementations can be computationally expensive, introducing latency that may hinder performance in time-sensitive scenarios.

These limitations highlight the need for alternatives like fine-tuning, where a model's behavior can be adjusted to specific requirements using curated datasets, allowing it to deliver better performance for domain-specific tasks.

This means we can take general-purpose LLMs as a starting point before tailoring them into task-specific models that specialize in various data processes. These task-specific models, which sit within our capability model, focus on particular functions, such as data cleaning, entity recognition, and natural language querying, enabling them to deliver higher accuracy and reliability in these areas.

Fine-tuning for individual tasks creates a library of specialized LLMs, enabling engineers to "plug and play" these models for rapid pipeline building and extension. We also anticipate data marketplaces incorporating custom LLMs to support specific activities for common datasets extracted from SaaS platforms like Salesforce, HubSpot, or Google Analytics.

Integrating task-specific LLMs into data pipelines allows for automation in key areas—data ingestion, transformation, validation, and quality checks—enabling more efficient workflows. In the following sections, we'll explore how task-specific LLMs interact with each lifecycle phase of data workloads.

The Role of Copilots in Workflow Assembly

Within the intelligent agents section of our capability model, we introduce copilots. Copilots play a transformative role in automating the creation of data pipelines by intelligently inferring the necessary transformations based on the data sources and specific business requirements. They take on the heavy lifting of repetitive tasks, allowing data engineers to refocus their efforts on strategic, high-impact projects. This automation not only enhances productivity but also accelerates the delivery of data products. The ability to create data workflows swiftly means that teams can react to new business demands with agility, reducing the time between data sourcing and actionable insights.

By offloading routine tasks, copilots create an environment where data engineers can prioritize their expertise on complex, value-driven initiatives. This shift in focus ultimately drives a more innovative and responsive data engineering practice, enhancing the overall output and strategic capacity of the team.

While copilots significantly enhance productivity, there are potential risks, such as overreliance on automated workflows, misinterpretation of complex requirements, and challenges in debugging errors from AI-driven decisions. To mitigate these risks, organizations should implement robust testing frameworks and maintain human oversight during critical phases.

Now we've introduced two new core capabilities—copilots and task-specific LLMs. Next, we'll guide you through the specific benefits they bring through each stage of the four-phase development lifecycle. We'll provide practical insights on deploying copilots effectively, equipping your organization to harness the power of AI in composable data architectures.

Data Pipelines—the Engine Room for Your Data Management

When it comes to delivering clean, accurate, timely data to your consumers, data pipelines are the engine room driving your data workloads. Embedding business logic directly into your pipelines creates a solid foundation to activate a range of use cases and ensures consistent data delivery at scale.

Decomposing your architecture into its component parts—as we've been exploring throughout this guide—brings significant benefits. However, there's a tipping point where over-decomposition can introduce unnecessary complexity and inefficiencies. For data pipelines, this often appears when organizations split their ELT processes into separate EL and T functions with a dedicated tool for each phase: a data loader for the extract and load phase, another tool for transformation, and a reverse ETL tool to "push" data to downstream systems. While each tool in this approach may perform its specific function well, the lack of cohesive oversight can lead to a large volume of noncritical data being extracted from source systems and loaded into the data platform without assessing business value. This approach carries a risk: your data platform can quickly become flooded with data. Although the initial cost of building pipelines and bringing this data in may be low, the ongoing operational overhead to manage, update, and govern this data can grow quickly.

This is especially challenging when data teams don't have a clear understanding of the benefits each dataset brings to the organization. For developer-centric teams or small, straightforward projects, such separation may make sense, as it allows flexibility and modularity to move quickly without significant complexity. However, larger organizations or projects with multiple dependencies risk introducing bottlenecks and redundancy, particularly in scenarios where the data may not add strategic value.

In traditional on-premises ETL setups, data teams were forced to think about business outcomes proactively to justify the cost and effort. In modern, highly decomposed pipelines, this focus can be diluted as multiple tools add costs and require deeper expertise to manage efficiently. A well-balanced approach, leveraging unified tools when possible, can prevent the pitfalls of over-decomposition, keeping data operations lean, aligned with business objectives, and responsive to organizational needs.

Lifecycle Phases of Data Pipelines

To provide structure to the complex process of running data workloads, we break down the data pipeline lifecycle into our four key phases we introduced in Chapter 2: design, runtime operations, operational troubleshooting, and self-optimization (Figure 4-3).

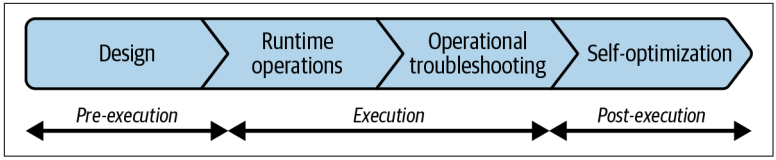


Figure 4-3. The four-phase development lifecycle

Each phase leverages AI-driven tools, such as copilots and task-specific LLMs, to enhance efficiency, data quality, and resilience. In this section, we explore how these intelligent tools optimize each phase, creating a cohesive and adaptable system that aligns with the composable architecture approach.

Design phase

The design phase is foundational to the data pipeline lifecycle and demands extensive planning, architecture, and testing to ensure pipelines meet organizational standards. During this phase, copilots play a crucial role in reducing manual effort by automating repetitive tasks like schema generation, data type suggestions, and metadata tagging. These copilots ensure alignment with business requirements by turning them into code, helping to build a consistent and high-quality data architecture that aligns with strategic goals. Additionally, copilots streamline data exploration, transformations, and testing, creating a strong basis for the next stages.

Runtime operations phase

In the runtime phase, pipelines rely on efficient management to maintain smooth operations. AI-powered tools dynamically handle dependencies and resources, automating operations to improve efficiency and reduce manual oversight (see "AI-Powered Orchestration" on page 80).

Operational troubleshooting phase

During operational troubleshooting, intelligent agents monitor pipeline health in real time, detect anomalies, and suggest corrective actions. This proactive approach minimizes downtime and ensures reliable, high-performance workflows (see "AI-Powered Orchestration" on page 80).

Self-optimization phase

In the self-optimization phase, pipelines leverage AI-driven tools to continuously refine their own performance. By analyzing historical data flows and identifying patterns, copilots can adjust configurations, reallocate resources, and fine-tune settings to avoid bottlenecks and maximize efficiency. This ongoing optimization process ensures that data pipelines remain agile and resilient, adapting to evolving business needs and delivering dependable data insights in real time.

Self-optimizing systems require extensive historical data, reliable monitoring frameworks, and robust infrastructure to function effectively. Organizations without these prerequisites may face delays or inaccuracies during optimization processes. Furthermore, ensuring compatibility with existing tools and workflows can pose a significant hurdle.

With a solid understanding of how these capabilities support each phase of the development lifecycle, we're ready to explore the specifics of running data workloads within a composable architecture. In the following sections, we'll dive deeper into data ingestion and integration, data processing and transformation, and running analytics and BI applications, examining how these stages can be streamlined and optimized by leveraging intelligence to meet the demands of modern data environments.

Data Ingestion and Integration

In data ingestion and integration, task-specific LLMs play a crucial role by aiding in both data exploration and data transformation. These models support the ingestion phase by enabling engineers to explore data quality and consistency quickly while also assisting in generating transformations based on business rules. With LLMs, data engineers can build comprehensive pipelines from raw data sources to refined, analytics-ready datasets, streamlining the integration process.

Tools such as Fivetran and Matillion play critical roles in the ingestion and transformation process (Figure 4-4). AI capabilities augment these tools by automating repetitive tasks, enhancing metadata generation, and improving data quality without replacing established frameworks. For example, copilots can generate SQL

scripts that complement Matillion workflows, reducing development time.

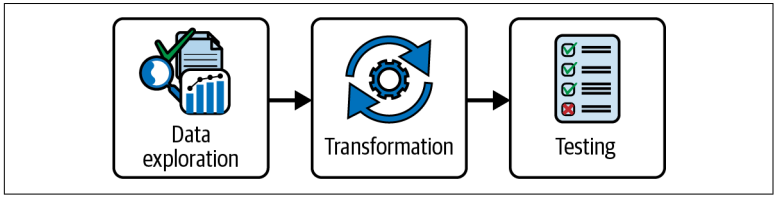


Figure 4-4. The three phases in data ingestion and integration

Data Exploration

Before pipelines are built, understanding the quality and consistency of incoming data is crucial, especially when working with new sources. Copilots can significantly speed up this stage, identifying missing values, data distributions, or metadata documentation by answering natural language questions, such as:

- Is there metadata available that documents the purpose, source, and data types of each column?
- What are the minimum, maximum, and average values for key numerical columns?
- Are there duplicate records in the dataset, and what percentage do they represent?
- Which columns have the highest percentage of missing data?

With a copilot, data engineers can quickly explore and analyze data without needing to learn complex schemas or write multiple queries. Using task-specific LLMs, copilots can even generate SQL code in response to questions, saving time and enabling rapid insights. This stage sets the groundwork for reliable data transformation and ensures that only high-quality data flows through the architecture.

Transformations

Copilots and task-specific LLMs each bring unique benefits to the transformation phase. While copilots automate repetitive tasks, manage metadata, and align transformations with business rules, task-specific LLMs bring advanced capabilities like language translation, anomaly detection, and entity recognition, especially valuable when handling unstructured data. Together, these tools accelerate

transformation processes, ensuring pipelines are both efficient and aligned with organizational goals.

By transforming business requirements into code, copilots ensure pipelines are aligned with expected outputs. They also optimize downstream data storage by selecting appropriate materialization methods (e.g., tables, views, or materialized views) based on usage patterns, minimizing latency and reducing storage costs. As the solution scales, copilots dynamically adapt pipeline configurations to align with evolving demands, ensuring optimal performance over time.

Testing and Validation

Task-specific LLMs enhance the testing and validation process by generating test scripts and scenarios that replicate real-world conditions, catching potential errors before deployment. This approach ensures that pipelines meet quality standards and helps data teams confidently move workloads into production.

Copilots automate the generation of test scripts to simulate common pipeline challenges, such as intermittent API responses or delayed data arrivals. They continuously monitor pipeline behavior, logging inputs and outputs and comparing them to expected results. If any deviations or anomalies are detected, copilots alert data engineers, allowing for timely intervention.

Data Processing and Transformation

Modern data platforms require sophisticated processing capabilities to handle the increasing volume and complexity of data transformations. These capabilities are being enhanced by AI-powered tools that streamline workflows and improve data quality.

Automated Metadata Management

As workflows are built, copilots capture and enrich metadata, creating lineage documentation and glossaries that adapt as pipelines evolve. This automated documentation maintains data transparency and reliability, providing stakeholders with accessible information on data origins and transformations.

Adaptive Intelligence and Growth

Unlike traditional tools, copilots bring an adaptive, self-learning edge to data engineering. They evolve continuously, gaining knowledge from past interactions and transformation efforts to improve their capabilities. This learning isn't generic—it becomes tailored to individual users over time. Every data engineer approaches their work differently, bringing unique strengths, preferences, and experiences to their development activities. Copilots, through consistent interaction, discern these individual characteristics and adapt their support accordingly.

For instance, an engineer adept at building complex data models might receive proactive assistance in data validation or schema design from their copilot. As interactions grow, the copilot learns where the engineer's strengths shine and where additional support is needed. This adaptability allows copilots to offer targeted, proactive recommendations and support, making data pipeline generation more precise and efficient. They cater not just to the broader needs of an organization but also to the nuanced working styles of individual team members, evolving alongside them to create an optimized development experience.

Uplift in Data Quality

One of the key advantages of incorporating copilots is their capability to proactively identify and address data quality issues within pipelines. This ensures that data reaching end users is more consistent and reliable. By automating checks and corrections, copilots contribute to a significant reduction in data latency and promote greater confidence in data outputs.

Higher data quality means fewer errors downstream and a smoother experience for data consumers. Engineers can trust that their copilots are safeguarding the integrity of their work, allowing them to focus on creating value rather than constantly troubleshooting data issues.

Running Analytics and BI Applications

Once deployed, pipelines rely on AI-driven tools to manage dependencies, task sequencing, and execution. Copilots streamline

workflows, optimizing resources and enhancing efficiency without requiring manual intervention.

Task-specific LLMs further enrich this phase by processing unstructured data (e.g., PDFs, images) and integrating insights into target datasets. For example, they could assess component failures or update logistic records in real time. Copilots also streamline task-based pipelines for specific workloads—such as model training or application deployment—ensuring optimal resource allocation within the architecture.

Data observability is essential for maintaining pipeline performance and health. Copilots enhance observability by visually tracking key performance metrics and allowing engineers to set alerts for potential issues. In an advanced setup, copilots can perform multistep reasoning to identify root causes of anomalies and recommend or initiate corrective actions.

Beyond just monitoring, copilots continually optimize pipeline configurations by analyzing workload patterns, balancing resource usage, and adjusting settings as data volumes fluctuate. However, this capability is not without challenges. Copilots must weigh trade-offs between computing costs, performance, and reliability, often relying on preset thresholds, predictive models, and historical data patterns to make decisions. For instance, they might prioritize cost savings during low-demand periods while allocating additional resources during peak loads to maintain performance. While effective, these decisions require careful calibration to avoid overprovisioning, underperformance, or escalating costs. Incorporating human oversight in defining priorities and validating adjustments can help mitigate these risks and ensure alignment with organizational goals.

Future-Ready Pipelines

As data environments evolve, copilots facilitate a dynamic, future-ready approach by proactively managing infrastructure adjustments based on workload demands. Copilots support composable and agentic solutions that can automatically scale resources and optimize configurations, ensuring that pipelines can handle increasing data volumes and complexity without requiring manual intervention. This capability keeps data solutions flexible and responsive, prepared to meet future business needs effectively.

Empowering Teams to Bridge Skill Gaps

With the support of copilots, data engineers can increase their output and take on more complex projects without needing to scale the team. Routine, repetitive tasks that would traditionally consume valuable time are automated, freeing engineers to focus on higher-order work. This also allows senior engineers to guide less experienced team members, effectively bridging skill gaps within the team.

This empowerment reduces the pressure on talent resources by enabling the existing team to be more productive and innovative. Copilots act as an assistant that helps guide best practices and shares knowledge, ultimately raising the overall skill level and cohesion of the engineering team.

Expanding Data Access Across Teams

One of the most transformative effects of copilots is their ability to lower the barrier to entry for data transformation. Their intuitive design makes sophisticated data workflows accessible beyond the realm of seasoned data engineers. This democratization of data capabilities empowers a broader range of users within an organization, including analysts and other nontechnical stakeholders, to build and interact with data pipelines effectively.

While this expanded access might seem to introduce risks, such as poorly designed transformations, lack of governance, or security vulnerabilities, these concerns are mitigated by the built-in safeguards and guided workflows provided by copilots. For instance, copilots enforce organizational best practices and governance policies through automated quality checks, RBAC, and standardized templates. They also validate transformations before deployment, ensuring compliance with security and data governance frameworks. Furthermore, for critical changes, particularly for promoting code from nonproduction to production environments, it should be considered practical for a human approval process to be operationally embedded to review and approve the changes. This additional layer of oversight ensures that transformations are well designed, secure, and aligned with business objectives.

Simplified, Outcome-Focused Design

The introduction of copilots shifts the focus from specifying each development step to concentrating on desired outcomes. This outcome-oriented approach simplifies the data pipeline design process and results in code that is more maintainable and reusable. Instead of engineers spending time detailing the minutiae of transformations and data flows, copilots streamline these elements, allowing them to concentrate on the end goal.

The resulting pipelines are easier to manage and update, removing the traditional complexities associated with modifying data workflows. This simplicity translates to reduced maintenance overhead, making it easier to onboard new team members and iterate on existing processes without introducing significant risk or delays.

AI Data Management

In the previous chapter, we explored ways to secure your infrastructure while addressing governance elements essential for auditability and compliance. Now, as we shift our focus to building and implementing data pipelines, we'll look at how to manage data effectively. Our goal is to ensure these pipelines deliver clean, standardized, and consistent data in a timely manner, which leads us to AI data management, the next capability in our model.

It's equally important that all data, including transformations, is easily traceable—both from a technical standpoint, to manage changes and pinpoint the root causes of issues, and from a business perspective, to provide a glossary of terms along with the data's origins. This dual approach promotes trust in the data among consumers by ensuring both transparency and accessibility.

The implications of adopting such intelligent systems, as you can see here, are substantial:

Scalable development capacity
: Organizations can expand their development capabilities by integrating virtual assistants into their workflows.

Continuous, around-the-clock operation
: Pipelines can advance without pause, maximizing productivity outside of standard working hours.

Uniformity and best practice adherence
: Virtual assistants ensure processes consistently follow established standards and best practices.

Enhanced knowledge retention
: Valuable organizational knowledge is embedded in metadata, making it easily accessible for future use.

However, these advancements also introduce critical considerations:

Workforce implications and economic shift
: The integration of virtual assistants could reshape traditional development roles, with potential impacts on employment.

Ethical dimensions of AI in the workplace
: As AI takes on more tasks, questions about its roles and responsibilities within the organization become crucial.

Maintaining quality and alignment with goals
: Ensuring AI-driven processes remain aligned with business objectives, ethical standards, and values is essential.

Limitations in creative problem-solving
: It's important to recognize the boundaries of AI's capability to innovate or address unique challenges.

Safety and risk management in autonomous systems
: Autonomous AI systems introduce unknowns that require careful oversight and safety protocols.

The following components—metadata-driven organization, data quality and consistency, data lineage, and data governance and policy enforcement—work together to build trust and maintain regulatory compliance across data pipelines. Intelligent tools like copilots automate metadata capture, enforce policies, and monitor data quality, ensuring transparency and alignment with business objectives in a composable architecture.

However, as you read through the sections that follow, it should be noted that implementing AI-powered data management tools often faces blockers such as data silos, lack of readiness in legacy systems, and resistance to change within organizational cultures. Training and deploying AI systems can be resource-intensive, with delayed returns on investment due to the time required to refine models and processes.

Metadata-Driven Organization

Keeping data lineage, glossaries, and documentation up to date can quickly become overwhelming. Traditionally, teams might dedicate significant resources to ensure these elements remain accurate, but even then, documentation is only as good as its currency.

A composable architecture encourages automation and a flexible, tool-agnostic approach to managing metadata, reducing dependency on any single platform. Adopting a metadata-driven approach alleviates the manual burden by enabling automated updates to lineage diagrams and documentation as the codebase evolves. Intelligent copilots and LLMs can harvest metadata from across your architecture, ensuring that your documentation remains accurate, current, and accessible to users.

With this approach, metadata can also be leveraged to enforce privacy rules and security policies dynamically, adapting to changes in regulations or organizational requirements. Intelligent tools can detect when changes in the data flow affect compliance and automatically adjust privacy settings, driving efficiency and compliance in real time.

Organizations with a multiplatform environment increasingly need flexibility across data platforms and compute engines. By centralizing metadata within this architecture, teams can leverage intelligent copilots to automatically update lineage diagrams, documentation, and privacy settings as data moves and changes across platforms.

A metadata-driven approach ensures data is easily traceable, with up-to-date lineage information available to both technical and business stakeholders. This level of transparency builds a foundation of trust in the data, as users can confidently rely on the lineage and provenance of information, regardless of which platform is hosting the data at any given time. Additionally, metadata can be leveraged to drive privacy rules and regulatory compliance dynamically, ensuring that the architecture remains adaptable to both business needs and regulatory requirements.

Data Quality and Consistency

Ensuring data quality and consistency becomes increasingly complex as more users across an organization access and interact with data. With the rise of the new age of AI, business users are closer

than ever to analytics workflows, bringing essential business context that complements the technical expertise of data teams. This collaboration is essential for creating accurate and relevant data products, but it requires rigorous data quality measures to prevent errors and misinterpretations.

Intelligent copilots embedded within our data pipelines can continuously monitor data quality, flagging anomalies, inconsistencies, or incomplete data. By automating these checks, copilots ensure that data products meet quality standards before they reach end users, allowing data teams and business users alike to have greater confidence in the insights generated. Furthermore, copilots powered by LLMs can interpret and apply data rules in natural language, bridging the gap between technical and business perspectives on data quality.

Data Lineage

As data workloads span multiple platforms, tracking data lineage across the entire architecture is essential for transparency and impact analysis. Copilots enable automated lineage tracking, creating dynamic visualizations that update as data flows through different transformations and storage locations. This automation empowers both technical teams and business users to trace data back to its origin, understand transformations, and troubleshoot issues more efficiently.

By maintaining a comprehensive and up-to-date view of data lineage, copilots support organizational accountability and build trust in data workflows. When a business user spots a discrepancy in a dashboard, for example, they can use the lineage map to trace data paths and identify potential sources of error. This capability strengthens the connection between data teams and business users, fostering a culture of transparency and shared ownership.

Data Governance and Policy Enforcement

With data spread across multiple platforms and regions, enforcing data governance policies consistently can be challenging. A composable architecture enables organizations to configure region-specific compliance modules, ensuring that data remains secure and meets regulatory requirements like GDPR or CCPA. Copilots leverage

metadata to automatically apply these policies, dynamically adjusting based on regional regulations or evolving business needs.

Examples of tools like Splunk, Grafana, and Azure for monitoring, or Snowflake and Databricks for data platforms, illustrate how AI systems can integrate seamlessly with existing governance mechanisms. AI tools rely on interoperability with these platforms to dynamically apply governance policies without disrupting workflows.

LLM-powered copilots can interpret policy changes as they arise and seamlessly integrate these updates into data workflows. This capability not only keeps data compliant but also minimizes manual interventions, allowing data teams to focus on strategic initiatives rather than repetitive compliance tasks. By aligning data governance with platform flexibility, copilots ensure that all data workflows remain secure, transparent, and compliant.

However, relying on AI for governance introduces trade-offs. AI models may lack contextual understanding, leading to oversights in nuanced regulations or business-specific exceptions. Misconfigurations or inadequate training can propagate errors at scale, while automation might reduce human oversight, increasing the risk of undetected biases or gaps in compliance.

To address these risks, organizations should implement regular audits, maintain human-in-the-loop mechanisms for critical decisions, and ensure robust training and monitoring of AI models. Explainability in AI outputs can also enhance trust and accountability, helping to balance efficiency with reliable governance.

Key Takeaways for Data Leaders

As organizations adopt AI-powered tools in their data architecture, leaders should focus on several strategic priorities:

Integrate intelligent copilots early
: Incorporate copilots and task-specific LLMs from the design phase to automate repetitive tasks, ensure alignment with strategic goals, and create a high-quality, consistent data architecture.

Optimize runtime with AI orchestration
: AI-driven orchestration dynamically manages dependencies and sequences tasks, enabling data pipelines to run efficiently

and respond to demand fluctuations without requiring manual oversight.

Prioritize proactive troubleshooting
: Intelligent monitoring tools enhance observability and perform root-cause analysis to address issues promptly, minimizing downtime and maintaining pipeline health.

Embrace self-optimization for scalability
: Leveraging self-optimizing capabilities within AI tools allows data pipelines to continuously refine performance, ensuring adaptability and resilience as business requirements evolve.

Focus on data quality and metadata management
: Automated metadata enrichment and continuous quality checks build trust in data outputs, supporting transparency and creating a foundation for reliable data-driven decision making across the organization.

Summary

In this chapter, we explored how composable data architectures can enhance the efficiency, scalability, and resilience of data pipelines by leveraging AI-driven tools such as copilots, task-specific LLMs, AI-powered orchestration, and intelligent data management. We delved into the four primary lifecycle phases of data pipelines—design, runtime operations, operational troubleshooting, and self-optimization—demonstrating how intelligent tools add value at each stage.

These phases, when supported by advanced AI capabilities, enable data leaders to build agile, adaptable data platforms that not only streamline data workflows but also drive greater business value. By automating repetitive tasks, improving data quality, and enabling real-time optimization, data teams can focus on strategic initiatives, fostering innovation and empowering broader data access across the organization.

In Chapter 5, we'll see how composable architectures can support advanced AI workloads and applications. We'll explore strategies for enabling data science, operationalizing ML models, and leveraging AI to deliver business-ready solutions at scale. This journey will highlight best practices for integrating AI workflows into your data platform to fuel transformative capabilities across your organization.

CHAPTER 5
Accelerating AI Initiatives

The bar seems to be constantly rising in terms of what data consumers expect technology can accomplish, which, in turn, means we need to understand how to accelerate AI initiatives across a broad set of use cases that drive the entire business.

Earlier in this guide, we emphasized the need to shift focus toward business value creation. AI is now seen as a key revenue and innovation engine. The use of data must be less about looking behind us and more about charting a path toward future revenue growth. Companies that cannot deliver AI/ML or data science models rapidly face existential competitive threats.

As a data leader, you want to be able to empower your teams to ship data products and services faster using AI. Providing a world-class composable architecture is one piece of the puzzle that can enable you to accelerate and support your customers' journey from ideation to production in the shortest time possible. Most chief data officers (CDOs)/chief data and analytics officers (CDAOs) currently lack the ability to fully measure the impact an AI/ML project has on the bottom line, leaving them to make critical business decisions without a clear understanding of their return on AI investments.

Another challenge is balancing the expectations of the board and leadership, who are looking for transformative results, with the reality that many data leaders lack the data science resources needed for large-scale AI initiatives. It's essential to ensure that AI systems are woven into the operational heart of the organization, directly contributing to business value.

While traditional AI was once confined to the hands of dedicated data scientists, it has now become democratized across the organization. Today, a wide range of teams—including data engineers, audit and compliance teams, MLOps engineers, and privacy specialists—need access to AI capabilities. This shift requires AI governance frameworks and responsible use policies to guide a diverse set of personas and functions, ensuring secure, ethical, and effective use of AI.

This chapter explores how to move from AI ideation to production, with a strong focus on fostering high-quality data ecosystems, AI governance, and intelligent automation. We'll discuss practical ways to empower teams to produce data products and services faster, leveraging intelligent agents and streamlined workflows. This shift not only accelerates delivery but also enhances transparency, security, and operational efficiency within AI initiatives.

In the following pages, we introduce the final three capabilities that complete our composable data architecture model: AI data governance, AI operations, and AI agents. Through each section, you'll discover how these elements work together to build a dynamic, scalable, and responsible AI environment.

As you progress, you'll gain a strategic overview of these tools and frameworks, understanding how each can integrate into your existing architecture to elevate business value and deliver a real competitive edge.

Enabling Data Science and Machine Learning

Today's market demands personalized user and customer experiences, real-time insights, and the ability to act on them, all of which hinge on the seamless integration of DSML into a business's core operations.

In this section we explore the essential building blocks that enable DSML to become a strategic advantage. By understanding and leveraging key capabilities such as intelligent AI agents and general-purpose LLMs, you can empower your data science teams to scale quickly, adapt to changing demands, and deliver impactful insights.

We'll also examine those ingredients that facilitate high-speed model training and testing while ensuring cost-efficiency, scalability, and flexibility, exploring how LLMs and task-specific models can be

fine-tuned to support specialized workflows and streamline complex ML processes. First, let's look into the role of AI agents, which allow data teams to automate repetitive, labor-intensive tasks and push their focus toward innovation and strategy.

AI Agents

In the previous chapter, we explored how copilot capabilities can take various inputs and use task-specific LLMs to increase productivity by handling repeatable, labor-intensive tasks involved in creating and running data pipelines. With natural language interaction, copilots make it easy for humans to engage in a conversational, intuitive way. But can intelligent agents go beyond just responding to questions, searching metadata, generating code, and extracting data from unstructured sources like PDFs and images?

Absolutely. This is where AI agents step in, bringing a unique ability to make rational decisions based on perceptions and data about their environment.

We're now seeing AI agents tackle complex tasks once thought too nuanced and costly for automation. Strategic areas such as customer behavior analysis, operational risk modeling, and performance optimization—domains traditionally managed by expert analysts—are now benefiting from AI-driven insights. By processing vast datasets and spotting patterns a human might miss, AI agents can deliver recommendations and insights that often match or even surpass human analysis.

Imagine an AI agent that can simulate multiple operational scenarios, assess emerging risks, or dynamically optimize resource allocation in real time. By adjusting models and decisions based on incoming data, AI agents empower organizations to make faster, more precise decisions. Data-driven precision and efficiency give companies a significant advantage, especially as they work to stay agile in today's competitive landscape.

Implementing AI Workflows and Applications

To fully unlock the potential of AI, organizations must move beyond isolated use cases and establish streamlined workflows that drive consistent, scalable insights across the business. Implementing robust AI workflows enables teams to connect various AI models,

tools, and data sources, creating a seamless pipeline from data ingestion to actionable outcomes.

By leveraging the composable architecture principles we introduced in Chapter 2, we can integrate flexible and adaptive AI applications that evolve with business needs, rapidly deploy solutions, and deliver tangible impact. This section explores how to build AI workflows that not only optimize current operations but also pave the way for future innovations.

So how do LLMs and AI agents work together? Imagine you're a retailer looking to send a highly personalized email to each customer, featuring product recommendations that align perfectly with their individual tastes. The goal is to increase the likelihood of conversion by tailoring the email content to each recipient's recent activity and preferences.

Here's how this process could work, using the three-step framework shown in Figure 5-1.

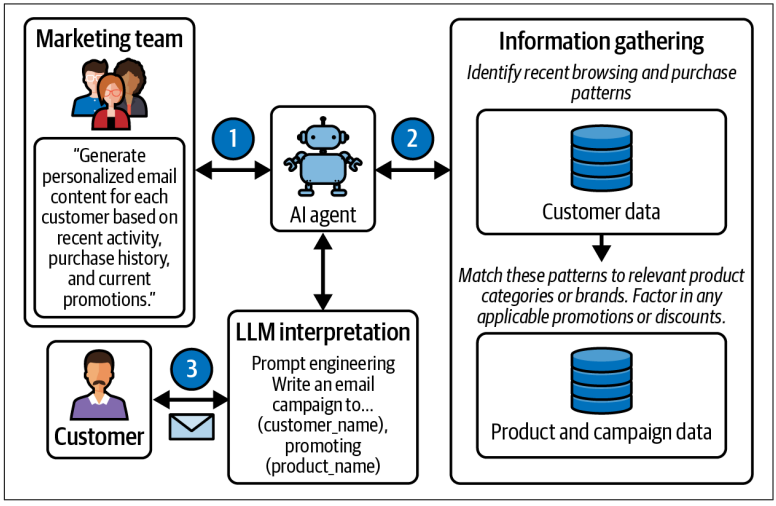


Figure 5-1. How AI agents and LLMs work together

Step 1: Receive the Instruction

The marketing team defines a broad instruction for the AI: "Generate personalized email content for each customer based on recent activity, purchase history, and current promotions."

The AI agent receives this instruction and begins breaking it down. For each customer, it will:

- Identify recent browsing and purchase patterns.
- Match these patterns to relevant product categories or brands.
- Factor in any applicable promotions or discounts.

Step 2: Gather the Information

The AI agent then gathers all relevant data for each customer to craft a targeted email:

Customer profile data
: Retrieves information like purchase history, browsing behavior, preferred brands, and size or style preferences. Additionally, includes demographic details such as age, gender, location, and income bracket to further tailor offers and messaging for maximum relevance and engagement.

Product data
: Pulls details on current stock, trending items, and available discounts to recommend in-stock items that align with the customer's profile.

Campaign metadata
: Accesses current promotions, discount codes, and special offers that may be relevant to each customer's interests.

Loyalty program data
: Integrates points or rewards the customer has accumulated, enabling tailored loyalty-specific promotions, such as "Redeem your points for an extra discount" or "You're just 50 points away from your next reward."

Additionally, the AI agent might cross-reference data from:

Weather data
: For local seasonal suggestions, like promoting coats to customers in colder regions.

Event data
: If the customer has searched for holiday-related items, the agent may prioritize gifting items or seasonal products.

Step 3: Execute the Actions

With all the necessary data gathered, the AI agent generates a tailored email for each customer:

Subject line
: Uses a catchy, personalized subject like "Hi [Customer Name], check out new arrivals in your favorite brands!"

Product recommendations
: Selects specific items based on browsing history, recent purchases, and trending categories. For instance, if the customer recently browsed winter jackets, the email might feature the latest coat collection, along with complementary accessories.

Special offers
: Applies relevant discount codes or promotions, and highlights these in the email, such as "20% off selected outerwear!"

Engaging content
: Adds micropersonalized messaging that feels conversational and relevant; for example, "We noticed you've been exploring cozy winter wear. Here are some handpicked favorites just for you!"

The AI agent ensures that each email is formatted consistently and inserts clear calls to action, such as "Shop Now" or "Explore New Arrivals." If the customer opens the email and clicks on a recommendation, the agent stores this interaction, enhancing future personalization.

Operationalizing Data Applications

Operationalizing data applications requires more than simply deploying AI models; it demands a structured approach to governance that ensures every phase of the AI lifecycle aligns with organizational goals, compliance requirements, and quality standards. With AI's increasing impact on critical business operations, data leaders must adopt robust frameworks that incorporate oversight, security, and adaptability.

In this section, we'll explore how an AI governance framework can guide organizations in maintaining accuracy, transparency, and accountability throughout the entire AI process—from model

development to ongoing monitoring. By establishing clear governance steps, organizations can manage the complexities of model deployment, performance monitoring, and risk mitigation. The goal is to not only ensure compliance and security but also to support scalable, reliable AI operations that drive true business value.

We'll break down the operational phases, highlight essential governance practices at each step, and demonstrate how a well-designed governance framework enables organizations to operationalize AI applications effectively, achieving both innovation and control.

The AI Governance Framework

Without robust governance, AI applications risk derailing due to unreliable inputs, which can lead to inaccuracies and inefficiencies. The composable data architecture, therefore, emphasizes not only speed but also responsible data use. By implementing strong governance policies and investing in tools that enhance transparency and oversight, organizations can harness AI's potential while maintaining security and data integrity.

To effectively operationalize AI governance, data leaders need a clear, practical approach that brings structure, accountability, and transparency to each phase of the AI lifecycle. Here's how to break down AI governance into manageable steps (Figure 5-2), ensuring compliance, risk management, and alignment with business goals.

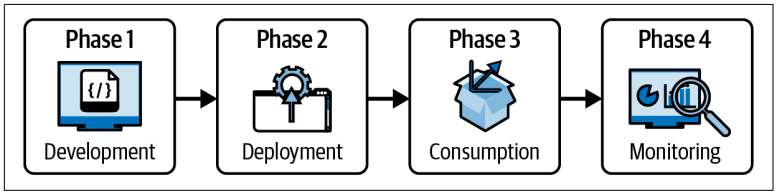


Figure 5-2. The AI governance framework

Phase 1: Development

In the development phase, AI governance starts at the core—model training and fine-tuning. Governance here focuses on verifying data quality, detecting bias, monitoring model drift, selecting models, and rigorously testing for accuracy and fairness. This phase includes tracking training costs and computing requirements, which are critical as costs drop, enabling more teams to fine-tune models and even train custom models internally.

With the vast array of models now available, managing which models enter your environment becomes essential. Model catalogs have emerged as a solution to document models in use, capturing essential details like version numbers, purpose, usage, and risk scores. These catalogs help identify data sources, track fairness scores, and provide transparency for compliance teams. For example, the Azure Model Catalog allows organizations to manage and organize ML models centrally. It provides features such as model versioning, metadata tagging, and deployment tracking, enabling seamless integration with operational workflows while ensuring traceability and compliance. Ideally, this model catalog should extend from your data catalog, aligning data and AI governance into a single, cohesive system.

Phase 2: Deployment

During deployment, governance ensures that only approved models make it into production, with all regulatory and operational approvals in place. Here, risk management involves mapping approved models to specific business cases, addressing explainability, performance, and data security. Tracking models as they transition into production is vital, including security checks for potential risks like data misuse and data loss.

This phase also emphasizes traceability—every model's lineage should be visible, showing approvals from legal, the chief information security officer (CISO), CDO, and other stakeholders, with processes to regularly reevaluate the model's performance and validity.

Phase 3: Consumption

When models are in use, governance shifts to overseeing how they're applied in real-world contexts, ensuring that the business use cases align with model outputs. Consumption governance addresses risk identification and mitigation, interpretability, cost, and performance. This phase involves monitoring model usage, identifying potential issues such as bias or drift, and ensuring compliance with organizational risk thresholds.

AI governance tools should enable risk documentation and mitigation strategies, providing stakeholders with model explanations to enhance transparency and trust.

Phase 4: Monitoring

Once models are deployed, ongoing monitoring becomes essential to maintain their reliability. Monitoring tools should track performance metrics, detect model drift, and scan for unwanted behaviors such as hallucinations. Given the unpredictability in AI model responses, continuous oversight is critical to ensure accuracy and relevance.

Automated alerts and notifications are necessary but must be tuned to avoid alert fatigue. Key governance metrics should include accuracy, cost, risk exposure, and relevance over time, all aligned with security and privacy protocols.

AI-Powered Data Governance

AI-powered data governance helps automate the enforcement of data policies, security protocols, and compliance standards across the organization's data platform. By embedding governance directly into operational workflows, organizations can ensure adherence to regulations without needing constant manual oversight. This AI-driven approach to governance allows for continuous monitoring and quick responses to compliance issues, significantly reducing the risk of noncompliance while freeing up human resources.

With new regulations on AI emerging globally, compliance is more important than ever. Recently released frameworks, like the EU AI Act, ISO 42001, and OpenAI's Preparedness Framework, are pushing organizations to build more structured and responsible AI practices. These standards aim to track, evaluate, forecast, and protect against AI risks, laying the groundwork for transparent, ethical AI deployment.

We are starting to see a heightened focus on AI governance across the industry, with vendors like IBM and Databricks expanding their offerings to include model and data catalogs that converge data governance and AI oversight, paving the way for more comprehensive governance frameworks.

To address this increasing scrutiny, a structured approach to AI operations is essential. LLMOps provides a path forward, addressing unique challenges such as prompt engineering, real-time observability, and dynamic model deployment. By implementing frameworks tailored to the nuances of LLMs, it will enable you to go beyond

innovation, creating AI-driven applications that are production-ready, scalable, and resilient.

Operationalizing these applications requires careful attention to governance, risk management, and continuous monitoring. Tools like model scorecards, usage dashboards, and automated workflows support proactive responses to model drift, unexpected outputs, and system vulnerabilities. With systems in place for versioning, tracking, and issue escalation, organizations can maintain robust AI applications that adapt to real-world demands and regulatory requirements.

Building the Complete Capability Model

As we near the end of our composable data architecture journey, we can bring together the diverse capabilities that enable an agile, resilient, and scalable data ecosystem, resulting in our completed composable capability model (Figure 5-3). Each layer in the composable data architecture serves a unique purpose, allowing data leaders to seamlessly integrate, manage, and govern AI-driven workflows, data pipelines, and applications. This final capability model provides a blueprint for how these components interact to create a dynamic, interconnected system.

In this chapter we've introduced the final group of three capabilities that complete our model:

AI agents
: Autonomous AI agents extend the capabilities of copilots by making decisions, automating complex workflows, and supporting advanced analytics in real time.

AI data governance
: AI data governance provides frameworks and tools to monitor compliance, privacy, and ethical standards for AI data.

AI operations
: AI operations facilitate ongoing maintenance, monitoring, and adjustments to keep AI applications running smoothly and securely.

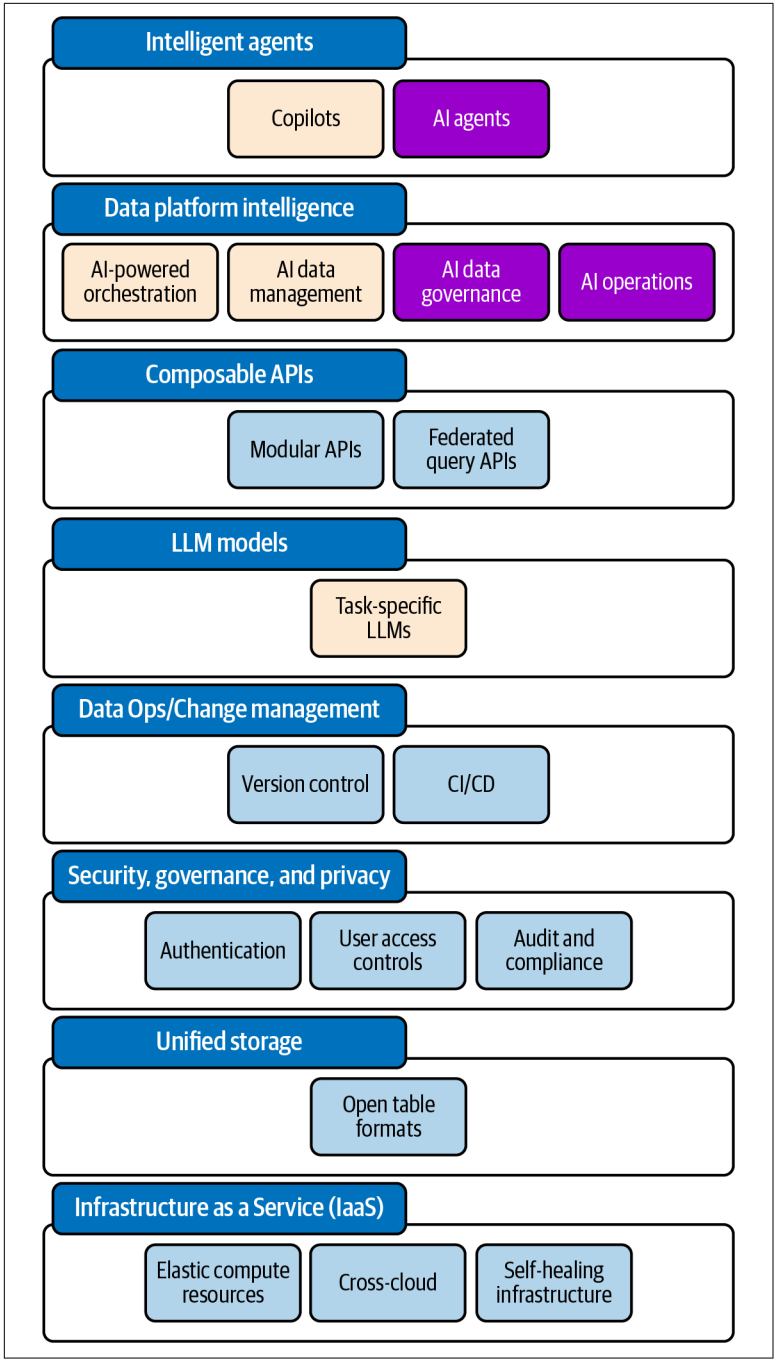


Figure 5-3. The complete composable capability model

Not only does this capability model embody the modularity and scalability that are core principles of composable architectures, but it also emphasizes the need for seamless integration and robust governance across layers. By implementing this model, organizations can transform their data ecosystem, empowering data teams to work more effectively and securely, accelerating innovation, and delivering sustainable business value.

This final capability model brings together all of the principles and practices explored throughout this guide, offering a structured approach for any organization seeking to future-proof its data architecture and fully leverage AI-driven insights.

Key Takeaways for Data Leaders

As organizations increasingly integrate AI into their data infrastructure, successful implementation requires careful attention to several critical areas. The following takeaways provide a framework for building robust, scalable, and efficient AI-enabled data systems:

Prioritize dynamic AI workload balancing
: To maintain stability and efficiency during peak processing times, deploy AI systems that can automatically balance workloads based on real-time metrics. This enables you to maximize infrastructure reliability without compromising efficiency, which is essential for supporting complex applications like LLMs and intensive data analytics.

Enhance interoperability across AI components
: Ensure that AI components within your system work together seamlessly. This interoperability allows for flexible workflows, enabling your teams to leverage various AI tools based on evolving project needs. A well-integrated AI ecosystem supports responsive and agile operations, ready to adapt to new opportunities or challenges.

Implement automated data quality management
: Consistent, high-quality data is critical. Leverage AI for continuous monitoring and validation to quickly detect and address data anomalies, reducing manual checks and ensuring that reliable data flows through your systems. This approach not only saves time but also strengthens the accuracy of your AI-driven insights.

Leverage AI-driven recommendations
: AI-powered recommendations provide personalized data insights based on user activity, helping your data consumers access relevant information faster. This targeted guidance enhances productivity and ensures that key data points are not missed, empowering decision makers to make data-informed choices.

Adopt a comprehensive AI governance framework
: Establish a governance framework that extends beyond model development, covering the entire lifecycle from deployment to ongoing monitoring. This framework should ensure compliance, data privacy, and security while minimizing operational risks.

Establish prompt engineering best practices
: With the introduction of prompt-based systems, invest in structured prompt engineering to ensure consistent and accurate model outputs. Standardize prompt templates and implement version control to maintain continuity, even as team members transition.

Implement real-time observability for AI models
: Deploy real-time observability tools to monitor model performance across prompts and outputs. This enables quick identification and resolution of issues such as biases, ethical concerns, or unexpected behaviors, maintaining trust in AI applications.

Summary

This chapter has outlined the core operational strategies for maximizing the potential of AI-driven intelligence within data workflows. By embedding AI capabilities such as modular AI/ML services, real-time workload balancing, automated data quality management, and proactive resource optimization, organizations can transform their data infrastructure into a responsive and scalable system. The insights and recommendations generated by AI empower teams to work more efficiently, make informed decisions, and drive continuous value across the business.

Integrating these capabilities creates a powerful AI operational framework that aligns with business objectives while ensuring compliance and data security. As data leaders bring together these

AI-driven tools, they lay the foundation for a dynamic data architecture capable of adapting to the evolving demands of modern business. This holistic approach to AI-powered operations positions organizations to navigate the future with confidence, leveraging data as a central engine for growth and innovation.

With a comprehensive understanding of our composable capability model, we hope you feel confident and well positioned to scale your AI initiatives, manage compliance, and deliver impactful, data-driven insights across the organization. However, the journey doesn't stop at implementing a composable data architecture. In the next chapter, we'll explore the broader implications of using composable data architectures through real-world case studies that showcase how these capabilities translate to tangible business outcomes. We'll also look ahead to future trends and considerations, examining how evolving technology and business needs will continue to shape data architecture strategies. This perspective will provide you with a deeper understanding of the transformative potential—and practical challenges—of fully embracing composable data architectures.

CHAPTER 6

Implications of Using Composable Data Architectures

As discussed, composable data architectures are more than just a technological trend—they are reshaping the way companies think about agility, scalability, and innovation. Data leaders who adopt composable architectures position their organizations to meet the demands of today's data-driven market with flexibility and speed, driving transformative outcomes while keeping pace with complex regulatory and operational challenges.

We've explored how composable architectures empower organizations to design scalable, AI-ready data environments that adapt as quickly as business needs evolve. But as this approach matures, a new set of considerations is emerging, spanning the technical, ethical, and strategic realms. Building a composable architecture means navigating critical questions: What operational, regulatory, and ethical challenges will we face? What can we learn from others who have been through this journey? How do we manage technical risks in AI-driven workflows? What trends are likely to impact our investments in composable solutions over the next five years?

In this chapter, we'll aim to answer these questions by shedding light on the practical implications of implementing composable data architectures, examining both the benefits and potential pitfalls. Real-world case studies will illustrate how leading organizations have transformed their data strategies and how they overcame the obstacles they encountered. We'll also explore future trends that

data leaders should monitor closely, from advances in synthetic data to increased regulation of AI-driven tools, all of which will shape how enterprises approach their data architectures. We hope this information will assist you to make informed decisions that prepare your teams and systems not only to meet today's needs but also to thrive in the data landscapes of tomorrow.

Case Study: Transforming Data Agility with Composable Architecture

We can derive a lot of value by directly examining organizations that have successfully transitioned to composable data architecture, leaving behind many of the constraints of the traditional monolithic architecture, which, as you'll soon discover, began to place significant constraints on how the subject of this case study could respond to market changes and business requirements. This case study, drawn from a leading global exchange, demonstrates how they navigated the complexities of on-premises, legacy systems while adapting to a rapidly evolving regulatory and market environment.

In this case study, the organization shares its real-world challenges, strategies, and outcomes, providing practical lessons and actionable advice. Their experiences bring the concepts we've discussed in this guide to life, empowering you to validate your strategies, anticipate challenges, and uncover new opportunities for innovation.

For years, the data warehouse solution that once served the organization well began to become outdated and expensive to maintain when compared with newer, cloud native SaaS offerings. The company's data architecture relied on an on-premises Oracle Exadata system, along with several other Oracle databases (see Figure 6-1).

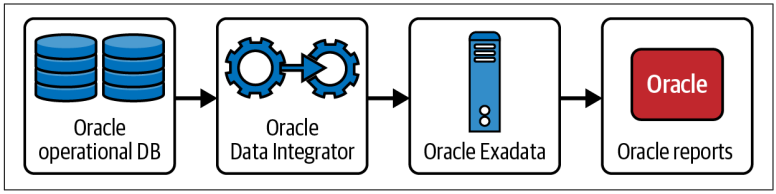


Figure 6-1. The on-premises Oracle solution was outdated and expensive to maintain

As the scale of the business continued to grow, their architecture was becoming prohibitively expensive, difficult to scale, and inflexible for their growing needs. The exchange grappled with a patchwork of tools and workflows inherited through mergers and acquisitions, including Oracle Real Application Clusters (RAC), Oracle Data Integrator, and in-house-built ETL tools, leading to inefficiencies and support challenges relating to software patching, upgrades, and end-of-life cycles.

In addition, the introduction of new regulations like MiFID II magnified these inefficiencies; teams were forced to identify and update complex business logics that had been replicated across disparate systems, tripling the workload and resource requirements. Adding to the challenge, their existing infrastructure struggled to handle the high volumes of data generated daily—up to 500 million rows per day—especially during volatile market events where data volumes spiked by 200% for just a day or two. The client couldn't afford the additional headroom needed to handle unexpected peaks on their on-premises platform. As a result, they made significant compromises. They limited the data stored on the platform to what was essential to the business and accepted slower processing times during periods of high demand.

To address these challenges, following an extensive vendor selection process, the organization made a strategic shift to embrace a composable data architecture centered on Snowflake and Matillion. The transition began cautiously in 2020, unfortunately coinciding with the COVID-19 lockdowns. Their goal was not merely a lift and shift but a complete reengineering of processes to embrace cloud scalability and ELT principles. Matillion stood out during the POC phase for its simplicity, performance, and ability to handle their use cases without requiring extensive professional services.

The adoption of Snowflake's cloud-first platform and Matillion's ELT capabilities eliminated these constraints. Snowflake's dynamic scalability enabled the client to adjust resources in real time, seamlessly handling spikes in data volumes without over-provisioning or incurring excessive costs. This flexibility ensured that the organization could store and process all required data without compromising speed or regulatory obligations (Figure 6-2).

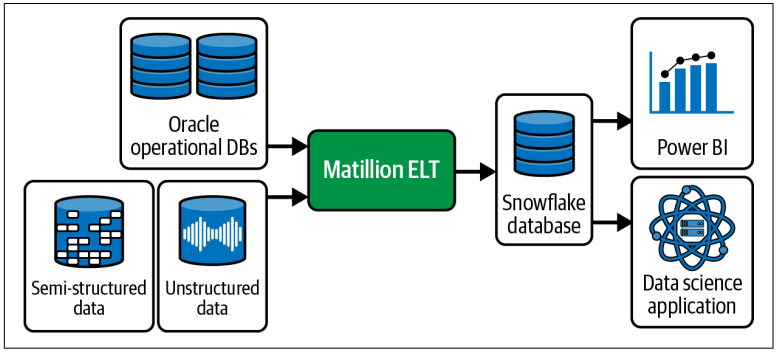


Figure 6-2. The new cloud-based composable architecture improving speed of delivery

Matillion's user-friendly interface simplified the process of building and managing data pipelines, enabling teams to design transformations with drag-and-drop functionality. This ease of use not only reduced the time to value but also lowered the learning curve for new team members, cutting onboarding times from three months to just four weeks.

Following the successful transition 18 months after the project began, the exchange runs over 700 pipelines daily, leveraging the scalability and efficiency of Snowflake for high-volume data transformations and using Matillion for orchestrating batch ELT workflows. This composable setup has streamlined operations, reduced onboarding time for new team members, and enabled faster delivery of insights to the business. Despite challenges in governance and resource availability, the team now operates with greater agility, supporting business-critical functions with a scalable, efficient, and cost-effective data architecture.

The intuitive nature of these tools has empowered data engineers to focus on strategic tasks rather than troubleshooting or managing cumbersome systems. By integrating Snowflake and Matillion, the organization achieved faster delivery of insights, more efficient regulatory compliance, and a scalable platform that could adapt to future needs. Teams could now implement changes quickly and confidently, knowing their architecture could support both the immediate demands and long-term growth.

Looking ahead, the exchange is beginning to explore AI-powered innovations, including leveraging LLMs for purposes such as automating data analysis, enhancing predictive modeling, and generating actionable insights. They are also exploring the creation of synthetic data for testing to simulate real-world scenarios, train AI models effectively, and ensure compliance with regulatory requirements. Their lab environment allows data scientists to experiment safely with production data in sandbox environments while adhering to strict regulatory requirements, paving the way for advanced analytics and forecasting capabilities.

We've acknowledged the fact throughout this guide that although the journey to a composable data architecture provides transformational outcomes, the path is far from straightforward and often filled with challenges. To help you navigate these challenges and exploit opportunities as they arise, this customer has shared some tangible recommendations based on their firsthand successes and obstacles. These insights provide you with a practical roadmap to avoid common pitfalls, overcome hurdles, and accelerate your progress toward building a scalable, agile, and cost-effective data environment:

Build a strong foundation for scalability
: Invest time up front to establish robust frameworks for database structures, permissions, ingestion pipelines, and CI/CD processes. These foundations enable scalability, simplify maintenance, and accelerate future development. Though it may feel slow initially, this groundwork ensures long-term efficiency and adaptability. By setting up repeatable frameworks early, your teams will be better equipped to handle future growth and maintain consistency across projects.

Simplify, standardize, and modularize processes
: Avoid complexity by standardizing workflows for data ingestion, transformation, and storage. Use modular, repeatable

patterns to ensure consistency, improve maintainability, and streamline troubleshooting. Simplified processes enable faster onboarding, reduce operational costs, and enhance collaboration across teams. Reducing the number of ways to perform similar tasks ensures that even during critical incidents, troubleshooting and recovery remain straightforward and efficient. Use modular and reusable templates for common tasks like slowly changing dimensions or data ingestion. This ensures consistency across teams and improves maintainability.

Repeatable patterns also make collaboration and handovers more efficient.

Balance costs and scalability with business needs
: Design scalable architectures that dynamically handle data volume spikes without over-provisioning resources. Use cost-effective practices like demand-driven data ingestion to align expenses with actual business needs. Everyone always asks for "all the data, in near real time," but once they understand the costs of implementing and running this, it may not make good financial sense. This approach ensures you maintain agility without incurring unnecessary costs during high-demand periods. Balancing scalability with cost control prevents unexpected financial strain while delivering the flexibility needed to adapt to volatile business conditions.

Focus on usability and agility with governance "baked in"
: Select tools with intuitive interfaces, such as Matillion, to reduce onboarding times and empower teams to work efficiently. User-friendly tools lower the learning curve, enabling quicker value delivery and allowing teams to focus on strategic initiatives rather than operational hurdles. With less time spent learning complex tools, your team can start producing tangible results faster, accelerating business impact. Agile development may outpace governance processes, particularly in regulated industries. Streamline governance frameworks to avoid this becoming a delivery bottleneck. Automate governance where possible to keep pace with faster delivery cycles.

Foster innovation with high-quality data and emerging technologies
: Ensure data quality before adopting advanced AI models or analytics. Provide secure lab environments for experimentation while maintaining governance and compliance. Continuously

evaluate emerging tools like LLMs and synthetic data generators to address evolving business needs without disrupting existing processes. By creating a safe space for innovation and leveraging cutting-edge technologies, you position your organization to remain competitive in an ever-changing market. Before exploring advanced AI models or analytics, ensure your data quality is impeccable. Poor data quality leads to unreliable outcomes, especially with LLMs or synthetic data generation. Start with building a clean, well-structured data environment and then progress to AI experimentation.

Approaching a transition to a composable data architecture requires a mix of technical rigor and strategic foresight. By focusing on scalability, simplicity, and alignment with business goals, data leaders can set their organizations up for long-term success.

Future Trends and Considerations

In this section we'll take a deeper look into where this next wave of AI and intelligence may take us. One thing is for sure: data never sleeps; it continues to grow at exponential rates. Figures vary, but the total global data storage is projected to exceed 200 zettabytes by 2025, with a staggering 80%–90% of that total being unstructured. There's no doubt about the fact that data volumes are continuing to grow, and the pressure to be able to leverage all of this data and a wide range of diverse formats such as documents and images increases the complexity of what data analytics teams need to contend with today.

The Rise of Adaptive Intelligence as a Service

Data leaders are increasingly under pressure to do more with less and, as a consequence, also demand more from the vendors and their platforms. We saw a comparable trend several years ago as vendors started to offer their platforms in a variety of flavors—from IaaS to Paas, and then SaaS. This shared responsibility model allowed customers to offload the responsibilities of managing and maintaining data services within their architecture. We see the next step to be a further evolution of this shift to AI-powered services where vendors integrate and embed intelligence into their service model to focus on delivering solutions that learn and adapt in real time, using LLMs and AI to evolve with business needs.

Artificial intelligence as a service (AIaaS) platforms continually respond and adapt to the conditions of their environments to resolve issues, dynamically optimize resources, and automatically handle scaling to meet demand. Technologies like AI-powered observability, real-time anomaly detection, and predictive maintenance are transforming data architectures from passive structures into active, self-sustaining environments.

Hyper-Personalization of AI Models with Domain-Specific LLMs

Earlier in the guide, we touched upon some of the challenges and risks relating to the use of LLMs, such as hallucinations, and discussed task-specific LLMs within our capability model. You can think of domain-specific as a variation on these task-specific models that aims to improve the relevancy and accuracy for specific industries such as finance, healthcare, and manufacturing.

One example comes from France, where a company called Predictice has gathered over 25 million legal documents to produce a service for legal professionals. Their repository includes all legally accessible court decisions and is updated 24 hours a day. This helps with legal research and includes information such as court decisions.

BloombergGPT is Bloomberg's LLM model created using over 50 billion parameters specifically trained on a vast array of financial data. This model is designed to enhance various natural language processing tasks within the financial sector, including sentiment analysis, named entity recognition, news classification, and question answering.

This shift allows organizations to use specialized AI agents and models that understand industry-specific terminology, regulations, and data patterns, delivering more relevant insights and reducing generic AI errors. We're beginning to see an emerging trend toward a marketplace of pretrained domain-specific models, allowing you to "plug and play" these agents directly into your composable architecture, enabling faster deployment and higher-quality outputs by augmenting it with your own curated data assets.

Increased Emphasis on Responsible AI and Transparent AI Governance

Regulatory bodies worldwide are ramping up requirements around AI transparency, bias mitigation, and data privacy. Enterprises are adopting responsible AI frameworks that incorporate explainability, risk assessments, and continuous monitoring to ensure compliance and trustworthiness. This trend is pushing organizations to treat AI as an operational and strategic asset that requires as much governance and oversight as traditional systems.

Data Collaboration Through Secure Data Sharing and Data Clean Rooms

Data clean rooms are becoming pivotal for companies collaborating on data insights while maintaining privacy and security. These secure environments enable data sharing across organizations without exposing raw data, supporting joint analytics, cross-industry benchmarking, and collaborative AI model training. Data clean rooms offer a compliant, controlled way to share insights and drive data innovation across industries.

The AI-Enhanced Center of Excellence for Data and AI Strategy

As the strategic importance of data and AI grows, many organizations are evolving their centers of excellence to incorporate AI-driven governance, training, and quality assurance. These CoEs foster best practices, support cross-functional collaboration, and manage AI capabilities to ensure alignment with business objectives. An AI-enhanced CoE enables data teams to be more proactive in governance, ensuring responsible, scalable AI deployment.

Implications of Using Intelligence-Driven Composable Architectures

The widespread integration of AI into composable architectures has created unprecedented opportunities to enhance agility, scalability, and innovation. However, this shift also brings new complexities that data leaders must navigate carefully to harness AI's potential while mitigating risks. Here are the key considerations.

Governance and Risk Mitigation

AI systems, particularly LLMs, introduce unique challenges. As highlighted in Chapter 4, LLMs can exhibit "hallucinations," where outputs appear accurate but lack deep contextual accuracy. This unpredictability necessitates robust governance frameworks to oversee their deployment and usage effectively.

The following are opportunities to leverage intelligence to improve governance:

- Automating quality checks and code generation can streamline workflows and reduce development time.
- Intelligent monitoring tools, such as copilots, can detect and correct anomalies in real time.

You must also be conscious of the following risks when using intelligence-driven processes:

- Overreliance on machine-generated outputs without oversight risks technical debt and errors in production systems.
- Lack of governance can lead to compliance violations, especially in regulated industries.

To manage these risks we recommend you consider establishing stringent validation processes, including human oversight and routine audits of AI-generated outputs, to ensure reliability and alignment with business objectives.

Privacy and Security

AI workflows rely on vast data volumes, often sourced from diverse environments. In this chapter, the importance of secure data sharing via clean rooms was emphasized as a way to maintain privacy while collaborating across organizations. Similarly, synthetic data, discussed as an alternative in constrained environments, plays a critical role in protecting sensitive information.

The following are opportunities when leveraging intelligence-driven processes:

- Privacy-preserving technologies such as synthetic data and data masking enable innovation without compromising compliance.

- AI-powered security systems can detect anomalies and potential breaches proactively.

Be aware of these risks when using intelligence-driven processes:

- The potential exposure of sensitive data during processing or training phases poses significant risks.
- Using third-party models can introduce vulnerabilities if the data-sharing frameworks are insufficiently secure.

To mitigate these risks, it is worth considering implementing rigorous security protocols, such as encryption and anonymization, and ensure compliance with data protection regulations like GDPR or CCPA.

Ethics and Bias Management

Bias challenges in AI were highlighted in Chapter 5, particularly in training LLMs. Task-specific models and domain-specific datasets can mitigate these risks but require proactive management to ensure fairness and transparency.

When leveraging intelligence-driven processes, the following opportunities exist:

- Ethical AI practices strengthen stakeholder trust and reduce reputational risks.
- Using fine-tuned LLMs (Chapter 5) tailored to specific use cases can mitigate bias and improve decision-making accuracy.

You must also be conscious of the following risks:

- Skewed training data in general-purpose models may perpetuate systemic biases.
- Lack of ethical oversight risks undermining the credibility of AI-driven decisions.

To mitigate these risks, we recommend you regularly audit AI outputs for fairness and accuracy, and implement human-in-the-loop reviews for high-stakes decisions. Fine-tune general-purpose LLMs with diverse datasets to enhance contextual understanding and reduce bias.

Infrastructure and Resource Constraints

In Chapter 3, the importance of scalable infrastructure and elastic compute resources was emphasized as a core principle of composable architectures. However, AI systems require significant computational power, which can strain existing infrastructure.

Here are some opportunities that exist when leveraging intelligence-driven processes:

- Elastic compute and storage resources (Chapter 3) ensure cost-efficient scalability during peak demand.
- AI-powered orchestration tools dynamically allocate resources based on real-time requirements, optimizing efficiency.

You must also be conscious of the following risks when using intelligence-driven processes:

- Infrastructure limitations can hinder the deployment of resource-intensive AI workflows.
- Rising costs associated with scaling AI workloads can strain budgets, reducing return on investment.

To mitigate these risks, we recommend you lverage cloud native solutions with elastic compute capabilities to scale AI applications efficiently. Monitor infrastructure performance and costs continuously using AI-driven observability tools to maintain cost-effectiveness.

A Balanced Approach

While AI-driven composable architectures promise unparalleled efficiency and scalability, they require careful planning and oversight to succeed. By leveraging the opportunities while proactively addressing risks, your organization can build intelligent, resilient systems that align with strategic goals and foster innovation responsibly.

This balanced perspective aligns with the frameworks outlined in this guide, ensuring that intelligence-driven composability delivers sustainable value while safeguarding against potential pitfalls.

From Strategy to Execution: Practical Steps to Achieve Composability

Throughout this guide, we've introduced the composable capability model, which outlines the modular and scalable components of a modern data architecture, and the autonomous data architecture evolution levels, which chart the path from manual to fully autonomous systems. When used together, these frameworks enable organizations to build architectures that are not only modular and adaptable but also capable of self-optimization and resilience.

Now, it's time to translate these frameworks into actionable steps. By integrating the autonomous framework levels we introduced in Chapter 2 into each phase of the development lifecycle, we can create systems that not only respond to today's needs but also evolve dynamically over time. This section brings together the concepts we've discussed, offering a unified roadmap for implementing composable architectures effectively.

Step 1: Understanding the Ecosystem

The journey to composability begins with a comprehensive understanding of your existing data ecosystem. Mapping your current technologies, workflows, and dependencies is essential for identifying areas where modularity and scalability can deliver the most value.

To understand your current ecosystem, follow these steps:

1. Conduct an ecosystem audit to catalog tools, workflows, and dependencies, identifying areas that can benefit from modular design or automation.
2. Use the composable capability model to evaluate your current architecture against key components, such as unified storage, orchestration, and governance.
3. Introduce assisted automation tools like metadata capture, simple pipeline monitoring, or CI/CD workflows to reduce manual effort and enhance visibility.

Following these steps will result in a clear understanding of your ecosystem and a roadmap for modularization that aligns with early-stage automation.

Step 2: Identifying the Need for Composability

Composable architectures are not one-size-fits-all solutions. In this step we are aiming to align business goals with intelligent design. Composable architectures thrive when paired with targeted, high-impact use cases. At this stage, organizations should focus on bridging Level 3 (augmented) capabilities with modular components from the composable capability model to achieve measurable business outcomes. We recommend you focus on the areas that will return the most value for your business to secure buy-in and build momentum while securing some early wins.

To identify the need for composability take the following steps:

1. Identify strategic areas for composability, such as enabling secure data sharing, reducing pipeline latency, or accelerating AI/ML deployment.
2. Use the composable capability model to prioritize areas where modular APIs, federated query capabilities, or task-specific LLMs can deliver immediate value.
3. Introduce augmented capabilities, such as copilots, to automate repetitive tasks like schema generation, data profiling, or basic quality checks.

Following these steps will result in targeted use cases that demonstrate the power of modularity and augmented automation, laying the groundwork for broader adoption.

Step 3: Architecting the Solution

With priorities in place, the next step is designing the architecture. This involves applying composable principles to create self-contained modules that address specific needs while maintaining interoperability. In this step we need to think about how we best design for scalability and adaptive intelligence. This phase integrates the principles of Level 4 (agentic) autonomy into the architecture design process. Modular components from the composable capability model are enhanced with AI-driven tools, enabling the architecture to adapt dynamically to changing needs.

To architect the solution, follow these steps:

1. Implement modular, scalable pipelines that leverage components, such as open table formats, elastic compute, and intelligent orchestration.
2. Embed intelligent agents, such as AI agents, to manage workflows dynamically, respond to anomalies, and optimize resources in real time.
3. Use AI-driven governance capabilities to enforce data policies, ensure compliance, and maintain transparency across platforms.
4. Design systems to support multiplatform environments, leveraging composable APIs and metadata-driven organization to enable seamless interoperability.

Following these steps will result in a modular, scalable architecture that incorporates adaptive intelligence for enhanced performance and operational efficiency.

Step 4: Building and Iterating

Composable architectures thrive on iteration. Your business cannot stand still, and neither can your architecture—it's a living ecosystem. By adopting an agile approach, your organization can refine and optimize its architecture to adapt to changing needs and opportunities. This final step strives to reach Level 5 (full automation) in the autonomy framework, where systems achieve self-healing, self-optimization, and resilience. Modular components are refined through iterative feedback, creating a dynamic architecture that evolves with the organization's needs.

To build and provide iterative business value, follow these steps:

1. Pilot self-optimizing capabilities on high-priority use cases, such as real-time anomaly detection or dynamic workload balancing.
2. Implement real-time observability tools to monitor pipeline performance, feeding insights back into optimization algorithms.

3. Use agile methodologies to iterate on modular components, ensuring they remain aligned with business objectives and scalable for future growth.
4. Expand autonomous capabilities incrementally, scaling self-healing pipelines and AI-driven governance across the organization.

Following these steps will result in a fully integrated, composable, and autonomous architecture that delivers agility, scalability, and innovation.

Summary

The four-step roadmap—Understanding the Ecosystem, Identifying the Need for Composability, Architecting the Solution, and Building and Iterating—offers a structured approach to implementing composable data architectures.

By combining the composable capability model with the autonomous data architecture evolution levels along with our real-world case study experiences, we hope your organization can achieve a balanced approach that emphasizes both modularity and intelligence:

- The *composable capability model* provides the structural blueprint for modular, scalable architectures.
- *Autonomous levels* guide the evolution of these architectures toward self-optimization and resilience.

Composable data architectures, when paired with autonomous, intelligent capabilities, represent a paradigm shift in how organizations manage data. This roadmap ensures both frameworks are integrated into a cohesive strategy:

Establish modularity
: Begin with the core components of the composable capability model to build scalable, flexible systems.

Enhance automation
: Progress through the autonomous levels to reduce manual effort and increase efficiency.

Drive innovation
: Leverage intelligent tools and frameworks to empower teams and unlock new opportunities.

Achieve resilience
: Scale toward self-optimization and continuous improvement, ensuring your architecture evolves with your business.

By following this roadmap, you can transform your data ecosystem into a powerful engine for growth and innovation, capable of meeting today's demands while anticipating tomorrow's challenges.

Conclusion

The journey toward composable data architectures has become essential as organizations strive to manage, scale, and innovate their data ecosystems. The vision behind composable architecture goes beyond solving today's business needs—it's about creating a resilient and future-ready architecture that adapts and grows in tandem with organizational goals. By balancing composable principles with autonomy, teams can transition from just managing data to enhancing and streamlining its impact on business outcomes.

As we close this guide, we return to the central themes that have shaped our exploration: adaptability, scalability, and a strategic approach to harnessing the power of AI in composable data architectures.

This guide has explored how composable principles and autonomous capabilities can transform the way organizations approach data management. By embracing these frameworks, leaders can move beyond merely managing data to actively creating value from it. The core pillars of composability—modularity, scalability, and interoperability—offer the flexibility needed to navigate today's dynamic business environment. When integrated with AI-driven autonomy, these architectures become powerful engines of efficiency, innovation, and resilience.

Embracing Modularity for Resilience and Flexibility

The evolution of data architecture has shifted the focus from infrastructure-centric solutions to modular, business-aligned systems. Chapter 2 in this guide laid the groundwork by introducing the principles of composable architectures, emphasizing how modularity empowers organizations to adapt quickly to change. Modularity enables:

Reconfiguration
: Organizations can disassemble and reassemble components as business needs evolve, ensuring continuous alignment with strategic objectives.

Collaboration
: Shared frameworks and tools improve interoperability, making it easier for cross-functional teams to work together seamlessly.

Scalability
: Modular designs ensure that growth is not hindered by technical constraints, allowing systems to scale fluidly as data volumes and complexity increase.

For example, the composable capability model provides a roadmap for implementing modular solutions that reduce inefficiencies and encourage reusability. By standardizing workflows and using modular APIs, organizations can eliminate redundancies, shorten development cycles, and improve overall system agility.

Leveraging Modular Data Pipeline Solutions

In Chapter 3 we emphasized the critical role of modularity and intelligent automation in designing data pipelines, and this is where these solutions shine. Products that provide end-to-end pipeline management are instrumental in realizing composable architectures. By enabling organizations to build pipelines with drag-and-drop simplicity, implement ELT processes, and scale dynamically, these tools align perfectly with the composable capability model.

Key contributions include the following:

Simplified pipeline management
: Intuitive interfaces reduce the learning curve for teams, enabling faster onboarding and efficient pipeline design.

Dynamic scalability
: Leveraging elastic compute and storage resources ensures that pipelines can handle variable workloads without overprovisioning or downtime.

Integrated automation
: AI-powered features, such as copilots for schema generation or data quality checks, eliminate repetitive tasks, freeing teams to focus on strategic objectives.

Data pipelines not only embody the principles discussed in Chapter 3 but also accelerate their adoption by providing ready-to-use tools that align with composable and autonomous frameworks. Organizations leveraging these products gain a competitive edge by reducing time to value and maintaining operational excellence.

AI and Automation

As we explored in Chapters 4 and 5, AI is the linchpin of autonomous, composable architectures. It drives innovation by automating repetitive tasks, optimizing workflows, and enabling intelligent decision making. Large language models (LLMs), copilots, and intelligent agents play pivotal roles in enabling this transformation:

Opportunities in AI integration
: AI-driven capabilities like copilots simplify pipeline management by automating schema generation, data profiling, and quality checks. Task-specific LLMs fine-tune operations by improving data labeling, anomaly detection, and model training. These tools reduce manual effort, enhance accuracy, and accelerate time to value.

Addressing risks
: Chapters 5 and 6 highlighted the risks associated with AI integration, such as potential biases, hallucinations, and overreliance on machine-generated outputs. Effective governance frameworks, as outlined in Chapter 6, ensure these risks are mitigated while maintaining transparency and accountability.

By embedding AI throughout the data pipeline—from design to deployment—organizations can achieve autonomous levels that enhance efficiency and reduce downtime. The journey to Level 5 autonomy, as described in the autonomous data architecture evolution levels, represents the pinnacle of this transformation, where systems self-diagnose, self-optimize, and adapt dynamically to evolving business needs.

Driving Innovation Through Composability and Collaboration

One of the most significant benefits of composable architectures is their ability to foster innovation by breaking down silos and encouraging collaboration. As we discussed in Chapter 6, composability enables seamless data sharing, secure collaboration through clean rooms, and the integration of diverse data sources. These capabilities allow organizations to:

Leverage advanced analytics
: By integrating synthetic data and domain-specific LLMs, teams can explore new use cases without compromising privacy or security. This accelerates model development and improves decision-making accuracy.

Promote interoperability
: Open table formats and federated query APIs make it easier to share insights and collaborate across departments or organizations, as illustrated in Chapter 3.

Enhance governance and security
: AI-powered governance ensures compliance with data privacy regulations and protects sensitive information, enabling innovation without unnecessary risk.

The real-world case study included in this guide demonstrates how composable architectures empower organizations to implement cutting-edge technologies while maintaining operational integrity. These examples provide practical insights into navigating challenges and leveraging opportunities to achieve transformative outcomes.

Leading with Vision

At its core, composable architecture represents a shift in how we approach data strategy. Instead of focusing solely on technical infrastructure, it empowers data leaders to design systems that are adaptable, scalable, and aligned with business priorities. By embracing composability, organizations can move beyond traditional silos, creating interconnected data ecosystems that support a diverse range of stakeholders—from data engineers and scientists to business analysts and executives.

As we conclude, here are key takeaways to help guide data leaders as they embark on this journey:

Adopt a strategic mindset
: Data architecture should be driven by business needs, not technology for its own sake. By aligning architecture with strategic priorities, data leaders can ensure that every capability contributes directly to business outcomes.

Prioritize adaptability and resilience
: The pace of change in data technology is accelerating, and architectures must be able to evolve alongside it. Composable principles enable organizations to stay agile and responsive, maintaining a competitive edge.

Champion responsible AI practices
: As AI becomes more integrated into data workflows, leaders must implement robust governance, transparency, and ethical guidelines. Responsible AI is not just a regulatory necessity; it's a foundation for sustainable innovation.

Cultivate a culture of continuous improvement
: Autonomy is not a one-time achievement; it requires ongoing refinement and optimization. By fostering a culture that values learning and innovation, organizations can ensure that their data systems continue to deliver value as they evolve.

Champion composability as a strategic advantage
: Composability is more than a technical approach; it's a strategy for resilience and adaptability. Encourage cross-functional collaboration and invest in modular solutions that enable teams to innovate, scale, and pivot without disrupting the overall architecture.

Leverage AI with caution and control
: While AI drives efficiency and innovation, it requires responsible use. Implement robust governance to oversee AI-driven components, ensuring transparency, accountability, and alignment with business objectives.

Maintain flexibility through modular design
: A composable architecture thrives on flexibility. Prioritize modular design, orchestrate processes to optimize data flows, and ensure that your architecture can easily integrate new tools or replace outdated components.

Embrace continuous improvement
: Composable data architectures are designed for evolution. Regularly assess and optimize each component, building feedback loops that drive ongoing improvements and allow the architecture to stay aligned with business goals.

Build for the long term
: Composable architectures are a commitment to future-readiness. Balancing immediate needs with long-term strategy ensures that the architecture remains adaptable, scalable, and prepared for emerging opportunities.

The Future of Composable Data Architectures

Looking ahead, composable architectures are poised to become the gold standard for data management. With advances in AI, data science, and autonomous operations, we anticipate a future where data systems are fully self-sustaining, capable of delivering insights and managing workflows without human intervention. This vision is ambitious but achievable—and by following the principles, frameworks, and models outlined in this guide, data leaders can position their organizations to succeed in this new era of data innovation.

Composable architectures offer a way to stay resilient, agile, and aligned with business goals. By embracing modularity, interoperability, and autonomy, data leaders can create systems that not only meet today's demands but also adapt to the unknown challenges and opportunities of tomorrow.

By embracing composable principles, building autonomous levels, and following the capability model, organizations can move beyond traditional data management. Instead, they enter a new era of

data-driven innovation and agility, where data is a strategic asset enabling growth and transformation.

The composable data architecture journey is one of continuous exploration, discovery, and advancement. As you move forward, remember that composability is a dynamic and evolving path. It requires commitment, strategic alignment, and a willingness to adapt as new technologies and business needs arise. The capability model and autonomous levels provide a guiding framework, but the ultimate success of a composable architecture depends on a mindset that values flexibility, accountability, and collaboration. With these principles at the core, you're well prepared to navigate the complexities of modern data management and to harness the full power of AI in building a competitive, future-proof enterprise.

About the Author

Adam Morton is a distinguished data leader and author whose expertise has earned international recognition in the field of data and analytics. His outstanding contributions to the industry were formally acknowledged when he was awarded a Global Talent Visa by the Australian government, marking him as a leader of exceptional ability in his field.

As the founder of Mastering Snowflake, Adam works at the forefront of data transformation, partnering with leading organizations across insurance, banking, and financial services. His approach combines deep technical knowledge with a passionate focus on delivering measurable business outcomes, helping organizations break down data silos, streamline operations, and drive innovation through data.

Adam is the author of three books, including the Amazon best-selling *SnowPro Core Study Guide*, and has enabled thousands of professionals globally to maximize their data investments. His practical, business-focused methodology reflects over two decades of hands-on experience in navigating complex data challenges and architecting solutions that deliver tangible results. Through Mastering Snowflake's services, including rapid implementation programs and strategic consulting, he continues to help organizations unlock the full potential of their data and achieve operational excellence.