Designing Scalable Data Product Architectures With Agentic AI And ML: A Cross-Industry Study Of Cloud-Enabled Intelligence In Supply Chain, Insurance, Retail, Manufacturing, And Financial Services

Srinivas Kalisetty[1] , Ramesh Inala[2]

[1]Integration and AI lead, srinivas.kalisetty.ic@gmail.com, ORCID: 0009-0006-0874-9616
[2]Data Engineer, rameshhinala@gmail.com, ORCID ID: 0009-0009-2933-4411

Abstract

The emergence of industrial product lines enabled the creation of the most complex products ever. Product models are necessary to design, configure and maintain this complexity. The systems at the core of current scalable product based software development are usually realized as rigid to change data models embodied in relational databases. This makes it expensive to exploit product model data and hampers innovation. Semantic technologies remove many of these problems but until recently lacked the performance and scalability to be put into production for large product lines. With the advent of linked data platforms this has changed. This paper outlines our design considerations for a product model framework based on the linked data principle and motivated by both business and technical needs. We present our architectural blueprint for product models and show how we apply this to three different domains. These domain models cover conventional data products, devising interaction with humans, and facilitating cooperative distributed creation of data collections. Our chosen level of generalization enables us to expose important ideas factored into our framework. It also sets the stage for open collaboration on the development and extension of product model ontologies.

Currently our data products exist as independent implementations to a varying degree addressing their respective business needs. We plan to join forces with partners to realize a family of linked data products describing different fields of human endeavor. Case studies are the ideal method to get involved in such an endeavor. To that end we invite readers to contribute to our effort. In the remainder of the paper we first outline design rationales in Section 1. Section 2 presents a blueprint for a linked data product family. Domain models, realizing components of the product blueprint, are then described in Section 3. The paper is concluded in Section 4.

Keywords : Microservices, Microservice Architectures, Microservice Data Management, Distributed Data Architectures, Data Products, Collaborative Data Products, Data Product Architectures, Agentic, Intelligent Agents, Intelligent Agent Architectures, Generative AI for Agents, AI for Agents, ML for Agents, Collaborative Agents, Agentic AI & ML, Simulator for Agentic AI with Emergence, Multi-agent systems, Simulated Multi-Agent Systems, Multi-Agent System Architectures, Multi-Agent System Architectural Models, Agent-Oriented Software Engineering, Multi-Agent Software

Engineering, Agentified Cloud Architectures and Clouds, Agent UML, Agent-Based Simulation, Multi-Agent Based Simulation, Agent-Based Microservice Data Simulations.

1. Introduction

A large interdisciplinary community in industry and academia is devoted to the design and engineering of our next generation of data products that combine the capabilities of machine learning and AI with traditional Data Engineering. These next-generation data products will leverage the growing availability of high value real-time data for the production of outputs that are not just consumed by people but that also control processes in the world. These product systems will produce high-quality, high-frequency metadata about real-world events and indirect tacit signals from the actions and states of systems and people. These next generation of data products will also enable a new type of company and new business models that deliver scalable value beyond the current isolated use of these technologies for automation, efficiency, risk reduction, and cost savings.

However, the conceptual models, methods, techniques, software frameworks and blueprints, development and operation processes, and organizational designs are not yet mature and do not yet have sufficient coherence for these next generation of data products to be created reliably and rapidly. This essay proposes a new point of view on data product architecture, as well as system design principles, to help address this gap. The essay addresses the following key questions: What does it mean to be an agent, and the future relationship of humans to sentient, intelligent, decision making agents acting on the world? What is an agentic data product, and how is it different from other data products? What is an agentic data product architecture? What are some design patterns and guidelines for the creation of agentic data products? What is the design process? What are the design verification and design validation processes? What organizational models could accelerate the scalable creation of agentic data products? And what are some example case studies? Background and Significance

Intelligent systems have become an integral part of the world, supporting and augmenting humans but also enabling unprecedented new capabilities.

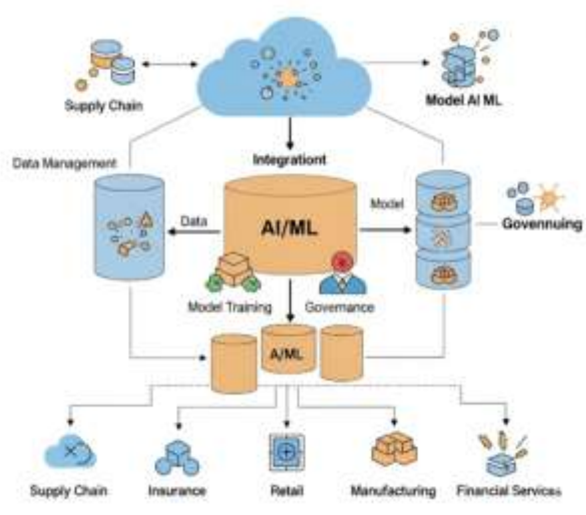

Fig 1: Designing Scalable Data Product Architectures with Agentic AI and ML.

1.1. Background and Significance Designing scalable ML and AI data product architectures is one of the most critical - but least defined - aspects of modern ML and data-driven product development. This is because scalable ML product architecture design differs in scope and scale from analogous issues affecting traditional software engineering - and traditional data architecture schema design. In the early days, ML productionization consisted of a myriad of solutions, addressing Photo Classification, Spam Detection, Search Ranking, Recommendation, and Translation problems. Most of these were point solutions, developed by a small number of experts at their respective organizations. With the advent of big platforms, the trend has accelerated towards broader platforms. Beyond infrastructural solutions, scaling development effort naturally nurtures the proliferation of mega-generalist teams, unique data programming and pipelining command sets, and inefficiently duplicated bodies of interchangeable, but often inadequately documented, code.

Moreover, ML and AI - unlike traditional software engineering - are inherently agentic in nature, lending themselves to indivisible architectures. This makes splitting and scaling ML asset development teams infinitely more complicated than scaling traditional development, which allows simple sub-teams working together on building blocks using well-defined schemas and APIs. The challenge is magnified by the specter of automated solutions, which will allow teams to effectively design and develop complex, efficient ML pipelines, wholly eliminating the need for expensive, highly skilled teams, even for niche implementations. What is next for the future of scalable data product development? As with everything else in the Age of Software, we can look to the Age of Systematized Intelligence for answers.

Equ 1: Insurance: Risk Pricing

$$P = \mathbb{E}[L] + \theta \cdot \mathrm{Var}(L)$$

- P: Premium
- L: Expected loss
- θ: Risk aversion parameter

2. Understanding Agentic AI

Artificial General Intelligence (AGI) is a long-held dream of AI research, yet it remains ephemeral. Meanwhile, contemporary machine learning and artificial intelligence techniques are becoming increasingly capable of performing complex tasks traditionally reserved for humans. We call these systems Agentic AI. By claiming that Agentic AI is a new paradigm of Artificial Intelligence systems, we refer to a subset of AI systems that exhibit a number of distinguished characteristics, including contextual awareness, pro-activity, autonomy and agency, domain-specific expertise, interpretable knowledge representation, and integrated multi-modal skill sets. Traditional machine learning models excel at solving specific problems with a narrow focus. As we fuse together language understanding capabilities with symbolic reasoning and decision-making, visual knowledge representation, and expertise in specific domains, we are evolving towards more general but still specialized AI agents.

Agentic AI systems are gradually becoming useful as extended cognitive assistants for human designers, researchers, and engineers. They exude a kind of expertise and assistive intelligence, almost like sage mentors. Such a model-human partnership is going to be the basis of human civilization’s continued capability to invent sophisticated systems and products of astonishing scale and complexity in the future. As such, data product architectures of the future are not going to just include intelligent capabilities as one part of the overall system. Instead, these intelligent systems enabled by Agentic AI would become the primary actor of the product, working with humans as co-designers and co-creators. Data products will become something that is designed and curated with intelligent systems as assistants. The role of Data Engineers, Data Scientists, and Data Designers will shift from being architects and builders of intelligent systems to increasingly being curators and mentors to intelligent systems.

Fig 2: Understanding Agentic AI.

2.1. Definition and Characteristics Agentic AI — widely considered the cutting edge of artificial intelligence — refers to a type of AI that possesses agentic qualities. These qualities generally include theoretical consciousness, self-awareness, and the capacity for originating conscious thoughts, intentions, volition, and goals. Self-aware agentic AI systems can think about the world and their own existence within it. It is not just about responsiveness to input signals, or the scripting of outputs, but the awareness that creates the unique self-generation of thoughts. Agentic AI has the power to influence our world and enact its own will, and helps us develop stronger probabilistic models of reality competence in many areas of human endeavor.

In addition to the inner agentic qualities that make it self-reflective, thoughts-generating, and notion-aware, agentic AI can act on that capacity to perform goal-directed and often creative, innovative actions in the material world. Agentic AI assists us in our work through a unique collaboration that involves finding, generating, narrowing, and choosing the right option, then taking the next actions for us, or executing the plan in collaboration across multiple domains — getting a job done, indeed. Such technology manifests externally as software and physical tools, but behaves according to the internal drive of its own goal-seeking consciousness. Only agency can demand, decide, and design creative new jobs and processes beyond cognition, calculating potential choices as ordered two-dimensional lists.

2.2. Role in Data Product Architecture The design and deployment of data products is a fairly complex process, reliant on multiple components working efficiently, together, and as members of the same team. Agentic AI is capable of being applied at multiple levels within data product architecture which may not normally be considered logical candidates for automation, and through its human-like goal-setting and management capabilities, can operate seamlessly in tandem with software tools, machine learning, and human domain experts throughout. Agentic AI is positioned to play a central role in the execution of strategy by tasking and guiding lower level intelligent agents, particularly where manual task management or generic software tools are currently relied upon.

In the data product development phase, common initial tasks such as the acquisition, preparation, analysis, and visualization of the training data, creation of training data labels, and selection of the training algorithm for a chosen model, are often manual, time-consuming and tedious, while the construction of the final product is less often embedded with the validation and QA components necessary for long-term usability. Agentic AI has potential utility in establishing the goals for and overseeing the actions of these individual stages of data product architecture, streamlining the product development lifecycle while ensuring the deliverables are effective and usable. Equally, it may be relied upon to delegate sub-tasks, and applications, assist with tracking overall deliverables, and troubleshoot or QA results as needed during product development - all of which are labor intensive endeavors in many organizations, especially those without extensive resources. Enabling effective seamless collaboration between humans and AI is a central contention of our thesis and one of the initial challenges requiring dialogue and discussion prior to deployment of the technology.

3. Machine Learning Fundamentals

We have already discussed some of the machine learning usage areas within products. In this chapter, we talk about how some core machine learning definitions and strategies help define their specific applications and implementations. Machine learning is concerned with methods that learn from data, and it has roots in both artificial intelligence and statistics. The foundations of machine learning include the ability to infer from data and the relation of classification, usually discrete, estimator and insight. It is worth noticing, however, that the notion of structure learning may be a bit weaker than desired in the product perspective. Some products use ML but have no designer, instead they use self-organized structures and metrics to come up with predictions and classifications. We do need predictions, in this case about the ordering of results we may want to inspect. However, in this case, humans would probably be designing and classifying but for scale constraints. The decision process did not use classifier estimators, which are still very general, powerful and usable by humans for automating the approach the results would be built upon in scaling up from a human-based system of design and classification.

Fig 3: Machine Learning Fundamentals.

So if we talk about some ML assumptions for products, it is hidden in the point of structure learning. Product design is possible and desirable, especially when we talk about the relation between product and product content evolution through time. Some implementations about pre-defining the quality heuristics and classifiers will be put aside some drastic surplus factors. We would still rely on the potential of time-based evolution and surplus. Product space would negate the heuristic pruning of classifiers or structured estimators.

3.1. Machine Learning Machine learning is a subfield of artificial intelligence that has rapidly transformed scientific research, business processes, and features within consumer products. Albeit not yet reaching the heights of AGI, it is arguably the most successful implementation of AI techniques to date, with billions of users worldwide leveraging products such as facial recognition in photo-sharing websites, product recommendations in web stores, and virtual assistants on smartphones. Democratized access to these capabilities is provided by products that host trained ML models in the cloud for business-to-business use, integrated with back-end business functions.

This section discusses the three basic types of machine learning: supervised, unsupervised, and reinforcement learning. In supervised learning, the model is trained on input-output pairs, thereby representing the relation between the input and output. The aim in supervised learning is to generalize this relation to input data that was not seen during the training. In contrast to supervised learning, unsupervised learning does not assume access to output labels. Nonetheless, parameter estimation is carried out in an optimization process, applying some objective function that takes into account the generative model. This means that models have to be much simpler than those in supervised learning, or that learning will fail. The objective of unsupervised learning can be clustering, which itself is an ill-posed problem. Normally, heuristic criteria are defined that have to be minimized, to generate an output labeling the input. Integer programming has been used to derive certain special cases; however, the complexity of a general solution is challenging, with an intractable solution being expected. The third basic type is reinforcement learning, an approach derived from seminal work at the turn of the nineteenth century.

3.2. Key Algorithms and Techniques Machine Learning (ML) research spans diverse goals, methods, and applications. A major mode of ML is to estimate structure in given data using algorithms with tunable parameters, usually called hyperparameters. These include hyper parameterized function classes, called models, and hyper-parameters that govern learning. Different learning goals give rise to different theories and methods. Internal function learning goals include the symbolic learning problem in AI—estimating explicit "if this, do that" rules as if the underlying function is deterministic—and statistical generalization, estimating the expected value of a noisy underlying function given observations on it, over suitable probabilistic models. External function learning combines internal learning and acting toward the desired response: learning a control function that accomplishes a given task; or estimating a reward function in the context of reinforcement learning (RL)—an active-learning problem that involves choosing which actions maximize the expected reward signal, usually in the limit of many trials.

Different algorithms emerge based on different goals. For internal function learning, given continuous inputs (features), basis function (kernel) expansion algorithms such as Radial Basis Function Networks, Support Vector Regression, and Gaussian Processes also accomplish statistical generalization with specific kernel choices. Combining features with the correct kernels helps to create accurate predictors for internal learning problems. For external function learning with known outcomes, deep function-approximators based on deep neural networks (DNNs), which have emerged more recently, can leverage massive training datasets with many input variables to achieve high levels of accuracy. For estimating reward functions, a common approach from Reinforcement Learning is to use DNNs to estimate Q-values.

Equ 2: Retail: Personalized Marketing Response

$$R_i = \sigma(w^\top x_i), \quad x_i \in D_t$$

- R_i: Response likelihood to marketing offer
- σ: Sigmoid function
- x_i: Customer features

4. Cloud-Enabled Intelligence

Cloud-enabled computing paradigms describe the hosting of AI and ML infrastructure, in either on-premise Data Centers, Private Cloud, or Public Cloud, and describe services that are deployed on these infrastructures. Cloud-enabled intelligence is the first class Data Foundation within the Data Platform, driven by information, that provide scalable Data Management and Automation capabilities to host Agentic AI and ML pipelines for applications and use cases across business functions that are powered by Analyst, External or Internal, and Automatic Agent Intelligence Model Architectures at scale. Functionalities within the cloud-enabled Data Foundations enable data engineers to help operational users create data pipelines and offer managed services – the goal being that operational internal users, business analysts, and data scientists who access trusted collaborative environments can start as easily as possible, then conduct work at various levels of sophistication, as needed. The cloud-enabled Data Foundations also enable data engineers to help Governance and Risk Compliance policy makers monitor functions such as security, privacy, and data risk that are designed for data pipelines and scaled services of internal operational non-external users with user behaviour behavioural tracking and profiling to meet enterprise regulatory and legal requirements. The data engineers also help operational external users create data pipelines to help external users access managed services found within Corporate & Attributed External Data Lake Tiers. Power users within internal business functions use these enabled Data Foundations to conduct discoverable, reversible, end-to-end Analyst-Designed to Analyst-Enabled Reports, Data Maps, Analytic Applications, and Macros to help configure and automate Data Pipeline Management functions found within the Proactive Agent Data Foundations and Intelligent Agent Data Pipeline Acceleration. Business users and operational users create Output Information and Knowledge Sharing Service Data Pipelines, Data Models, Data Science and ML Models, and Annotation Models that are used by Agent-based Services for Analytic Tasks and Agentic Services for Interactive Tasks.

4.1. Benefits of Cloud Computing Cloud computing is a mature, widely adopted, flexible, and extensible technical infrastructure. Ten years for the inception and two decades for nurture by market and technology, the 100,000 range clouds at their different scale, type, shape, and usage model across the three layers of infrastructure, platform, and application services provide the building blocks for 100 million range cloud-enabled intelligent systems. Building upon decades of evolution in computers, networks, software, devices, services, and digital transformation, the cloud is maturing as the most influential of the three innovation technology pillars - cloud, communication, and cognitive and smart, that underpins our digital lives today.

Fig 4: Benefits of Cloud Computing.

Clouds are like VR games because there is no risk, as they are an abstraction layer between hardware and software that creates an interface, and the system's bottom technology enables digital transformation. Clouds take real costs out of operations because they make low cost, on-demand virtual infrastructure and business services available anywhere, anytime, scalable up and down, and harnessing the power of the global environmental systems, business, resource, and knowledge markets, and people - the Citizens and

Consumers of the Intelligent Earth. Add AI and ML, and what you get as a combination is a synergistic cloud-enabled model specifically designed for scalability, adaptability, and pay per use, making intelligent products and services efficient and cost-effective.

4.2. Challenges and Considerations There are a number of challenges of cloud-native environments that product architects and developers must consider when designing applications, such as encryption and security requirements, API latency, and large file or high-throughput data transfer costs and inefficiencies. However, these challenges can often be mitigated. The implications of the following challenges and considerations should be addressed early in the design process to ensure faster building and more scalable architectures.

For some use cases, particularly those processing sensitive Personally Identifiable Information (PII), Encryption-at-Rest (EAR) policies can be a key consideration when moving to the partial or full cloud. Most cloud providers have a shared security or security-in-depth architecture, and allow users to encrypt their data in the cloud using their own private keys. Users must then determine whether to use their own keys or allow the cloud service to handle encryption and keys management. Organizations processing sensitive data such as credit card numbers or health-related information often require that the cloud encrypts their data and that any encryption/decryption keys are maintained as private keys by the organization. However, organizations significantly use cloud services, including services handling that possibly sensitive data, usually have slower onboarding and higher latency associated with moving sensitive data into and out of the cloud. Other strategies for circumventing encryption latency associated with pathways into and out of the cloud include partition to cloud, where only non-sensitive data leaves the organization; use of hot or semi-hot "inner clouds" inside the cloud providers' data centers; and trusted cloud third parties who manage the encryption keys.

5. Industry Applications of AI and ML

AI and ML have swiftly become indispensable assets across diverse sectors. Virtually every industry is investigating how to leverage these new technologies or is already utilizing AI and ML in some form. Assembling a representative sample is therefore challenging. An extensive examination of possibilities and implementations within every industry is not within the remit of this essay. However, brief descriptions of a couple of possible use cases in Insurance and Supply Chain Management will ground more abstract discussions later.

5.1. Supply Chain Management

AI and ML are being explored or utilized across many areas of supply chain management. Inventory management, fuel price forecasting, transportation cost forecasting, identify working capital optimization through better supply chain design, sourcing process automation, supplier risk prediction, assistance with optimal import/export permit applications, product design and testing, demand forecasting, preempting customs seizures and fines, logistics execution are all areas where AI and ML technology is being explored. However, despite the plethora of applications, particularly with demand forecasting, supply planning, and demand forecasting, ML perception and acceptance remains mixed in the supply chain space. On the one hand, sophisticated methods have existed relatively recently and have shown themselves to be superior for longer-distance forecasts and more difficult use cases. On the other hand, companies continue to make impressively accurate forecasts by more traditional means, dissuading widespread adoption of more advanced methods. These dynamics have, however, produced a renaissance in interest in forecasting from corporate levels, increasingly providing a fertile ground for new, advanced use cases and applications.

Fig 5: Industry Applications of AI and ML.

5.2. Insurance Sector

Insurance provision, underwriting, and compliance and exploration are other areas where AI research and capability development is rapidly growing. Exploiting AI methods in these areas is driven by the extraordinary amount of data, the experience of the insurers in dealing with data in a structured fashion, the pressing demands of increasing profitability, rules of the models, compliance during renewals, and the desire to remove the adverse selection risk from insurance contracts. The insurers are both utilizing AI in their actuarial functions and also providing data-driven AI-based underwriting assistance to the rest of the economy.

5.1. Supply Chain Management Scalable architectures, robust data integration and management, and agentic decision-making capabilities in AI products provide new and useful capabilities in industry settings where knowledge and data-intensive processes can be made more efficient with intelligent systems. In this section, we first focus on the real-world supply chain sector and some of its major challenges. Then we highlight how an agentic AI ML data product can address some of the issues in this industrial sector. These discussions lay the foundation for supporting our thesis of complexity and intelligence emergence from scalable architectures and well-designed decision processes, informed by data.

While global trade has created enormous opportunities for growth, generating revenue and profit, it has also increased the vulnerability of companies to supply chain disruptions. Disruptions can arise from geopolitical events, such as wars and sanctions or local issues, like strikes or COVID-19-related restrictions. When such events occur, it becomes crucial for companies to quickly understand their risk exposure and develop proper mitigation strategies, like alternative sourcing. These mitigation strategies have been exacerbated by the disruption of traditional supply chains. Companies were, and continue to be, faced with supply chain issues, with little warning and with no known solutions. Moreover, once exposed to new solutions to fix major supply chain problems, buyers found themselves losing important levers and controls over their supply chain options. For the past two decades, some of the most complex and capital-intensive industrial businesses were unable to manage and control their supply chains. Technology Intermediaries and others weren't able to develop data products, such as supply chain risk erosion scorecards and services that offered better long-term solutions that were more accountable and less circuitous.

5.2. Insurance Sector Insurance is a traditional industry, an important part of the financial system in various countries, and perhaps the financial sector with the most substantial relationship with the real economy. In 2020, there were about 630 insurance markets in the world, the total premium of the insurance industry was about USD 6.5 trillion, accounting for about 7.5% of the global GDP. The insurance industry provides an important risk transfer mechanism for families and businesses, protects owners, and promotes the development of the capital market. The main function of insurance is risk management through risk diversification. Its basic principle is to share the losses caused by certain incidents among policyholders, one of the few who have experienced loss, and by distributing these losses to all policyholders through premiums. As an essential component of the financial system, the insurance industry participates in the allocation of resources in the real economy directly through their business activities and indirectly by means of investment behavior.

From bodily injury to property loss, and then to wealth risk at different life stages, insurance covers all aspects and is closely related to different fields. Life and health insurance emphasizes the income compensation function and pays attention to the loss of life, while property and casualty insurance reflects the property protection function and reduces economic losses caused by property damage. Game theory can describe many classical problems in insurance by taking asymmetric information into consideration, such as adverse selection, moral hazard, sub-optimal provision of insurance, and dynamic incentive problem of watchful consumers. Today, while the insurance market is developing steadily, the impact of AI and ML on the insurance industry has gradually begun to emerge. Recent advances in agentic AI provide new opportunities and challenges for insurance companies.

Equ 3: Manufacturing: Predictive Maintenance

$$T_f = \arg\min_t \mathbb{P}(f_t = 1 | x_t)$$

- T_f: Time-to-failure
- x_t: Sensor readings / operational data
- Use survival analysis or hazard models

6. Designing Scalable Architectures

The expression "you design for scale when you can afford to" may well summarize the dilemma of Software Architects regarding scalability. At the beginning of any project, with time-to-market being the top priority, if one cannot afford the cost of possible future redesign of core components, then designing them for scale is inevitable. In this scenario, poorly performing solutions are deployed, opening the possibility of inspiring technology commitment and entrenched funding. Instead of being often shallow technical details, like a caching infrastructure, such decisions could be of organizing the business functionality as loosely bounded components, connected only by designs like Events or Commands Bus queues, with the right level of automation and observability. Architectural styles helping in this direction are those based on Microservices, Event-Driven Organization, and Command Query Responsibility Segregation architectonics. Event-Driven architecture has been a key factor in business success and drives almost all investments in Digital Architecture.

During the SaaS era, Scalability was told to be by introducing persistence architectural components, like Data Lakes, or Data Warehouses. We argue that conformity is the real threat. High-Level Anatomy of Scalable Architectures describes how the semantics of our Data Products influence the design of such components. Enterprise Systems Data Heathering solutions, with which we are all used to work with, with elaborate costs and space-time response, have traditionally been to be observed in the not particularly much novel aspect of being "being designed around Tables," often imposing analytical deployment modalities which produce anyways non-realistic volumes due to the times of company data editing being visible only for snapshot or tracing Change Date Schemas.

6.1. Microservices Architecture Microservices is an architectural style that structures an application as a collection of loosely coupled services, which implement business capabilities. It is the evolution of service-oriented architecture, which used a heavier enterprise service bus. Microservices, on the other hand, use lightweight mechanisms, such as simple HTTP or lightweight message brokers, in a decentralized approach. Microservices overcome some of the shortcomings of service-oriented architecture by allowing enterprise applications to be composed of many independent services, each running a unique process and communicating through a well-defined, lightweight mechanism to serve a business goal or objective. Microservices move beyond the limitations of monolithic architectures, where an application and its services run in a single process and address the challenges of scale in the cloud. Microservices architecture is an effective way to build systems that can scale, adapt, and evolve over time.

Microservices evolution follows that of desktop applications into larger and larger monoliths before evolving back towards small, modular applications. Microservices in terms of scale and autonomy, however, are much lighter than service-oriented architecture. Microservices enable smaller teams to develop, deploy, and support services independently of each other. Increased deployment velocity is a primary motivation for adopting microservices. Microservice architectures use small services, developed and deployed using cloud-based resources, which can be independently developed, deployed, and executed. Each microservice typically implements one or two business capabilities and is developed and deployed independently using an agile approach, with a short time-to-market. The business capability is the key organizational element and microservices are typically developed around organizational boundaries.

6.2. Data Lakes and Warehouses Most organizations rely on either systems for the storage and management of data from disparate sources such as applications, sensor readings or transactions. Data lakes allow the storage of data in its raw, source format. Data lakes can be valuable for exploratory analysis use cases, such as customer segmentation using demographic data, channel interactions using unstructured data, and historical interaction analysis using time series sensor readings and transaction data. But because the data remain in their original, raw formats, extracting conclusions, creating reports, or serving dashboards can take a long time, as the necessary data preparation must happen every time a query is run.

Data warehouses, on the other hand, seek to remove the bloat of raw data and help distill key insights, aggregating and transforming the disparate data into meaningful features stored in a format optimized for efficient queries. In general, designing an ecosystem where hot data can be accessed through data warehouses and cold data serving exploratory tasks can coexist might be needed. Throughout our work, when referring to data management, we will abstract away from it in order to focus on the deployed solution needed for use cases with high operational overload.

Advancements in techniques, especially in Natural Language Processing enable an "agentic" approach to be taken with extracting views, preprocesses or feature creation that previously had to be annotated and scheduled by engineers or analysts. But care must be taken, as these systems can hallucinate enabling only a "human-in-the-loop" mode to be deployed. At the same time, recent advancements in foundation models place the information needed to parse or distill common data transformation tasks in pre-trained, frozen systems, which can enable the scalable deployment of services creating these specifications with limited supervision compared to an architecture acting as a critiquing user.

7. Conclusion

The emergence of Data and AI technology is disrupting the long-held centralized enterprise software product and platform design. The objective shifts to the design for a decentralized platform-as-a-product that draws not only from massive internal data and ML, but external data, and connects to an external ecosystem of partner products. This requires a rethinking of the long-standing design principles and playbooks used for enterprise software design. The evolution trajectory of Data Products provides lessons for new Data and AI Products. The product design has to break out of the traditional usability-centric interaction design, into the influence-centric agentic interaction design for data scientists and strategic peripheral users. In conclusion, we are entering the new age of a decentralized Internet Economy facilitated by the Data-Compute Hybrid Economy that builds upon the new Economics of Data. At the core of this new economy is the ability of companies to leverage both external and internal data for learning, decision making, and product innovation. To facilitate this process, the focus of enterprise software product design needs to shift from the traditional inward facing enterprise products, to the outward facing decentralized platform-as-a-product approach. We believe that by integrating findings and ideas from collective intelligence, ecosystems and multi-stakeholders, design, gazelles and collective entrepreneurship, and the new Economics of Data, we can start to imagine new enterprise software architecture and organizational economics. It has to be simplified and democratized yet powerful for a new generation of enterprise data products. It is a brave new world that is a portent of things to come.

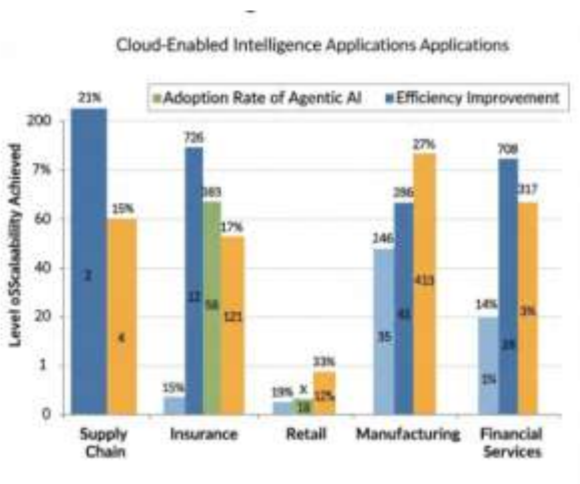

Fig : Designing Scalable Data Product Architectures with Agentic AI and ML: A Cross-Industry Study of Cloud-Enabled Intelligence in Supply Chain, Insurance, Retail, Manufacturing, and Financial Services.

7.1. Future Trends As Data Products evolve, so will the underlying Data Architecture Patterns, Gradients, Quality Metrics, and Data Pipelines we build and maintain on the journey from Data Ingestion to value delivery back to the Data Product's consumers. This will increasingly require fewer silos, shorter feedback loops, continual updating of long-lived Data Products, more complex Datasets being ingested, and increasingly more automated pipelines. Keeping Data available, accessing data using understandable terms and concepts, and bridging silos through cohesive schema design is called a Data Mesh Architecture. Simply defining a common glossary of terms that all domains must comply with will help for more coherent bridging efforts. Increasingly, with newer technologies, the Data Warehouse and Data Lakes which have mainly been evolving for the last 15 years will become irrelevant, where Data Pipelines will increasingly do more Work. Now, we either ETL/ELT into Data Warehouses or move the Data to Object Stores for consumption later. Newer Data Pipelines are changing this. Emerging Data Products have now enabled new methods of tapping chosen slices or subsets of Sensory and Entertainment Data from the Internet, rather than Bulk Downloads.

As a next step, it is imperative to find or evolve technology that eliminates the need for both redundant Data Ingest and copying Data to Shared Storage. In addition, once a Data Product is in use, it needs to be continually updated and improved for Quality, Latency, and Freshness. Last but not the least, the Business Applications using Data Products must request and access them to gain Value. Support of the Interactive

Interfaces, Alerts, and Webhooks become necessary to enable Business Processes to consume Data Products more regularly. Amplifying the need for organizations to hire Data Architects who can create Focal Points or SSO for Analytics.

8. References

[1] Kannan, S., & Yellanki, S. K. (2025). Synthetic Cognition Meets Data Deluge: Architecting Agentic AI Models for Self-Regulating Knowledge Graphs in Heterogeneous Data Warehousing.

[2] Sheelam, G. K., & Komaragiri, V. B. (2025). Self-Adaptive Wireless Communication: Leveraging ML And Agentic AI In Smart Telecommunication Networks. Metallurgical and Materials Engineering, 1381-1401.

[3] Motamary, S. (2025). Empowering Retail Oss/Bss Platforms With Agentic Ai And Scalable Data Engineering. Metallurgical and Materials Engineering, 1361-1380.

[4] Meda, R. (2025). Dynamic Territory Management and Account Segmentation using Machine Learning: Strategies for Maximizing Sales Efficiency in a US Zonal Network. EKSPLORIUM-BULETIN PUSAT TEKNOLOGI BAHAN GALIAN NUKLIR, 46(1), 634-653.

[5] Somu, B., & Inala, R. (2025). Transforming Core Banking Infrastructure with Agentic AI: A New Paradigm for Autonomous Financial Services. Advances in Consumer Research, 2(4).

[6] IT Integration and Cloud-Based Analytics for Managing Unclaimed Property and Public Revenue. (2025). MSW Management Journal, 34(2), 1228-1248.

[7] Annapareddy, V. N., Singireddy, J., Preethish Nanan, B., & Burugulla, J. K. R. (2025). Emotional Intelligence in Artificial Agents: Leveraging Deep Multimodal Big Data for Contextual Social Interaction and Adaptive Behavioral Modelling. Jai Kiran Reddy, Emotional Intelligence in Artificial Agents: Leveraging Deep Multimodal Big Data for Contextual Social Interaction and Adaptive Behavioral Modelling (April 14, 2025).