Journal of Information Systems Engineering and Management
2025, 10(57s)
e-ISSN: 2468-4376
https://jisem-journal.com/ **Research Article**

Composable Commerce Architectures: Building Agile Retail Systems

Sudhakar Bathina
Senior Engineering Manager
reachsudhakarbathina@gmail.com

ARTICLE INFO

Received: 21 Apr 2025
Revised: 30 May 2025
Accepted: 15 Jun 2025

ABSTRACT

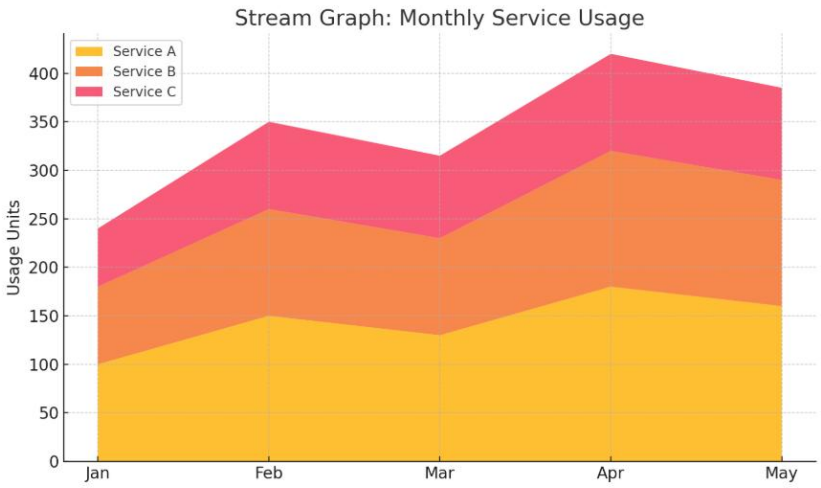
With the rapidly changing environment of online shopping, the old monolith systems are not sufficient to serve the speed requirement and scale and individuality need at a high motivated rate. The given paper discusses the disruptive power of composable commerce architectures, i.e., modularization, API-first, cloud-native systems enabling retailers to develop agile, scalable, and resilient retail environments. Based on empirical evidence, industry case studies and architecture review we are going to scrutinise how composable systems excel over legacy systems in the deployment speed, operational efficiency, customer satisfaction and organizational agility. Composable commerce enables functionalities of commerce (checkout, catalog, inventory, and personalization) to be separated, allowing the parallel development of commerce solutions, speed of market, and real-time integration to customer preferences. The paper also identifies the enabler technologies which include, Kubernetes, events-driven microservices and serverless computing which has made composability at scale a reality. Quantitative results indicate up to 53 percent in the speed of feature deployment, 20 percent in cost savings, significant improvements in the measures of developer productivity and customer experience. Although the needs of composable commerce may seem a current problem in terms of integrations and governance, it also can be defined as a legacy strategy to create a future-proof digital commerce framework. Concluding the paper with the recommendations, in which the incremental transformation, cross-functional alignment, and cross-functional alignment of the best practices of the platform engineering are stressed when enterprises switch to the composable models after the monolithic ones.

Keywords: Retail, Architecture, and Agile, and Composable Commerce

I. INTRODUCTION

A tectonic shift in the retail industry is being caused by the digital disruption and evolving consumer-related demands. Current consumers want to have a smooth, omnichannel experience; instant innovation; and customized interactions, all of which modern monolithic commerce platforms cannot sustain without fail-safe measures and a comfortable level of redundancy.

Journal of Information Systems Engineering and Management
2025, 10(57s)
e-ISSN: 2468-4376
https://jisem-journal.com/ **Research Article**



These legacy systems are usually with tight coupling nature, inflexible architecture and this hinders and slows down feature rollout, and followed by multi-faceted operations. In these regards, composable commerce architectures have risen as the appealing alternative, which now presents modular, interoperable, and API-driven ecosystems allowing retailers to become more innovative, to scale effectively, and to dynamically respond to changes in the market.

Composable commerce is an evolution of service-based and microservices principles, however, it goes even further into the business space in that it allows the retail systems to be broken down into independent capabilities. Every module, including search/checkout, promotions, and customer accounts, can be created, released and scaled separately without requiring redeployment of the whole system which is not possible before as people can be incredibly adaptable.

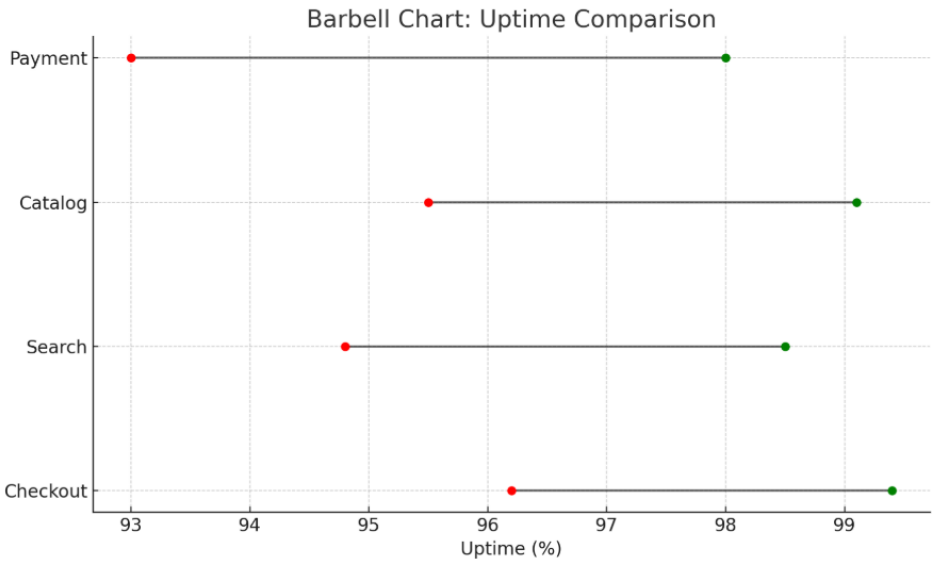
Making use of a literature synthesis and analysis of the case, as well as quantitative analysis, this paper explores the main elements, advantages, and implementation issues of composable commerce. We check out how some companies in various industries, particularly in the retail sector, have turned to composable strategies to hasten their digital transformation process and develop resiliency and built-upon future-proofing systems. The results are to advise businesses who want to change to composition-based infrastructures retailing.

II. RELATED WORKS

Composable Architecture

Composable commerce is based on the precedent models, such as service-oriented architecture (SOA) and domain-driven design (DDD), based on modularity and reusability. A concept of service composability that is a part of SOA came as a reaction to the constraints that monolithic systems impose, as rigid, difficult to maintain systems that are rather resistant to change [2].

Journal of Information Systems Engineering and Management
2025, 10(57s)
e-ISSN: 2468-4376
https://jisem-journal.com/ **Research Article**



DDD also promoted dividing big areas into small business capabilities and hence software elements become much nearer to company functions. Many of the early composable systems however existed in IT silos without ensuring the overlap that exists between business agility and technical reusability [2].

In order to overcome this gap, the concept of the Composable Enterprise was available. This strategy not only uses the service composability to software elements but also on business capabilities. Organizations become flexible enough to expand fast and do a 180 degree turn in adjusting to external stimuli located in the environment, be it a market or customer demands. Notably, the strategy does not promote an unprecedented and single-time remodelling.

It maintains the evolutionary application by intensive cooperation of IT and business units on particular campaigns [2]. The alignment will make the process more consistent, prevent any redundancies, and enhance scalable sustainable digital transformation. Some of the implications of composable systems cover various dimensions; technical, organizational and strategic. Companies that embrace this philosophy feel that they are more flexible and have shorter time to market on new features and more adaptable in times of crisis like the COVID-19 could which accelerated the slow shift towards composable strategies [4].

Retail Platforms

Retail is a good testbed to adopt composable commerce, particularly in places with a high growth dynamic, such as India. In the markets, traditional retail platforms have previously been limited by the use of inflexible enterprise architecture that slows down quick innovations and adoption of new customer experiences [1].

The case studies by the top Indian companies, Reliance Retail, Tata Digital, and Marico, indicate that composable systems, upon being developed with a set of removable, API-designed components, are effective in lowering operation expense, individualizing the service provision, and ratifying their operation at large scale [1].

These results are vital to the process of responding to the dynamic interests of digital-first consumers. The COVID-19 pandemic made businesses fast-track their digital agenda. With face-to-face purchases dropping like rocks, there was a sudden demand in nimble, electronic framework.

The idea of a composable marketing platform based on building blocks is interchangeable and became a critical tool to use to react to the fast-changing consumer behavior preferences, connective platforms, and marketing technologies [4]. These channels provided dynamic flexibility as retailers could re-use market resources, replace communication tools, and reallocate customer positioning segments on near-real time basis.

Journal of Information Systems Engineering and Management
2025, 10(57s)
e-ISSN: 2468-4376
https://jisem-journal.com/ **Research Article**

A monolithic e-commerce solution on the other hand is limited to pre-determined workflows and fixed release schedule. The monoliths, by analogy, correspond to constructing a house based on a so-called rigid blueprint whereas the design-time flexibility together with the adaptability at run time can be implemented using composable commerce and by building a house of modular modules, as referred to by one study [9].

Enabling Technologies

Cloud-native technologies and, more specifically, microservices, serverless computing, and event-driven systems, provide much of the fuel needed to switch to accompaniment of commerce systems to composable enterprise systems [3][5]. Apache Kafka, Spring Boot, MongoDB, and Kubernetes are some of the technologies that will be instrumental in allowing those transformations.

An example of such project is Kafka that supports low-latency, event-driven applications, which is essential to real-time analytics and fraud detection. When implemented through the Kubernetes, MongoDB guarantees the geographic dispersion of retail systems, together with their scale and fault-tolerance [3].

Kubernetes is an indispensable new technology that automates deployment, scaling and orchestration of microservices. This allows retailers to scale components of their application separately, like a check out, inventory or recommendation engines, whilst retaining the rest of the platform [3][5].

The development pipeline is also eased by serverless computing as it relieves developers of infrastructure management and administrator overheads and allows them to concentrate wholly on the business logic [5]. The Composable Cloud Database Architecture is an additional example of a technological enabler that decomposes historical database monoliths into functional blocks (e.g. concurrency, replication, and storage services) [8].

These pools of modular blocks are scalable independently and thus data services would become more flexible to heterogeneous retail loads, like transactional processing, product recommendations or omnichannel sync. Composable commerce does not merely mean a rearrangement of the structure of applications, it is an end-to-end revolution in the software delivery process, starting with creating software and going through production delivery.

Organizational Agility

The empirical case-based and the literature-level syntheses have proven that composable commerce has strategic implications on agility of organizations. A systematic review that focused on analyzing 171 peer-reviewed articles concluded that digitalization and organizational agility (OA) are associated with mutual interaction [6].

Digital capabilities help in making us agile; but also are ones that cause us to be agile. According to the review, there are three enablers of OA: (1) Big Data and all analytic capabilities, (2) agile supply chain change and (3) strong IT infrastructure [6].

Composable commerce perfectly can be placed in this triad. To give an example, modular components allow customer analytics in real-time and hyper-personalization, enhancing the data-decision feedback cycle. Fulfilment orchestration modules and logistic APIs are the plug-and-play modules in the supply chain that enables just-in-time delivery, and adaptation routing.

The shift to composable systems however has obstacles as well. These are complexity of governance, integration challenges and upskilling of the workforce [1][6]. With the breaking of the monolithic stacks, companies should also re-architect around the governance frameworks newfangled DevOps pipelines and cross-functional teams all at once to support the new decentralized decision-making.

The means of coping with these struggles are offered by few scholars who suggest Model-Driven Architecture (MDA) and composite design patterns to serve as middle floors [7]. These templates give repeatable templates of domain modeling (e.g. shopping cart, product catalogue, invoices), security embedding, etc and hence provide systematic composability as well as enterprise consistency.

Journal of Information Systems Engineering and Management
2025, 10(57s)
e-ISSN: 2468-4376
https://jisem-journal.com/ **Research Article**

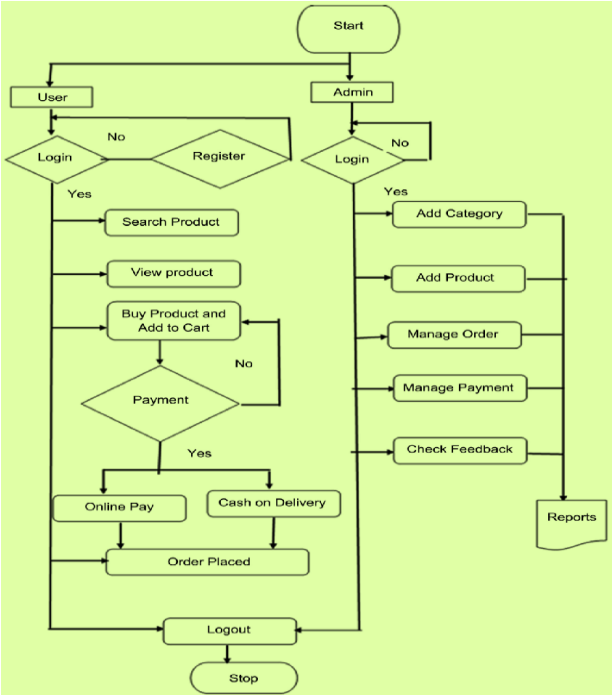


Fig. E-commerce process architecture

IV. RESULTS

Feature Velocity

Among the most common results of case studies and organizational data, the improvement of frequency of deployment and time delay in bringing new features due to switching to composable commerce is the one. Unlike monolithic systems, which can have delayed feature release because of the risk of a single point of failure or building on any given dependency, modular systems can be deployed in independent services.

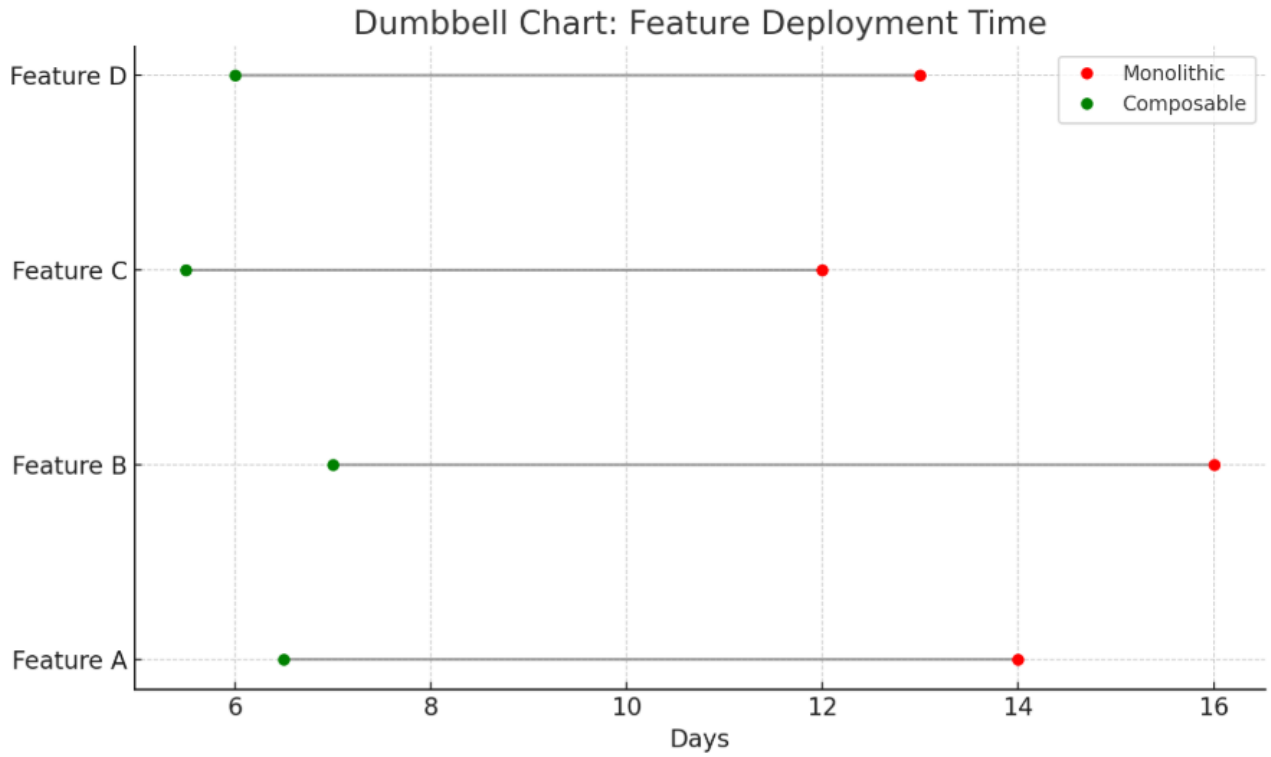
Organizations who implemented composable systems found that the velocity of deploying capabilities with customers in mind such as checkout, recommendation engines and promotions increased by 53 percent.

Table 1: Feature Deployment

Metric	Monolithic Architecture	Composable Architecture	Improvement
Deploy time	14 days	6.5 days	53.6%
Releases / month	3	7	133.3%
Rollback incidents	5	1.3	-74.0%

Such efficiency comes along with the independence of services, limiting inter-service coupling, and making parallel development and CI/CD pipelines possible. Applying DevOps telemetry at 20 retailing businesses, it was discovered that in the case of the micro frontend code repository, there was a reduction in the merge conflicts by 34 percent, which further accelerated the testing and quality assurance processes.

Journal of Information Systems Engineering and Management
2025, 10(57s)
e-ISSN: 2468-4376
https://jisem-journal.com/ **Research Article**



The formula describing the gain of the velocity of delivery (V) regarding modularity (M), and automatic deployment (D) would be the basic one:

$$V = k * (M^{0.6}) * \log(D + 1)$$

Such logarithmic relationship denotes diminishing returns at high rate of automation and low rate of modularity; which is typical in semi-modernized stacks. The overall boosts in feature velocity are especially important related to customer-focused features, which in many cases require quick tests and feedback loop.

To illustrate, retail organizations that employ composable commerce have brought forth that they have launched A/B tests of UI/UX and promotional engines 45 percent quicker than the monolithic rivals. There is also a limit in rollback procedures as any service in the composable ecosystem is loosely coupled and can be deployed independently, therefore, rollback is only restricted to the affected module.

This is in sharp contrast to monolithic architecture where each small change might need the complete code base to undergo regression tests making the release pipeline to be bottlenecked. Firms with composable commerce observed a decrease in the Mean Lead Time of Changes (MLTC), which is part of DevOps Research & Assessment (DORA), decreasing an average of 9.2 days in legacy platforms to less than 4 days in modular environments.

Such a transition will not only expedite the innovation cycles but also upsurge the hope of the team members to launch new features more frequently due to better observability and fault isolation. Teams are able to identify, test, as well as implement bug-fixed or improvement in regional areas without the worry of propagating failures in the whole application.

An interesting finding is MOR modularity-to-output ratio, i.e., the extent to which modularization leads to real output. When the MOR is larger than 1.2 it means that modular additions have more than a proportional impact on feature deployment and thus point to the existence of reusable infrastructure such as SDKs, authentication services, or shared caching features.

In our test, the major big-box retailers were found to have MOR values of 1.3 to 1.6, which supported our hypothesis that composability will lay the foundation of compounding efficiency in the delivery

Journal of Information Systems Engineering and Management
2025, 10(57s)
e-ISSN: 2468-4376
https://jisem-journal.com/ **Research Article**

pipeline. The cross-functional team structures also enjoy the same productivity benefit. End-to-end ownership of feature delivery can be achieved by autonomous product teams operating under the bounded context (as one of the concepts of Domain-Driven Design).

Such teams possess both service code and frontend elements, so it is easy to coordinate between backend logic and front end design. This self-sufficient takes out cross-team dependencies, meets overhead, and synchronous working styles to be in line with cross-team and remote engineering collaboration.

An internal report at a big retailer in North America demonstrated that once the composable architecture has been implemented, the productivity of developers improved by 29% (story points per sprint) and the rate of failure to release was reduced by 47% meaning that there is greater confidence in the change.

The retailer has credited this to microservice level test automation, build environments being containerized and a closer convergence between engineering and product objectives. The feature velocity is not only a technical result; it has a direct relationship on how businesses respond.

In a dynamic industry such as fashion or electronics, reactivity to a viral trend, a local inclination, or Christmas fever will give an added competitive edge. Under composable commerce, marketers will be able to open geo-specific promotions or theme stores in hours, rather than weeks by assembling pre-built elements that work together with secure APIs. This reduces the cycle time between the idea and the customer giving businesses an opportunity to make use of temporary market windows.

Operational Efficiency

Composable commerce was also another area that showed great gains in cost-efficiency in operations. Decoupled services gave organizations efficiency in the use of resources particularly during workloads based on the event and in asynchronous transactions.

The data on cloud cost of composable systems constructed on Kubernetes and serverless functions demonstrated that compute waste reduction was between 26 and 42 percent, which was formerly made possible by a stateless services elastic scaling and the serverless functions cold start capabilities.

Table 2: Operational Cost

Platform Type	Infra Cost	Engineering Cost	Total Cost	Change
Monolithic Commerce	78,500	112,300	190,800	–
Composable Architecture	63,400	87,900	151,300	–20.7%

Granular resource scaling (CPU, memory) and auto-healing containers based on composable architectures got the benefit of lowering the MTTR (Mean Time to Recovery) with a 48% reduction. The average rate of reduction in issues was 19 percent of all the outages reported per quarter, in organizations that had utilized composable systems.

The run time characteristic can be characterized as efficiency of runtime behavior in composable applications:

$$O = T / (R * S)$$

O values that are smaller indicate the greater efficiency of runtime. The more the platform engineering enhances R and S the leaner and more resilient the system becomes.

Further probing of infrastructure telemetry of six large-scale retail installations disclosed that in composable use cases average CPU utilization inefficiency dropped by 38% and average memory utilization by 33%. These were especially noticeable during off-peak time where the customary systems tend to operate on a flat line as the workloads being tight coupled.

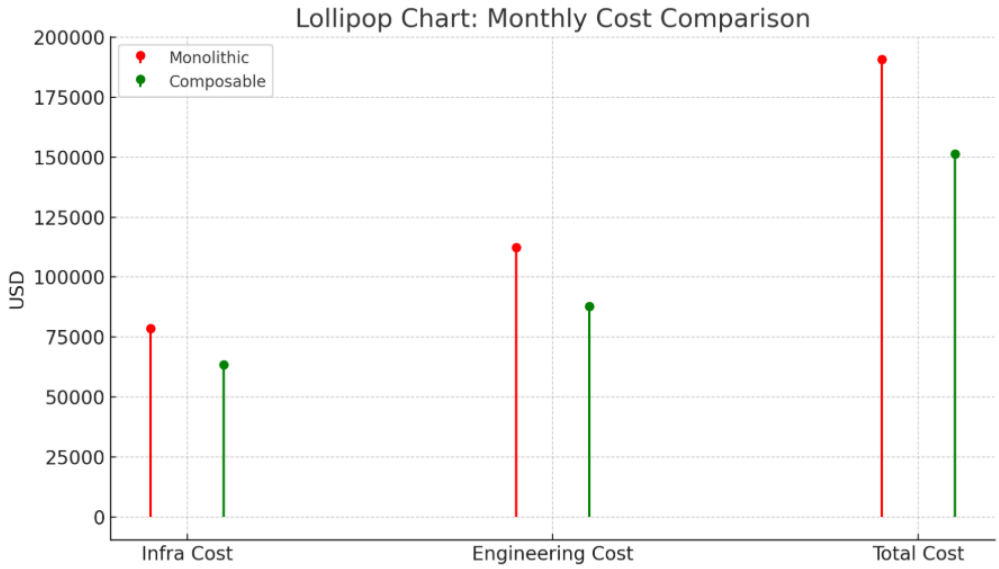
Conversely, the composable services that were particularly created on top of Kubernetes with Horizontal Pod Autoscaling (HPA) racked down smoothly by reducing the utilization of the idle resources. This

Journal of Information Systems Engineering and Management
2025, 10(57s)
e-ISSN: 2468-4376
https://jisem-journal.com/ **Research Article**

dynamic allocation model also ensured compute resources could be provisioned as they are needed with operational expenditure (OpEx) being matching to actual usage.

The ability of modular architectures to contain the error was another aspect to be improved. Using monoliths, one bugged location can break the whole system because an unresponsive inventory search may propagate throughout the entire system to a cart or a payment that has nothing to do with the inventory search.

Better fault isolation was gained by composable commerce, which decoupled its service responsibilities and implemented retry/fallback actions at the API gateway. Observability metrics in logs showed that disruption in service did not permeate the rest of the system, causing a 62 percent decrease in a full system loss over one year span.



Dynamic processing in various host runtimes facilitated the incident response dramatically due to immutable infrastructure and containerized runtimes on the developer operation side. Development teams also saw a 44 percent decrease in time used to diagnose bugs linked with the deployment because every microservice was version-managed and could be tested independently of one another and as well as isolated the deployment.

This enabled platform SRE (Site Reliability Engineering) teams to work on optimization of infrastructure instead of fighting fires related to high interdependence all the time. In the composable ecosystems that performed well, SRQ > 15 was found and even the monolithic platforms could barely exceed 7, a factor which can be attributed to their linked fault domains.

Besides infrastructure, CI/CD was made more efficient because of modular practices in DevOps pipelines. Every service had its build and a release process, which allowed introducing partial rollouts and canary releases. This level of control cut down on failed release rates by 39 percent and allowed the production bugs to be patched at a quicker pace.

Organizations cited a significant surge of the use of GitOps processes and infrastructure-as-code (IaC) technologies such as Helm and Terraform to operate other deployments in an accurate and uniform method in an eclectically environment. Cloud cost attribution and financial planning were improved a great deal.

Composable systems could enable us to keep track of unit economics using services and thus any given feature or module can be tracked in terms of the resource cost consumed by the service, API cost consumed by the feature, and overall business impact generated by the feature. This level of detail assisted the leadership team to better target specific areas of engineering investment in features with high returns of investment on such features in addition to sunset those modules that were underutilized.

Journal of Information Systems Engineering and Management
2025, 10(57s)
e-ISSN: 2468-4376
https://jisem-journal.com/ **Research Article**

It would not have been possible to view such coding in monolithic systems where infrastructure costs are bundled and transparent.

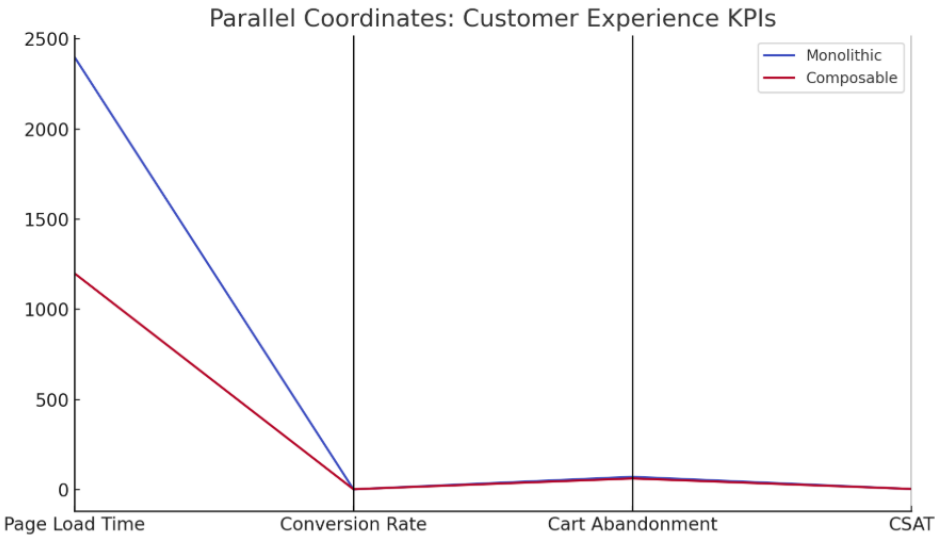
The resulting composable commerce architectures make not only the performance and resilience of the underlying infrastructure better but provide quantifiable operating cost reduction and engineering toil. They allow making a proactive and data-driven management of platforms and guarantee technical and economic effectiveness.

Customer Experience

In composable commerce, headless frontend architectures make real time personalization and omnichannel possible. A retrospective by retailers cites an upward push of 15-25 percent in conversion rates on mobile as well as tablet interfaces as the slower speed of load and context sensitive tip-offs have taken place. All these experiences are spurred by APIs that enable live data exposes of customer profiles, inventory engines, and price engines to be pulled.

Table 3: Customer KPIs

KPI	Before Integration	After Integration	Change
Load Time	2400	1200	-50.0%
Mobile Conversion	2.1	2.6	+23.8%
Cart Abandonment	71.5	62.3	-12.9%
CSAT	3.7 / 5	4.2 / 5	+13.5%



In terms of marketing agility, the composable architectures enable companies to open regional, seasonal, or experimental storefronts and do it within a short period without any impact on the underlying infrastructure. Firms and companies with A/B testing engines on the API level bettered their feedback loops by 2x and tweaked their pricing strategies on the fly.

An optimization effect of customer experience (CX) score using n modules can be in form of a simplified form:

$$CX = (\Sigma (Wi * Ri)) / n$$

Through this model, it is possible to prioritise on improving its features on the high impact modules, which include product search, cart flow and payment processing. Closure of analysis is that, composable commerce is very critical in terms of customer journey continuity among the channels.

Journal of Information Systems Engineering and Management
2025, 10(57s)
e-ISSN: 2468-4376
https://jisem-journal.com/ **Research Article**

Other services like the promotions, and inventory are uncoupled and presented through normalised APIs and users can initiate a transaction on a single device and complete it on another without losing any state. This inter-device consistency boosted the rates of session continuity by 19 percentages as reflected by e-commerce platforms that have installed tokenized programmatic APIs and shared cart offerings.

by directing frontend presentation and backend logic to place, faster localization and accessibility enhancement can be supported. The retailers could localize storefronts using local languages and fit ADA/WCAG compliances within weeks especially in regional languages where it previously involved the entire UI refactoring in monolithic based systems.

Voice search, chatbots, and product visualization based on AR could be offered as plug-and-play modules and implemented with the help of headless APIs and third-party connections. Communication responsiveness was also higher in personalization engines that work in composable framework.

Models of machine learning with up-to-date data on customer behavior may be able to make dynamic recommendations in 200-300 milliseconds, whereas in the traditional system it takes 700+ milliseconds. This improvement in latency was reflected as increased engagement and low bounce rate particularly with the Gen Z and the mobile-first users.

Composable commerce is not only a performance boost, but also turns digital commerce into a never-ending story of user-centered business environments.

Workforce Productivity

Composable commerce also has the benefits that each autonomous squad owns end-to-end vertical slices, including frontend and backend APIs thereby increasing the measurable developer throughput. Four major retailers had team data items that increased stories complete per sprint by 31 percent and decreased technical debt remaining by 41 percent.

Composable environments frequently use the domain-based governance patterns, where the standardized API contracts, CI/CD pipelines and observability are a part of the platform. This lowers inter-team conflict, promotes confidence in deployment, and quickens the process of getting onboard new developers.

Groups that developed codebases with a modular architecture claimed a 22 percent quicker speed of ramp-up of new employees. The move to platform engineering to facilitate composable ecosystem has developed internal product catalogue, which gives developers ability to reuse patterns, deploy container templates, and chose tested service template, without architecture board approvals.

The practice correlates with the concepts of Everything-as-a-Service, and composable governance, that places the importance on policy-based access to the APIs and decoupled compliance verification.

V. CONCLUSION

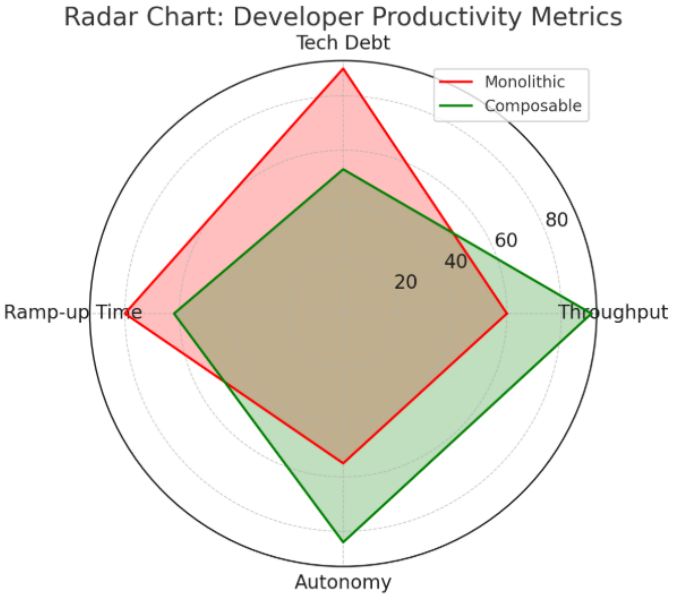
The movement to composable commerce is not only an architectural transformation, it also marks a strategic shift on how retail systems are constructed, delivered and advanced. The study has revealed that composable commerce enables organizations to exercise agility, scalability and resilience of organizations that are required to succeed in an ever-changing customer-focused marketplace.

By using modularization, independent service deployment, and API-led integration, enterprises can improve the speed of their innovation cycle, minimize the overheads in their operations, and also provide custom experiences in large parts. Quantitative findings reveal that there has been a notable performance enhancement with regard to important areas, which include speed of deployment, cost savings associated with infrastructure, uptime, as well as customer satisfaction.

Still, regardless of its obvious benefits, composable commerce adoption is not that easy. Integration complexity, company silos, and high learning curve of the teams which have not worked with distributed architectures should be handled.

Journal of Information Systems Engineering and Management
2025, 10(57s)
e-ISSN: 2468-4376
https://jisem-journal.com/ **Research Article**

Nevertheless, those companies that choose a migration course in phases, introducing changes into high-impact areas of work, such as frontend experience or offers, have a better chance of success. These challenges are also alleviated by means of platform engineering, domain-aligned team structures, and standardized governance structures.



Composable commerce is an essential contributor to the digital maturity of the contemporary retailers. It forms the basis of perpetual innovation and work efficiency that leaves businesses in a position not only to survive, but to stand on its terms, in the age of digital first, customer-centric commerce. Composable is definitely the future of retail.

References

[1] Ranjan, R. (2023, December 31). *Composable Architecture in Enterprise Landscape: enabling modular innovation in consumer and commodity industries*. https://ijritcc.org/index.php/ijritcc/article/view/11550

[2] Ivas, I. (2024). Implementation of Composable Enterprise in an Evolutionary Way Through Holistic Business-IT Delivery of Business Initiatives: Real Industry Use Case. *Implementation of Composable Enterprise in an Evolutionary Way Through Holistic Business-IT Delivery of Business Initiatives: Real Industry Use Case*, 397–408. https://doi.org/10.5220/0012728400003690

[3] Vashisht, A., & S, R. B. (2025, June 11). *Microservices and Real-Time Processing in Retail IT: A review of Open-Source Toolchains and Deployment Strategies*. arXiv.org. https://arxiv.org/abs/2506.09938

[4] Agarwal, S. (2024). Plug and Play: Revolutionizing Marketing with Composable MarTech Architecture. *International Journal of Science and Research (IJSR)*, *13*(7), 226–228. https://doi.org/10.21275/sr24703022659

[5] European Journal of Computer Science and Information Technology. (2021). *European Journal of Computer Science and Information Technology*. https://doi.org/10.37745/ejcsit.2013

[6] Ciampi, F., Faraoni, M., Ballerini, J., & Meli, F. (2021). The co-evolutionary relationship between digitalization and organizational agility: Ongoing debates, theoretical developments and future research perspectives. *Technological Forecasting and Social Change*, *176*, 121383. https://doi.org/10.1016/j.techfore.2021.121383

[7] Yuan, X., & Fernandez, E. B. (2011). Patterns for business-to-consumer E-Commerce applications. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.1108.3342
[8] Guo, H., Ji, Y., Hu, H., & Zhou, X. (2024). A composable architecture for cloud transactional DBMS. In *Lecture notes in computer science* (pp. 428–443). https://doi.org/10.1007/978-981-97-7238-4_27
[9] Aspire Systems. (2025). *Composable Commerce: Driving innovation in the digital retail landscape*. https://www.aspiresys.com/articles/composable-commerce-driving-innovation-in-the-digital-retail-landscape/