KIBO®

//OLR

Through the Lens of History:

Examining the **Monolith** vs **Composable** Debate Through Time

Do technology trends repeat every generation? It would seem that the monolith versus composable debate is exactly the same as at the birth of eCommerce, before the evolution of monolithic solutions.

The history of technological evolution often unfolds in a pattern of familiar debates and choices. Today's discussions on software architecture, particularly the contrasting approaches of monolithic and composable architectures, bear a striking resemblance to the challenges faced during the inception of eCommerce.

The Birth of eCommerce: A Journey Through Architectural Choices

In the early days of eCommerce, businesses embarked on a quest to establish their digital presence. With aspirations to tap into the burgeoning online market, critical decisions had to be made regarding the underlying technology stack. This era witnessed the emergence of architecture, where all components of an application were tightly integrated into a single, monolithic system.

The monolith approach seemed promising at first glance, offering simplicity and ease of development. However, as eCommerce platforms expanded in complexity, inherent limitations became increasingly apparent. Changes to one part of the system necessitated redeployment of the entire application, leading to longer development cycles and scalability challenges during peak traffic periods.

The Rise of Composable Architectures: Learning from the Past

Fast forward to the present day, and we find ourselves at a crossroads reminiscent of the early eCommerce era. The monolith vs composable debate has resurfaced, reflecting a desire for more agile, scalable, and resilient software solutions.

Composable architectures, inspired by the principles of microservices, offer a compelling alternative to the traditional monolithic approach. By breaking down applications into smaller, independently deployable services, composable architectures provide greater flexibility and scalability. Each service can be developed, deployed, and scaled independently, leading to faster innovation and easier integration of new technologies.

Drawing Parallels Between Past and Present Trends

As we reflect on the evolution of technology architectures, it becomes evident that history often repeats itself. The challenges faced by early eCommerce pioneers mirror the dilemmas confronting modern-day software developers. The monolith vs composable debate transcends time, serving as a reminder of the cyclical nature of technological trends. Let's take a look at the fundamentals from a business rather than a technical perspective:

Aspect	Monolith	Composable
Architecture	Monolithic architecture refers to a single, unified application where all components are tightly coupled and deployed as a single unit. Changes require redeployment of the entire system.	Composable architecture follows a microservices approach where various independent services are built and deployed separately. Services communicate through APIs.
Scalability	Scaling can be challenging as the entire application needs to be scaled up or down together, leading to inefficient resource utilization and difficulties in handling traffic spikes.	Inherently scalable as individual services can be scaled independently based on demand, allowing for better resource utilization and more efficient traffic handling.
Flexibility	Lacks flexibility as changes to one part may require modifications to other parts, leading to longer development cycles and increased risk during implementation.	Offers greater flexibility as individual services can be modified, replaced, or updated without affecting the entire system, facilitating faster innovation and easier integration.
Maintenance	Maintenance and deployment can be complex, especially as the application grows larger, requiring extensive testing to ensure changes do not impact other parts.	Maintenance and deployment are typically easier since changes can be isolated to individual services, allowing for faster deployment cycles and easier rollback.
Technology Stack	Often relies on a single technology stack for the entire application, limiting flexibility and the ability to use the best tools for each component.	Allows for greater flexibility in choosing technologies for each service, enabling developers to select the best tools for the job, leading to more efficient solutions.
Development Team	May require larger development teams since developers need expertise across the entire application stack.	Can enable smaller, more specialized development teams since each service can be developed and maintained independently.

Propelling Forward: Unified Architecture

As technology trends evolve, so must our approach to eCommerce architecture. Leveraging modular architecture enhances customization capabilities, enabling the rapid development of new functionalities like loyalty programs, customer attribution, etc. with faster ROI.

Kibo propels eCommerce forward, meeting all MACH requirements while also boasting the ability to develop additional applications swiftly, thanks to its unified architecture. The unified architecture offers a logical solution: by grouping services into cohesive modules, it effectively reduces complexity within applications. This ensures that critical business functions operate reliably and sustainably over time, providing a robust foundation for adaptation and growth.

Deployments and upgrades are also significantly accelerated too. APIs evolve and new ones become available, in true composable architectures with thousands of microservices, APIs are backward compatible so technology providers can deploy changes without affecting the applications, using the underlying services.

Invariants: Unchanging Principles in eCommerce Architecture

On top of that, irrespective of the architecture, some things will never change when considering your commerce platform, let's look at some of these:

- Clear and unambiguous requirements are essential, covering both functional and non-functional aspects.
- Validated design ensures alignment with requirements and facilitates user testing.
- Robust testing practices, including end-to-end and regression testing, validate system integrity and user experience.
- Version control is crucial for managing software updates and ensuring compatibility.
- These foundational principles are vital for eCommerce success, regardless of architectural differences.
- Analyse and report on the objective performance of the application and make decisions based on facts, so that you evolve in a method of constant change using AB testing techniques to ensure that you optimised your application.

Conclusion

As the saying goes, ‘what goes around comes around,’ reflecting that while monoliths were popular at a particular time, we have evolved towards the individual services that make composable architectures compelling.

While both eCommerce monolithic and composable architectures have their advantages and disadvantages, at the moment, composable architectures offer greater scalability, flexibility, and ease of maintenance, making them well-suited for modern eCommerce platforms that need to rapidly adapt to changing business requirements and technological advancements.

Composable can significantly reduce the total cost of ownership by allowing faster development: easier to test, build and deploy. Support is simplified because you have clear APIs against which you can test the application and address customizations in a timely manner.

However, it’s important to acknowledge that the choice between monolithic and composable architectures ultimately depends on the specific needs and circumstances of each business. While composable architectures provide significant benefits in terms of adaptability and scalability, they also come with increased complexity and potential integration challenges. Therefore, businesses should carefully evaluate their objectives, technical capabilities, and resource constraints before making a decision on the architectural approach that best aligns with their goals.

About Kibo

Kibo Commerce is a composable commerce platform for retailers, manufacturers, distributors, and wholesalers who want to simplify the complexity in their businesses and deliver modern customer experiences. Kibo is the only modular commerce platform supporting experiences that span Order Management, eCommerce, and Subscriptions. Companies like Zwilling, Ace Hardware, Boscov’s, Nivel, and REEDS Jewelers trust Kibo to bring simplicity and sophistication to commerce operations and exceed customer expectations.

KIBO®

About OLR

OLR is a specialist systems integrator of Oracle Retail and Commerce Solutions, headquartered in the London with operations in the UK, Portugal, USA, Mexico and India. OLR specialises in providing a full lifecycle of services for end-to-end Oracle Retail Centric Solutions for Stores, Merchandising, Commerce & Application Managed Services. OLR has a proud track record of delivery of Oracle Retail solutions to some of the most well-known global retail brands.

//OLR