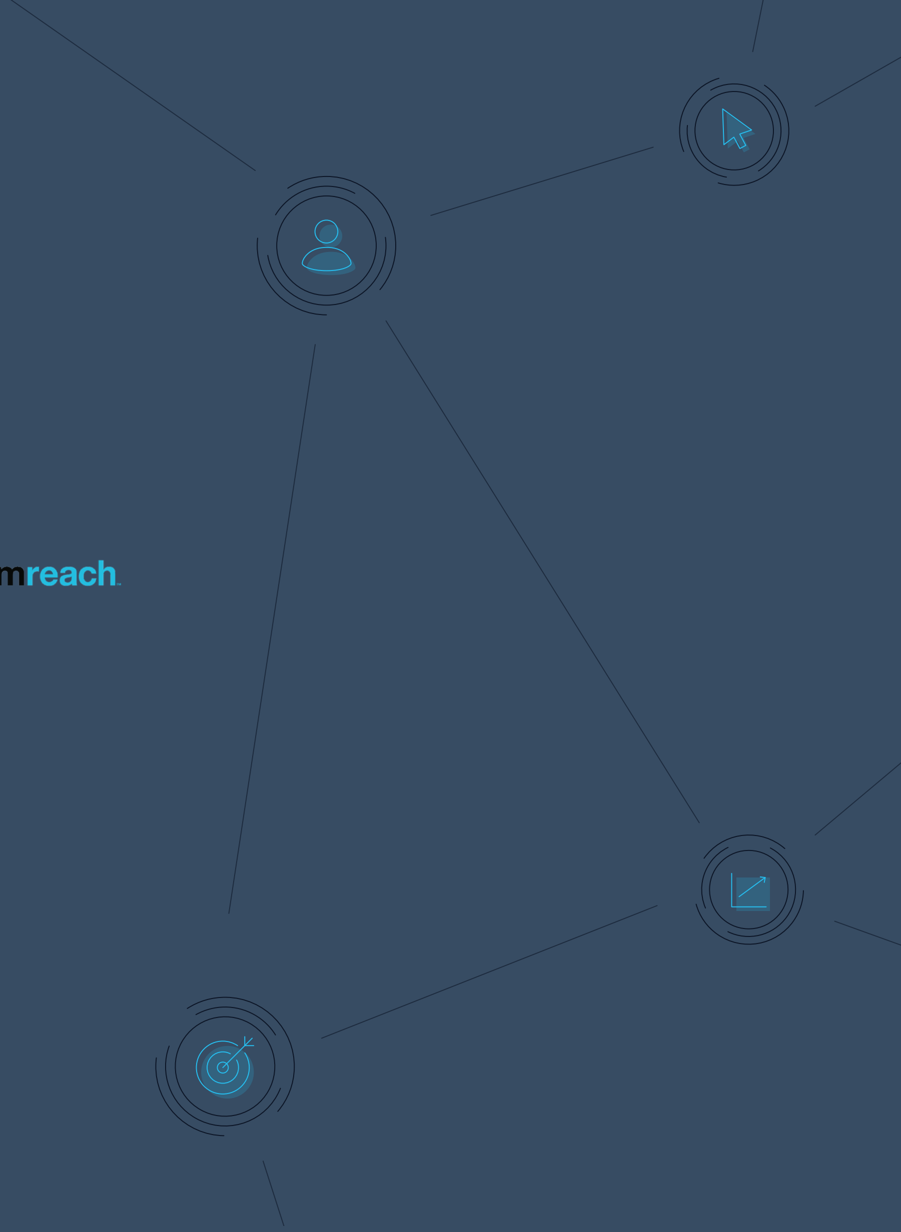
Brought to you by:

ZAELAB

In partnership with:

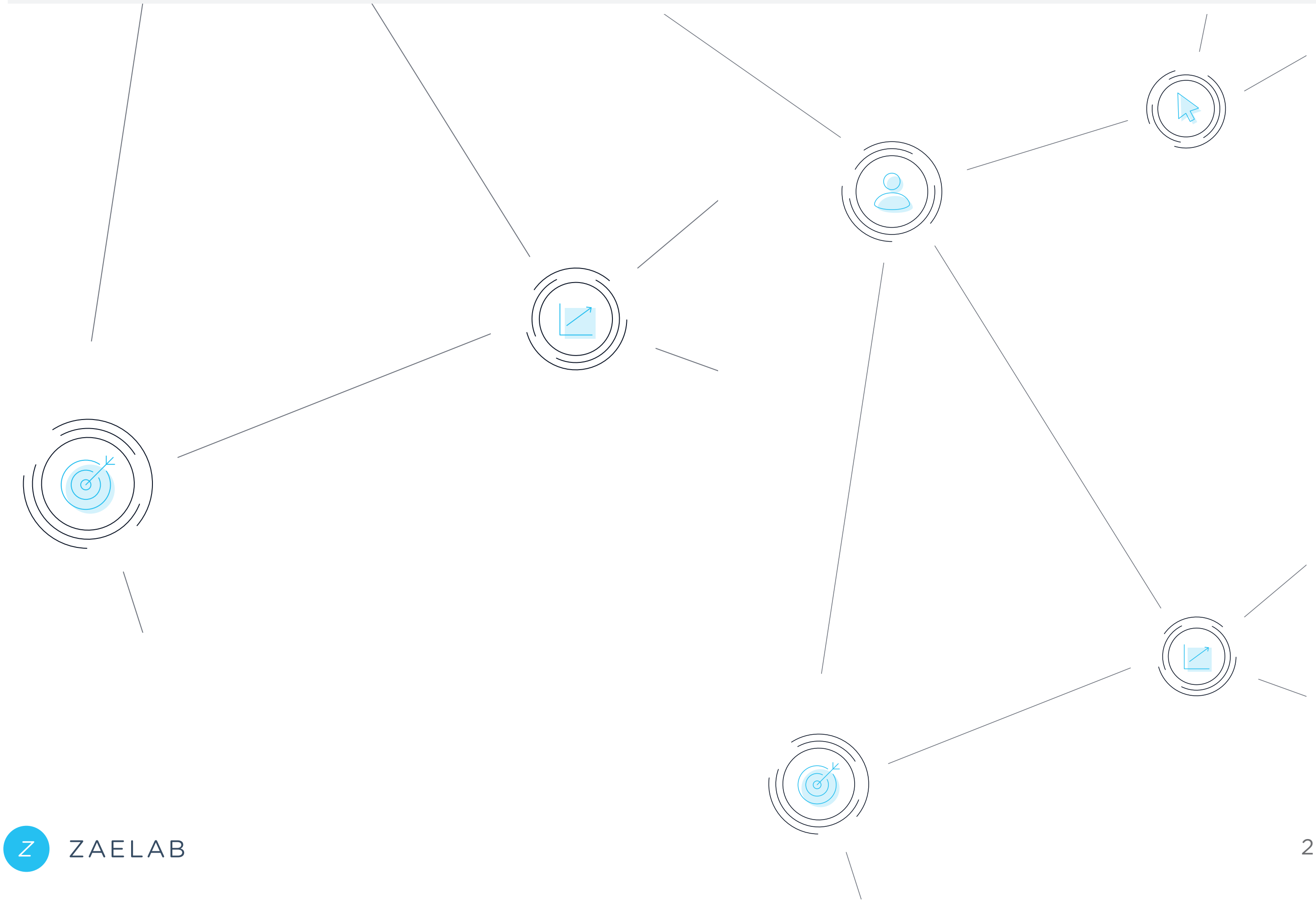
commercetools *Next generation commerce* + bloomreach



From Monolith to Microservices:

Learn how enterprises are boosting their speed and flexibility by shifting commerce architectures.

Table of Contents



Executive Summary

Until recently, digital commerce platforms were almost exclusively deployed by monolith architectures. While this type of framework remains a viable and revenue-producing platform for many enterprises today, it also poses challenges.

Heavy, feature and functionality rich, and slow, monolith commerce platforms provide too much of what is unneeded. Moreover, frontend templating logic is geared towards classic, browser-based ecommerce sites and webshops. As a result, monolith frameworks are not nimble enough to deliver what is required within a reasonable time, or built to scale.

The business driver for increased enterprise flexibility is the customer and experience-led commerce. With monolith architectures, out-of-the-box is a gray area and integration seems to consume implementation - often at the sacrifice of the customer experience. Businesses are seeking out new ways to increase their agility in order to meet customer needs.

Contrary to a single monolithic application that handles all business logic and offers all features, a microservice-based approach encapsulates each business capability into individual services and lets them interact with each other. The result: reduced software complexity and increased speed and flexibility.

A business' commerce platform should be a revenue-enabler and not its handicap when it comes to quick response to market demands. Learn why enterprises are migrating onto microservices architectures - even after implementing a monolith platform - to increase speed and flexibility.

Introduction

When enterprises implemented ecommerce platforms in the early 2000s, the monolith architecture delivered. It was one, lightweight and unified application. The typical, and most essential features and functionalities were pre-packaged. Requirements were easy to meet. Omnichannel was a mere twinkle in one's eye.

Over time, as one would expect, things changed. Customer and market demands became more complex as delivery expectations turned urgent. Monolithic architectures took on new features and functionalities to meet changing needs, but in doing so, became slower.

Today, an enterprise can still "get it all" from a monolithic ecommerce platform with the caveat that it will take time. Implementations, upgrades, and new integrations are possible, so long as you have a month or two to spare. The problem is that for most leading enterprises, time is of the essence.

Innovation and speed are competitive differentiators. But how can enterprises deliver seamless digital experiences right now when monolith platforms are slowing business down?

Contrasting monolith are microservices. Unlike heavy and slow monolith architectures, microservices are small services that are individually developed and deployed. Communicating via APIs, microservices reduce software complexity, scale vertically, and - you guessed it - increase flexibility, speed and resiliency.

So marks a shift in ecommerce; enterprises are seeking out platforms that not only deliver on today's needs, but that are also designed to scale with the ever-evolving digital landscape. In fact, a recent Gartner study revealed that 56% of enterprise organizations have adopted microservices or have planned and budgeted to deploy API-based or headless commerce architectures.

This turn toward microservices is enabling enterprise agility. Customer demands can be met in days or weeks versus weeks or months, which ultimately better positions enterprises against competitors, Amazon and upstarts.

The question is, how can enterprises adopt microservices architectures - particularly if their ecommerce solution is currently being deployed via a legacy, monolith platform?

WHAT YOU'LL LEARN:

- Why monolith digital commerce platforms are not designed to scale with changing market demands
- The benefits of headless commerce architectures, and how they can be leveraged by enterprises
- Why migration doesn't have to be a "lift and shift" experience, and how to move from a monolith platform to a microservices architecture in increments
- How to identify the right time for the business to begin migration
- Why APIs, open architecture and headless commerce platforms are enabling best of breed digital experiences

SECTION 1

This Way or the Highway: Traditional Digital Commerce

I. This Way or the Highway: Traditional Digital Commerce

There is an incredible amount of pressure on enterprises to digitally transform or risk irrelevancy. Capitalizing on the opportunity to open a new revenue channel and deliver strong ROI, many businesses turn to ecommerce platforms to kick-start their digital transformation.

At one point, monolithic ecommerce platforms were the only option. Enterprises selected, implemented and constantly updated these large commerce platforms to digitize the organization, diversify their revenue stream and build an online business. And it worked for a while - until monolithic architectures began to miss the mark.

The problem is two-tiered. Monolith platforms operate on a tightly coupled frontend and backend. Any personalization or customization request involves editing the database, code, and front-end platform, which is extremely time-intensive. Moreover, developers must be mindful that customization changes do not infringe on future upgrades.

On top of this, customer want and need more - right now. With so many options available, if one digital experience fails to deliver, a customers can quickly to turn to a competitor. In response, enterprises are demanding more flexibility from their commerce solutions and a faster time-to-market than what is typically provided in out-of-the-box framework accelerators.

This is a huge challenge, and one that will only continue to evolve. As IoT devices and connectivity become more prevalent, businesses will need to adapt their solutions to support them. This can be difficult without the right architecture and supporting technologies.

MONOLITH PLATFORMS ALSO POSE CHALLENGES FOR DEVELOPERS:

Software complexity
A commerce application is constantly evolving to keep pace with changing demands. As a result, these applications become more difficult to maintain, and more challenging to understand by the developers working with it.

Lack of agility
With monolithic commerce solutions, teams are usually structured according to their individual functions, such as frontend, backend or database. When a request is made that affects all of these functions, tasks have to be shared across multiple team members. As a result, rolling out new features or entering new markets takes too long and leads to missed business opportunities.

Fragility
In a centralized architecture, the individual parts are highly coupled and dependent on each other. This results in a single point of failure. If one little cog of the clockwork does not work as planned, this can bring down the entire system. This disincentivizes individual developers from touching anything and taking responsibility for the system.

Inefficient testing
Because of the single points of failure present in these applications, comprehensive and repeated testing is crucial; if only one small part of the application is changed, it needs to be tested in its entirety. Because of the software’s internal dependencies, the effort involved in (automatic) testing and quality assurance rises exponentially.

Scaling issues
Instead of adding power the existing platform, with most monolithic applications, adding more servers - or scaling horizontally - is the only option. This in turn creates many other issues.

SECTION 2

The Highway: Microservices Architecture

II. The Highway: Microservices Architecture

To quickly respond to customers' needs and stay in front of competitors, organizations need a flexible software infrastructure. The answer: microservices.

Headless commerce decouples the frontend presentation layer from the backend platform / data layer through APIs. Whereas with a monolith, a frontend team uses one frontend application that is collecting data from the backend, microservices are decentralized. The frontend or "the head" - think user interface, social commerce, digital marketplace, IoT and more - can be updated or changed individually, without interfering with the backend. The backend - items like infrastructure, pricing, checkout and security - invisibly runs in the background. The frontend and backend speak to each other via simple API calls. So, to deliver target layouts for specific devices with the optimal UX, a custom UI layer calls on an API layer for content, data and standard ecommerce functionality.

Microservices enable enterprise nimbleness, and because they operate as decoupled components, businesses can select and customize the features and functionalities they truly require from their commerce platform. In other words, microservice architectures allow enterprises pay for what they actually use, and use what they actually need.

INTERNALLY, HEADLESS ARCHITECTURES ARE MORE ENABLING FOR DEVELOPERS:

Frontend and backend are individually scaled
Because the frontend and backend are decoupled and individually scaled, heavy frontend traffic does not impact the backend. And, developers can extend services to new touchpoints without needing to rely on the entire system.

Increased speed and resiliency
Without having to maintain a full stack software, new user interfaces can be implemented rapidly. Development is more efficient because teams can work in parallel and due to decoupling, changes can be made to the UI without having to test core backend logic.

Experimental freedom
Development experiments are not dependent on modifications to both the frontend and backend code, and therefore less risky. Developers can A/B test specific parts of a commerce site or build new Progressive Web Apps (PWA) without requiring the backend.

More customization options
In a headless scenario, multiple frontends connect to one API and one underlying system, making it easier to implement new touchpoints.

SECTION 3

The Highway: Headless Commerce Architecture

III. Moving from Monolithic to Microservices

Customizations to monolithic architectures are possible, but arduous. Enterprises that have deployed their commerce solution on a monolith platform are likely making software changes today via specific customization requests or version upgrades. In both of these situations, with monolith, the frontend and backend are tightly coupled, meaning the entire platform is impacted by these changes. The result: risky, lengthy, expensive and resource-intensive experiences.

For monolith commerce platform users, change equals time - lots of time. In some cases, a monolith commerce platform implementation may still be fresh, which was a process that likely took months and possibly years to accomplish. With this in mind, imagining the time and resources that would be required to migrate from a monolithic to microservices architecture can seem cumbersome.

Here's the good news: moving from monolithic to microservices is fast and incremental.

Unlike moving one monolith platform to another monolith platform, where an entire "lift and shift" process is executed, moving to microservices can be executed in stages.

Migration in Increments

Imagine this: you are building a new house for you and your family to move into – in the meantime, you are all living in a hotel, which is obviously quite costly. Instead of waiting for the entire new home to be built, you'd like to move in when there are enough components available in the home for your family to live on. So, you prioritize the rooms that are built, ensuring that core components will be available first: kitchen, bathroom and two of the bedrooms. Once ready, you can move into the home, stop spending money on the hotel, and continue building the other rooms in the house sequentially. This is precisely how a monolith to microservices migration is conducted.

Migration via the Strangler Pattern

Migrating a monolith architecture to microservices involves identifying the business domains or functions that should be taken over by a microservice. From there, a new microservices capability can be introduced and replace the corresponding piece of the legacy system. Over time, your commerce platform will become more and more independent.

This method is called the strangler pattern. A concept designed by Martin Fowler, the strangler pattern begins by introducing a new microservice, which is built and introduced completely independent from the monolith; eventually, the microservice takes over the functionality that was originally delivered by the monolith. This pattern application reduces risk as individual components are built and implemented – the microservice only takes over the component of the monolith once deemed fully evolved and stable.

Migration to microservices can be done swiftly but requires expertise. Working with a digital advisory throughout all stages of migration ensures that the process is executed quickly and with accuracy.

Below is a proprietary six-step migration process offered by a digital advisory. It includes a current platform evaluation and audit, a migration blueprint, infrastructure and management system establishment, and the continuous innovation of breaking down the monolith, implementing microservices and ultimately removing monolith dependencies to retire the old platform over time.

<<< CONTINUOUS INNOVATION >>>

01	Evalution, Audit & Recommendations	Business audit, digital maturity assessment, examination of monolith platform including areas of performance and limitations. Insights delivered from evolution and audit. Next steps; if migration makes sense for your business, platform and timeline recommendations delivered.
02	Migration Blueprint	Should migration to microservices suit the business, migration blueprint provided; includes proposed process for breaking down monolith and sequential order for implementing microservices.
03	Infrastructure, Continuous Innovation Cycle & APIs Established Monolith Decoupled.	Required infrastructure, processes and management systems established for migration. Monolith decoupled and split into small databases aligned with applications.
04	Microservices Implemented	Beginning with core business capabilities, microservices capabilities introduced independent of the monolith.
05	Microservice takes over Monolith Capability	Once microservices deemed stable, it takes over the corresponding monolith capability.
06	Retire Former Monolith Code	With traffic directed to microservice and monolith dependency eliminated, retire old monolith functionalities, features, code.

SECTION 4

Identifying the Right Time to Migrate

IV. Identifying the Right Time to Migrate

Let’s recap. Enterprises are recognizing that microservices are a more scalable architecture for deploying commerce solutions, especially given the ever-evolving digital landscape. If an enterprise is already operating their commerce solution on a monolith platform, it is a viable option to migrate towards a microservices architecture. Moreover, doing so is not a “lift and shift” experience - it can be done incrementally.

This leaves one question: how do you identify when it is the right time for your organization to migrate from monolith to microservices?

The reality is that an enterprise can migrate from monolith to microservices at any time. Because the process is incremental, there is no downtime, flexible deadlines, and little to no risk. However, there are some instances when making the case internally to shared project stakeholders is easier than others. Here are three indicators that now may be the best time for your enterprise to migrate to microservices:

1. **Your business is committed to continuous innovation**
   It is arguable that implementation is never truly complete; most businesses are committed to addressing market demands and continuously innovating to stay competitive. And that’s the challenge: ongoing implementations – be it an MVP, launching secondary components, or upgrading to the latest version of the platform to access new features and functionalities – are expensive and time-intensive.

   Market demands are constantly changing, and it’s not enough for the technology to be capable – it must be available immediately. Microservices allow you to address individual components of your commerce platform at any time. Therefore, if your business wants to continuously innovate at a more rapid pace, adopting a microservices architecture fuels flexibility.

2. **Your current monolith platform requires an arduous upgrade**
   Monolithic commerce platform version upgrades are monotonous, and sometimes an ultimatum. Often monolith platforms will fail to support its users on outdated versions. New features and functionalities are only available on the latest version, and this requires lifting and shifting the entire software in order to access that one feature that the business would like to leverage.

   In addition to being expensive, monolith upgrades can also be disruptive to users. Sometimes a new version requires a high learning requirement across the entire platform, resulting in lengthy implementations. This is not a scalable solution.

   With microservices upgrades, it is at the individual component level. Companies are never required to replace the entire monolith system, which means microservices upgrades present costs-savings and rapid implementation timelines.

3. **You need to make enhancements to the customer experience**
   In the era of customer-centricity, every change is driven by customer behavior and demands, and meeting these needs rapidly is a competitive differentiator. On monolith platforms, customer experience enhancements are possible, but require time because the architecture simply cannot support a business's specific use cases, often due to misalignment between complex demands and lacking functionality.

   With microservices decoupled frontend, applications are smaller and independent, making enhancements to the customer experience quickly attainable. Moreover, migrating to microservices-based platforms opens the doors to other API-based customer experience tools, like a Digital Experience Platform (DXP).

   A DXP is an integrated software framework for engaging audiences across an array of digital touchpoints. As your customer touchpoints inevitably grow, your data can seamlessly flow from multiple sources to create a fully connected and customer-first experience. Using a DXP in tandem with a microservices-based commerce platform presents multiple benefits:

   - **Connectivity**
     Digital Experience Platforms act as an API orchestration and aggregation layer. This easily connects with third party and legacy solutions, meaning DXP is the ideal experience layer to use when migrating from monolith to microservices.

   - **Advanced Personalization**
     A DXP uses advanced algorithms to drive a holistic personalization. It enables merchandising and search teams to create and test intelligent, personalized and revenue-driving customer browse, search and landing page experiences by pairing AI with smart, intuitive merchandising tools and analytics.

   - **Consistency**
     Great customer experiences are meaningful, and fueled by consistency across content, code and presentation. DXP ensures these elements are kept consistent across all channels.

Conclusion: Time to Transition from Monolith to Microservices

At one time, monolithic architectures fully enabled digital transformation and digital commerce demands. But as technology and customer needs continue to evolve, the pressure to innovate faster and deliver on requirements is increasing.

For many enterprises, monolithic platforms cannot sustain digital transformation in the long run. As more technologies and systems are being developed and adopted, alongside the changing digital landscape, headless architectures are positioned as the most effective way to enable rapid transformation.

Migrating from monolith to microservices is an efficient and incremental way to begin delivering improved digital experiences. Businesses don't have finish lines. If your enterprise has invested in a monolithic commerce platform, but is interested in boosting its speed and flexibility, connect with an expert to get an audit and assessment about what migrating to a microservices architecture can look like at your organization.

About Zaelab

Zaelab is a leading digital advisory and solutions company. Through continuous innovation, Zaelab removes the complexity and friction of digital platform implementation and operations. Zaelab give its clients best-in-class tools and insights so that they can deliver exceptional customer experiences.

About commercetools

commercetools is a next-generation software technology company that offers a true cloud commerce platform, providing the building blocks for the new digital commerce age. Its leading-edge API approach helps enterprises create brand value by empowering commerce teams to design unique and engaging digital commerce experiences everywhere – today and in the future. commercetools' agile, componentized architecture improves profitability by significantly reducing development time and resources required to migrate to modern commerce technology and meet new customer demands. It is the perfect starting point for customized microservices.

About BloomReach

BloomReach Experience (BRX) is a smart, API-based DXP. It combines advanced search and merchandising capabilities with a super CMS, giving marketing, merchandising and IT teams the tools to create engaging, personalised experiences for each visitor across all channels. BloomReach is a Strong Performer in The Forrester Wave for WCM 2018 and Visionary in the Gartner Magic Quadrant for Digital Experience Platforms.