



The Benefit of Microservices for Your Next Software Development Project

Contents

Introduction	3

Modularity and Scalability	5
Agility and Faster Time-to-Market	

Flexibility and Technology Diversity	6
Fault Isolation and Resilience	

Improved Maintenance and Updates	7
Enhanced Team Autonomy	

Scalable Development	8
Adaptability to Cloud-Native Environments	

Easy Integration with External Services	9
Improved Fault Detection and Troubleshooting	

Easier Adoption of DevOps Practices	9

Does your organisation's monolithic architecture mean that you struggle to keep pace with changing needs and market demand? Perhaps you're slow to facilitate new market segments and product lines? It could be time to consider what a microservices architecture can bring to your software development team and organisation as a whole.

Unlike monolithic applications, microservices architecture favours decoupled services with autonomous data. Rather than having one massive codebase and database for the entire application, it's broken down into smaller, independent services. Each of these services runs its own unique process and manages its own dedicated database, allowing for more flexibility and easier maintenance.

If you are thinking about retiring legacy software and systems, adopting a microservices approach to development offers several benefits that can significantly enhance agility, scalability and maintainability.

Let's discuss some key advantages of microservices architecture

1 Modularity and Scalability

Microservices break down your application into smaller, independent services, each focused on a specific business function, whether it be payments, user profiles or product search. This modularity brings with it agility and scalability. Need to handle more user traffic? Just expand the payment service. Want to test a new feature? Deploy it as a separate service without risking the entire system. The ability to scale each service independently allows you to optimise resource allocation and reduce unnecessary costs.

2 Agility and Faster Time-to-Market

Waiting for an entire application to be ready before releasing new features puts your organisation at risk of falling behind competitors. Microservices architecture means that your development teams can work on different functionalities simultaneously, each building parts of a product. Teams can make changes to one service without waiting for others to finish. The result? Faster development, testing and deployment cycles. Develop and deploy new features quickly and upgrade older components as new technologies emerge.

3 Flexibility and Technology Diversity

Microservices gives freedom to developers to move away from a prescribed tech stack. Unlike monolithic structures, developers can choose the best tool for each job. Teams can select the language, framework and database that fit each service's purpose. New technologies can also easily be integrated into specific services, opening up a culture of experimentation and innovation. Removing limitations and empowering developers boosts morale and productivity of dev teams.

4 Fault Isolation and Resilience

One bug or failure in a monolithic system can bring down the entire application. Microservices break the chain - if one service experiences a fault, the other services are not necessarily affected. This 'fault isolation' keeps your system resilient, preventing a single issue from causing widespread outages.

5 Improved Maintenance and Updates

Modularity streamlines maintenance and updates. It allows for targeted maintenance without the need to rebuild the entire application from scratch. Developers can address issues within specific microservices without impacting the whole system. Modularity makes rolling updates easier to manage operationally, ensuring minimal downtime and uninterrupted user experience.

6 Enhanced Team Autonomy

Microservices architecture creates decentralised, autonomous teams. Each service or critical business function can be owned by a specific team, which allows for more efficient workflows. This autonomy means that teams can choose the best tools and frameworks for their service, rapidly repeating processes to optimise performance. Gone are the days of waiting for the entire codebase to be greenlit; with microservices, teams can push updates and features faster.

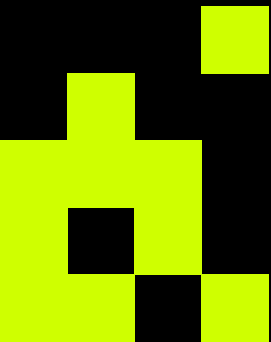
7 Scalable Development

In large-scale development projects, traditional monolithic structures often become bottlenecks, with different development teams hindering one another in a sprawling codebase. Microservices enable parallel development by multiple teams, allowing each to focus on specific modules without impacting one another. This scalability unlocks the agility needed for managing large projects with diverse requirements.

8 Adaptability to Cloud-Native Environments

Applications need to be adaptable. Microservices, which are suited to cloud-native environments, enable more agility. Each service handles specific functionalities, allowing for scaling, streamlined updates and innovation.

Allocate resources to each service based on real-time demands. Scale up or scale down based on requirements and optimise cloud-costs and performance.



9 Easy Integration with External Services

Microservices embrace open communication through APIs and enable seamless integrations with external services and third-party APIs. Plug in pre-built modules to your application for instant new functionalities and features. Access best-in-class functionalities and experiment with new features and technologies without building them in-house, freeing up resources for core development.

10 Improved Fault Detection and Troubleshooting

Troubleshooting monolithic applications can be time-consuming and frustrating - like searching for a needle in a haystack without knowing where to start. Microservices overcome this problem, with each standalone unit equipped with dedicated monitoring and logging. This granularity means that issues can be pinpointed to specific services, avoiding a system-wide search. Focused logs and metrics facilitate targeted debugging, leading to quicker problem-solving and minimal downtime.

11 Easier Adoption of DevOps Practices

Microservices can streamline your DevOps practices. Their modular nature aligns perfectly with CI/CD principles, enabling smaller, faster releases and continuous improvement. 'Mini-projects' for each service can be tested and deployed through automated pipelines. Promoting better collaboration, each team owns a service with shared ownership between developers and operations. This gives developers a deeper understanding of operational needs and ops teams more clarity on how to support deployments. Microservices enable better cross-team cooperation.

The transition from a monolithic architecture to microservices requires careful planning. Organisations need to consider factors such as cultural changes, infrastructure and the overall impact on existing processes.

However, there is no denying that the benefits of a microservices approach is substantial. Shifting away from monolithic architecture can give your organisation a competitive advantage and minimise operational risk.

With world-wide expertise in software development built on microservices, One Beyond has worked with organisations across multiple sectors, building agile web, mobile and cloud software solutions to unlock business growth.

Get in touch with the team today and talk to us about your bespoke software development requirements.

WEBENQUIRIES@ONE-BEYOND.COM

01252 902704

**ONE
BEYOND** 