

---

# Building web applications based on microservice architecture

**Сергей Михайлович Ухливанов,**  
старший программист /разработчик Оптум,  
США, Сан Франциско

**Sergey M. Ukhliivanov**  
Optum, Senior Software Engineer  
USA, San Francisco

E-mail: [sergey.ukhliivanov@gmail.com](mailto:sergey.ukhliivanov@gmail.com)

Keywords: web applications, development, microservices, architecture, deployment.

**Abstract:** The paper considers the peculiarities of building web applications based on microservice architecture. Special emphasis is placed on the fact that microservices architecture offers significant advantages for web development, especially for large and complex applications. It provides scalability, flexibility, increased fault tolerance and encourages technological diversity. This article describes the principles to be followed in the process of creating a microservice architecture and the basic steps of its construction.

## **Построение веб-приложений на основе микросервисной архитектуры**

Ключевые слова: веб-приложения, разработка, микросервисы, архитектура, развертывание.

Аннотация.

В статье рассмотрены особенности построения веб-приложений на основе микросервисной архитектуры. Отдельный акцент сделан на том, что архитектура микросервисов предлагает значительные преимущества для веб-разработки, особенно для больших и сложных приложений. Это обеспечивает масштабируемость, гибкость, повышенную отказоустойчивость и поощряет технологическое разнообразие. В статье описаны принципы, которых следует придерживаться в процессе создания микросервисной архитектуры, также обозначены основные шаги ее построения.

Millions of people around the world watch more than two billion hours of Netflix video every month. Another example is Facebook, which had approximately 2.85 billion active users in the first quarter of 2021. If you look at the work of the e-commerce giant Amazon, you can see that it receives more than 66 thousand orders per hour and 18.5 orders per second [1]. In light of these statistics, the question of how these organizations manage to manage such a colossal flow of consumers on their websites and applications is not trivial. How they allow millions of users to access their accounts, use available features, and perform desired actions without any hassle or downtime.

The answer to these questions is scalability and microservice architectures. Microservices architecture represents a refreshed approach. Instead of dealing with giant code bases, complex systems are broken down into smaller, modular parts (see Figure 1). Each of these parts, known as microservices, performs a specific task or function. This modular structure allows for cleaner, more manageable code, making development easier.

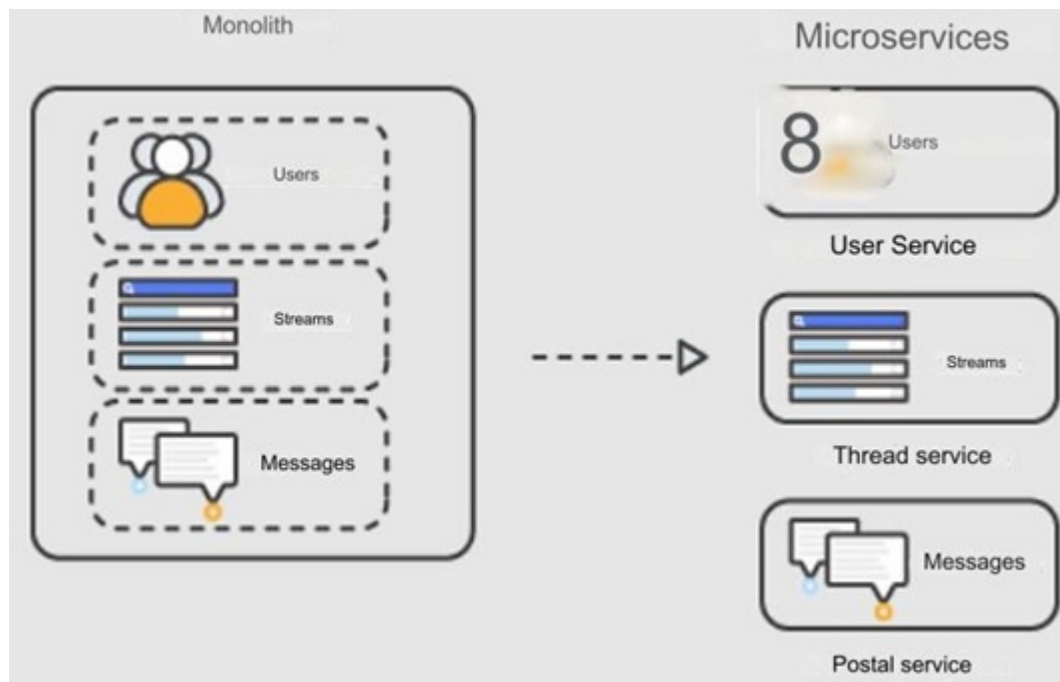


Figure 1 Breaking a monolithic application into microservices

Microservice architecture follows the design of SOA service-oriented architecture. Some experts argue that microservice is SOA 2.0 [2].

Taking into account the above, issues related to the study of microservices architecture for web applications, its basic principles, and the possibilities of real-life applications have important scientific and practical significance, which determined the choice of the topic of this article.

A comparison of monolithic and microservice architectures for creating web applications is carried out in their publications by Kiryanov V.A., Butorin K.A., Sukhov A.A., Moiseichik E.S., Davidovskaya M.I.

The defining features of microservice architecture and its various styles are described by E.V. Kryazheva, E.V. Amelin, O.V. Natrobina, V.Yu. Pashechko, A.S. Bekenova, Zh.S. Mutalova.

However, although a wide range of scientists and experts are interested in the problem under study, some points require clarification and additional analysis. Thus, the issue of creating a well-functioning API management platform that will eliminate the problems of microservice architecture and ensure its correct operation deserves special attention.

Thus, the purpose of the article is to study the features of building web applications based on microservice architecture.

First of all, it should be noted that microservice is a technique in which a single application is developed as a collection of small services, each of which has the ability to run its own process without depending on the others, and uses a lightweight mechanism for communication. These services share the least centralized control. In addition, they can be developed in different programming languages and use different technologies for storing data [3].

A microservices architecture requires careful planning and must be based on a set of core principles that guide its design and implementation for web applications. These principles lay the foundation for creating reliable, scalable and adaptable systems. Let's look at it in more detail.

#### *Single responsibility*

Every microservice should have a clearly defined, focused functionality. This approach promotes modularity, simplifies development and maintenance, and prevents changes to one service from affecting the entire application.

---

### *Free connection*

A microservices architecture for web applications communicates with each other through lightweight APIs, minimizing dependencies and encouraging independence. This makes it easy to scale and update individual services without affecting others.

### *Independent development and deployment*

Independently developing, testing, and deploying each microservice speeds up development cycles and allows for smoother upgrades. This allows web developers to work on different services simultaneously without affecting the entire application.

### *Technological independence*

Microservices architecture for web applications supports multiple languages and technologies, allowing developers to choose the best tools for each service. This makes it possible to avoid being tied to a manufacturer and adapt to future advances.

### *Alignment with Business Domains*

Designing microservices based on business domains promotes clear responsibility and accountability for specific application functionality. This promotes better understanding and maintainability.

### *API-based communication*

Microservices communicate with each other through standardized APIs, providing a consistent experience and facilitating seamless integration with other systems [4].

### *Fault isolation*

A key benefit of microservices architecture is its inherent fault tolerance. This means that if one individual service encounters a problem, it will not break the entire application.

The process of building web applications based on microservice architecture includes the following steps.

*Domain analysis.* To avoid some common mistakes when designing microservices, you need to use domain analysis to define the boundaries of the microservice. To do this, follow these steps:

1. Use domain analysis to model microservices.
2. Apply tactical domain-driven design tools.
3. Define the boundaries of microservices.

Designing services using a special methodology.

Operation in production. Because microservices architectures are distributed, reliable deployment and monitoring operations must be ensured.

Thus, to summarize, microservices are a popular architectural style for creating web applications that are fault-tolerant, highly scalable, independently deployable, and capable of rapid development.

## **References**

1. Ganzha A.Yu., Karelova R.A. Features of microservice architecture of event-driven web applications // Scientific review. Technical science. 2021. No. 6. P. 28-34.
2. Chernyakova E.I. Study of microservice architecture for a web application // Bulletin of the Master's degree. 2022. No. 12-6 (135). pp. 15-17.
3. Nagorny N.N. Main aspects of developing a microservice web application // International scientific research journal. 2023. No. 7 (133). pp. 76-81.

---

4. Vrublevskaya E.A. Designing a web application for monitoring consumption using microservice architecture // Scientific research of the XXI century. 2023. No. 1 (21). pp. 35-39.