

COMPARATIVE ANALYSIS OF MONOLITHIC AND MICROSERVICE ARCHITECTURES FOR SMALL PROJECTS

Vinnitsia National Technical University

Анотація

Публікація досліджує доцільність використання монолітної та мікросервісної архітектур у контексті розробки невеликих програмних продуктів. Здійснено детальне порівняння обох підходів за критеріями швидкості розробки, складності розгортання, масштабованості та витрат на інфраструктуру. Встановлено, що незважаючи на домінування мікросервісів у корпоративному сегменті, для малих проєктів використання модульного моноліту залишається більш ефективним рішенням. Це дозволяє уникнути надмірної інженерної складності та оптимізувати використання ресурсів на початкових етапах життєвого циклу програмного забезпечення.

Ключові слова: архітектура програмного забезпечення, моноліт, мікросервіси, патерни проєктування, розподілені системи, Spring Boot, модульний моноліт.

Abstract

The publication explores the feasibility of using monolithic and microservice architectures in the context of developing small software products. A detailed comparison of both approaches is carried out based on development speed, deployment complexity, scalability, and infrastructure costs. It is established that despite the dominance of microservices in the enterprise segment, using a modular monolith remains a more effective solution for small projects. This approach helps to avoid overengineering and optimizes resource utilization during the initial stages of the software lifecycle.

Keywords: software architecture, monolith, microservices, design patterns, distributed systems, Spring Boot, modular monolith.

Introduction

The evolution of software development approaches constantly shifts developers' focus between centralized and distributed systems. In recent years, microservice architecture has become the de facto standard for building scalable, high-load systems. As M. Richards and N. Ford point out, any architectural decision is a trade-off, and opting for a trend without considering the actual scale of the project often leads to disastrous consequences [1, p. 19]. The modern industry shows a tendency to apply microservices even to small applications, generating unjustified overhead in time and resources. The purpose of this study is an objective comparative analysis of monolithic and microservice approaches to determine the optimal architectural pattern for creating small systems based on modern frameworks.

Research results

Monolithic architecture involves compiling all the functionality of a software product into a single codebase and a single deployment module. Sam Newman emphasizes that a monolith is not an antipattern; on the contrary, for the vast majority of new or small projects, it is the fastest way to release [2, p. 79].

Let us consider this using the example of developing a small information system for controlling patient discharges and generating statistical reports, using the Java ecosystem and the Spring Boot framework. The main task of such a program does not involve complex calculations; it focuses on collecting data about treated patients and generating reports. At first glance, the system can be easily decomposed into independent microservices: a data entry module, a reference data control module (static reference information, such as the ICD-10 disease classifier, lists of hospital departments and staff), and a separate reporting module.

However, for such a project, the microservice approach generates unjustified complexity. Generating a report requires constantly joining patient data with system reference data. In a distributed microservice system, this would mean the need for constant transmission of large data arrays over the network (network overhead) or the creation of complex mechanisms for synchronizing databases between services [3]. Instead, the

monolithic approach allows keeping all data in a single relational database. This guarantees strict data consistency (ACID transactions), allows executing analytical SQL queries instantly, and minimizes deployment infrastructure setup [4].

Analysis shows that the threshold where the benefits of microservices begin to outweigh their complexity lies far beyond such small projects [3]. A more rational choice for such solutions is the Modular Monolith architecture. This approach involves developing a single application with strict internal isolation of components (for example, reference data logic and reporting logic are separated at the level of Java packages and interfaces). Consequently, the simplicity of the monolith is preserved, but the system remains flexible [5].

Conclusion

The conducted analysis demonstrates that choosing a microservice architecture by default for small projects is a technological error that leads to unjustified system overengineering. Monolithic architecture, especially in the form of a logically separated modular monolith, is the most appropriate and cost-effective choice for small systems. It provides a fast time-to-market, guarantees data integrity at the level of traditional databases, and requires minimal effort for deployment and maintenance. The transition to microservices should happen evolutionarily and only when the monolith can no longer handle the load or the organizational growth of the team.

REFERENCES

1. Richards M., Ford N. Fundamentals of Software Architecture: An Engineering Approach. 1st ed. Sebastopol: O'Reilly Media, 2020. 416 p.
2. Newman S. Building Microservices: Designing Fine-Grained Systems. 1st ed. Sebastopol: O'Reilly Media, 2015. 260 p.
3. Fowler M. Microservices Resource Guide. URL: <https://martinfowler.com/microservices/>
4. Spring Boot Reference Documentation. URL: <https://docs.spring.io/spring-boot/reference/>
5. Grzybek K. Modular Monolith: A Primer. URL: <https://www.kamilgrzybek.com/blog/posts/modular-monolith-primer>

Верещак Богдана Олександрівна – студентка групи 4ПІ-246, факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, м. Вінниця, email: ellae1419@gmail.com.

Науковий керівник: **Чопляк Вікторія Володимирівна** – викладач англійської мови, кафедра іноземних мов, Вінницький національний технічний університет, м. Вінниця, email: nikavnuchkova@gmail.com.

Bohdana O. Vereshchak – Faculty of Information Technologies and Computer Engineering, Vinnytsia National Technical University, Vinnytsia, email: ellae1419@gmail.com.

Scientific Supervisor: **Victoria V. Chopliak** – teacher of English, Foreign Languages Department, Vinnytsia National Technical University, Vinnytsia, email: nikavnuchkova@gmail.com.