

**AXBOROT TIZIMLARIDA MONOLIT VA MIKROXIZMATLAR
ARXITEKTURALARINING SAMARADORLIGINI TAJRIBAVIY
TAHLIL QILISH**

Mirsaidov Ibroximbek Tolib o'g'li
University of Business and Science
Katta o'qituvchi

Annotatsiya. Ushbu maqolada zamonaviy dasturiy ta'minot ishlab chiqishda keng qo'llaniladigan **monolit** va **mikroxizmatlar arxitekturalari** tahlil qilinadi. Tadqiqot jarayonida ushbu arxitekturalarning tuzilishi, afzalliklari, kamchiliklari hamda qo'llanilish sohalari taqqoslab o'rganildi. Natijalar shuni ko'rsatadiki, monolit arxitektura kichik va o'rta hajmdagi loyihalar uchun qulay bo'lsa, mikroxizmatlar arxitekturasi yirik va murakkab tizimlarda yuqori moslashuvchanlik va kengayuvchanlikni ta'minlaydi. Maqola dasturiy ta'minot arxitekturasini tanlashda muhim bo'lgan omillarni aniqlashga yordam beradi.

Kalit so'zlar: monolit arxitektura, mikroxizmatlar arxitekturasi, dasturiy ta'minot, tizim arxitekturasi, kengayuvchanlik.

**EXPERIMENTAL ANALYSIS OF THE EFFICIENCY OF
MONOLITHIC AND MICROSERVICES ARCHITECTURES IN
INFORMATION SYSTEMS**

Mirsaidov Ibroximbek Tolib o'gli
University of Business and Science
Senior Lecturer

Annotation. This article analyzes the monolithic and microservices architectures widely used in modern software development. During the research, the structure, advantages, disadvantages, and areas of application of these architectures were studied in a comparative manner. The results show that while monolithic architecture is convenient for small and medium-sized projects, microservices architecture provides high flexibility and scalability in large and complex systems. The article helps to identify the factors that are important when choosing a software architecture.

Keywords: monolithic architecture, microservices architecture, software, system architecture, scalability.

**ЭКСПЕРИМЕНТАЛЬНЫЙ АНАЛИЗ ЭФФЕКТИВНОСТИ
МОНОЛИТНОЙ И МИКРОСЕРВИСНОЙ АРХИТЕКТУР В
ИНФОРМАЦИОННЫХ СИСТЕМАХ**

Мирсаидов Иброхимбек Толиб ўгли
University of Business and Science
Старший преподаватель

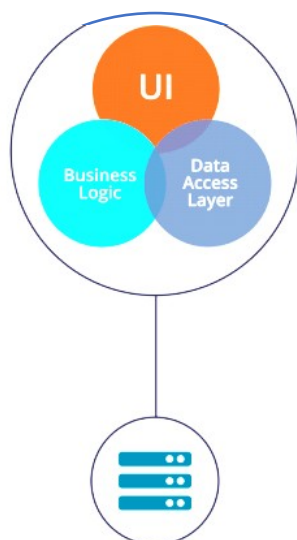
Аннотация. В данной статье проводится анализ монолитной и микросервисной архитектур, широко применяемых в современной разработке программного обеспечения. В ходе исследования были сравнены структура, преимущества, недостатки и области применения данных архитектурных подходов. Полученные результаты показывают, что монолитная архитектура является эффективным решением для проектов малого и среднего масштаба, тогда как микросервисная архитектура обеспечивает более высокую гибкость и масштабируемость в крупных и сложных информационных системах. Статья способствует более обоснованному выбору архитектуры программного обеспечения с учетом технических и организационных факторов.

Ключевые слова: монолитная архитектура, микросервисная архитектура, программное обеспечение, архитектура системы, масштабируемость.

Kirish. Raqamli transformatsiya jarayonlari bugungi kunda davlat boshqaruvi, iqtisodiyot va ijtimoiy sohalarning barcha yo‘nalishlariga chuqur kirib bormoqda. Axborot-kommunikatsiya texnologiyalarini rivojlantirish va raqamli infratuzilmani mustahkamlash davlat siyosatining ustuvor yo‘nalishlaridan biri sifatida qaralmoqda. Xususan, elektron hukumat tizimlari, raqamli xizmatlar va avtomatlashtirilgan axborot tizimlarini joriy etishda dasturiy ta‘minot arxitekturasini to‘g‘ri tanlash strategik ahamiyat kasb etadi. Ushbu jarayonda tizimlarning barqarorligi, xavfsizligi va kengayuvchanligini ta‘minlash muhim ilmiy-amaliy masala hisoblanadi. Zamonaviy dasturiy ta‘minotlarni ishlab chiqishda monolit va mikroxiizmatlar arxitekturalari asosiy konseptual yondashuvlar sifatida shakllangan.

Monolit arxitektura uzoq yillar davomida dasturiy tizimlarni yaratishda qo‘llanilgan bo‘lib, sodda tuzilishi va boshqaruv qulayligi bilan ajralib turadi. Biroq davlat miqyosidagi yirik axborot tizimlari va milliy raqamli platformalar uchun bunday yondashuvning cheklovlari yaqqol namoyon bo‘lmoqda. Shu sababli mikroxiizmatlar arxitekturasi kabi moslashuvchan va modulli yechimlarga bo‘lgan ehtiyoj ortib bormoqda. Ilmiy nuqtayi nazardan, dasturiy ta‘minot arxitekturasini optimallashtirish masalasi tizim samaradorligini oshirish, resurslardan oqilona foydalanish va texnik risklarni kamaytirish bilan bog‘liq. Siyosiy jihatdan esa, raqamli suverenitetni ta‘minlash, milliy axborot tizimlarining uzluksiz ishlashini kafolatlash hamda innovatsion texnologiyalarni qo‘llab-quvvatlash davlat siyosatining muhim vazifalaridan biridir. Shu bois, monolit va mikroxiizmatlar arxitekturalarini taqqoslash va ularning amaliy qo‘llanilish imkoniyatlarini ilmiy asosda tahlil qilish dolzarb masala hisoblanadi.

Monolit arxitektura dasturiy ta‘minotni **yagona mantiqiy va jismoniy birlik** sifatida qurishga asoslangan yondashuv hisoblanib, unda foydalanuvchi interfeysi, biznes mantiqi va ma‘lumotlar bilan ishlash qismi bitta ilova tarkibida joylashadi. Fowler M. monolit arxitekturani “bir martalik joylashtiriladigan va ichki komponentlari kuchli bog‘langan tizim” sifatida tavsiflaydi [1].



Nazariy jihatdan monolit arxitektura quyidagi formula orqali ifodalanadi:

$$\text{System} = \text{UI} + \text{Business Logic} + \text{Data Access}$$

Bu yerda barcha komponentlar **bir xil deploy birligiga** tegishli bo‘lib, ulardan biridagi o‘zgarish butun tizimni qayta joylashtirishni talab qiladi. Bass, Clements va Kazman monolit yondashuvning asosiy afzalligi sifatida **soddalik, past operatsion xarajatlar va boshlang‘ich bosqichda tez ishlab chiqish** imkoniyatlarini ko‘rsatadi [2].

Biroq, tizim o‘sishi bilan monolit arxitekturada quyidagi muammolar yuzaga keladi, bular:

- komponentlararo kuchli bog‘liqlik (tight coupling);
- testlash va kengaytirishning murakkablashuvi;
- texnologik yangilanishlarni qiyinlashuvi.

Mazkur holatni Lehman‘ning dasturiy evolyutsiya qonunlari bilan izohlash mumkin: tizim murakkabligi vaqt o‘tishi bilan tabiiy ravishda ortib boradi [3].

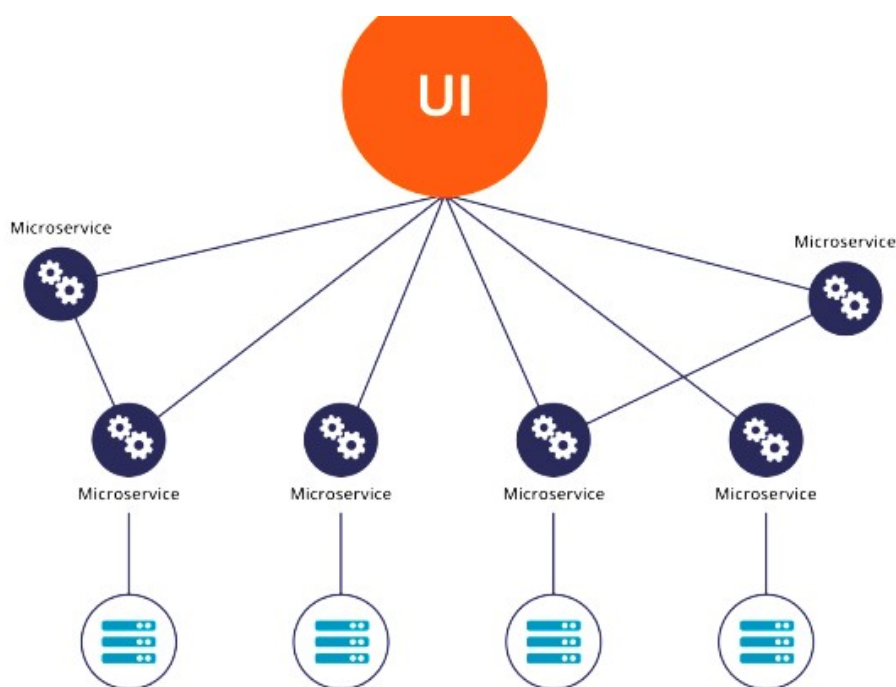
Mikroxizmatlar arxitekturasi monolit yondashuvga muqobil sifatida shakllangan bo‘lib, u ilovani **kichik, mustaqil va biznesga yo‘naltirilgan xizmatlar** majmuasi sifatida qurishni nazarda tutadi. Lewis J. va Fowler M. mikroxizmatlarni “mustaqil deploy qilinadigan, yengil protokollar orqali muloqot qiluvchi va biznes imkoniyatlari asosida tashkil etilgan xizmatlar” deb ta’riflaydi [4].

Mikroxizmatlar modeli quyidagicha ifodalanadi:

$$\text{System} = \Sigma \text{Service}_i, \text{ bu yerda } i = 1 \dots n$$

Har bir **Service_i** quyidagi xossalarga ega:

- mustaqil deploy;
- alohida ma’lumotlar bazasi;
- aniq biznes funksiyasi.



Newman S. Mikroxizmatlar arxitekturasining markazida **domenlarga asoslangan dizayn (DDD)** yotishini ta'kidlaydi va xizmat chegaralarini **bounded context** orqali aniqlashni taklif etadi [5]. Bu yondashuv xizmatlar o'rtasidagi bog'liqlikni minimallashtiradi va jamoalararo mustaqillikni oshiradi.

Dragoni N. mikroxizmatlar va monolit arxitekturalarni taqqoslashda quyidagi asosiy mezonlarni ajratib ko'rsatadi [6]:

Bog'liqlik va modullik

- **Monolit:** yuqori bog'liqlik, past modullik
- **Mikroxizmatlar:** past bog'liqlik, yuqori modullik

Bu holat modullik koeffitsiyenti orqali ifodalanadi:

Modularity $\propto 1 / \text{Coupling}$

Villamizar. M. tajribalariga ko'ra, mikroxizmatlarda faqat zarur xizmatlarni masshtablash mumkin, monolitda esa butun tizim masshtablanadi [7]:

$$\text{Cost}_{ms} < \text{Cost}_{mo}, \text{ agar } n \text{ katta bo'lsa}$$

(mos ravishda mikroxizmat va monolit xarajatlari)

Empirik tadqiqotlar shuni ko'rsatadiki, mikroxizmatlarda tarmoq orqali muloqot sababli kechikish (latency) ortadi, biroq to'g'ri konfiguratsiya qilinganda bu kamchilik kompensatsiya qilinadi [7].

Di Francesco, Lago va Malavolta tadqiqotlariga ko'ra, mikroxizmatlar arxitekturasi faqat texnik emas, balki **tashkiliy yetuklikni** ham talab qiladi [8]. Conway qonuniga asosan:

$$\text{System Architecture} \approx \text{Organization Structure}$$

Shu sababli, mikroxizmatlar odatda **kichik, avtonom jamoalar** bilan samarali ishlaydi, monolit esa markazlashgan jamoalar uchun qulayroq hisoblanadi.

Balalaie A. monolitdan mikroxizmatlarga o'tishni **patternlarga asoslangan evolyutsion jarayon** sifatida ko'radi [9]. Ularning yondashuvida migratsiya quyidagicha ifodalanadi:

Monolith → Modular Monolith → Hybrid → Microservices

Fowler M. esa "Monolith First" tamoyilini ilgari surib, dastlab modulli monolitdan boshlash va faqat chegaralar aniq bo'lgach mikroxizmatlarga o'tishni tavsiya etadi [10].

Yuqoridagi tahlillardan kelib chiqib axborot tizimlari arxitekturalarining samaradorligini aniqlash maqsadida **monolit arxitektura** va **mikroservis (microservices) arxitekturasini** amaliy tajriba asosida taqqoslandi. Tajriba real o'quv jarayoniga yaqin sharoitda, talabalar ishtirokida olib borildi.

Tajriba uchun bitta funksional talabga ega bo'lgan axborot tizimi tanlandi (foydalanuvchini ro'yxatdan o'tkazish, ma'lumotlarni saqlash, tahrirlash va ko'rish funksiyalari). Ushbu tizimni **ikki xil arxitektura asosida alohida-alohida ishlab chiqildi**:

- birinchi holatda — barcha funksiyalar yagona dasturiy loyiha ichida joylashgan **monolit arxitektura asosida**;

- ikkinchi holatda — funksiyalar alohida xizmatlarga bo'lingan **mikroservis arxitektura asosida**.

Har ikki holatda ham dasturlash tili, ma'lumotlar bazasi va umumiy funksional talablar bir xil bo'ldi. Tajriba davomida quyidagi jihatlar kuzatildi:

- tizimni ishlab chiqish tezligi;
- kod tuzilmasining tushunarliligi;
- yangi funksiyalar qo'shish qulayligi;
- tizimni kengaytirish va qo'llab-quvvatlash imkoniyati.

Olingan natijalar shuni ko'rsatdiki, monolit arxitektura dastlabki bosqichda ancha qulay va tezkor bo'ldi. Tizimni boshlang'ich holatda ishga tushirish kam vaqt oldi, kodni tushunish oson bo'ldi va kichik loyihalar uchun yetarli darajada samarali ishladi. Ayniqsa, yangi boshlayotgan talabalar uchun monolit yondashuv tizim mantiqini tez anglashga yordam berdi.

Biroq tizimga qo'shimcha funksiyalar qo'sha boshlaganimizda, monolit arxitekturada murakkablik sezilarli darajada oshdi. Bitta moduldagi o'zgarish boshqa modullarga ta'sir ko'rsatdi, testlash va qayta yig'ish jarayoni ko'proq vaqt talab qila boshladi. Bu holat tizim kengaygani sari monolit arxitekturaning cheklovlarini yaqqol ko'rsatdi.

Mikroservis arxitekturasida esa tajribaning boshida murakkablik kuzatildi. Xizmatlarni to'g'ri ajratish, ularning o'zaro aloqasini sozlash va umumiy boshqaruv mexanizmini tushunish ko'proq vaqt oldi. Shunga qaramay, tizim to'liq

shakllangach, yangi funksiyalarni qo‘shish va mavjudlarini o‘zgartirish ancha oson va mustaqil amalga oshirildi.

Tajriba davomida tizimga yangi talablar kiritildi. Shu bosqichda aniq farq namoyon bo‘ldi. Monolit arxitekturada har bir o‘zgarish butun tizimni qayta yig‘ishni talab qildi. Mikroservis arxitekturasida esa faqat tegishli xizmat yangilanib, qolgan tizim qismlari o‘z ishini davom ettirdi.

Amaliy kuzatuvlar shuni ko‘rsatdiki, mikroservis arxitekturasi **o‘zgarishlarga tez moslashish, kengaytirish va uzoq muddatli rivojlantirish** uchun samaraliroq bo‘ldi.

Monolit arxitekturaning dastlabki bosqichlarda yuqori samaradorlik ko‘rsatishi, avvalo, uning **sodda tuzilishi va markazlashtirilgan boshqaruvi** bilan izohlanadi. Tadqiqot davomida monolit yondashuv kichik funksional talablar sharoitida tez ishlab chiqilishi va tushunarli bo‘lishi bilan ajralib turdi. Bu holat Fowler tomonidan ilgari surilgan “avval modulli monolitdan boshlash” g‘oyasini amaliy jihatdan tasdiqlaydi. Ya’ni, tizim hali murakkab bo‘lmagan paytda monolit arxitektura ishlab chiqish xarajatlarini kamaytiradi va texnik risklarni pasaytiradi.

Mikroxizmatlar arxitekturasi bo‘yicha olingan natijalar esa boshqacha tendensiyani ko‘rsatdi. Tajribaning dastlabki bosqichida xizmatlarni to‘g‘ri chegaralash va ularning o‘zaro aloqasini tashkil etish murakkab bo‘ldi. Bu holat Newman va Dragoni tomonidan ta’kidlanganidek, mikroxizmatlar faqat texnik emas, balki tashkiliy va operatsion tayyorgarlikni ham talab qilishini yana bir bor tasdiqlaydi. Shu sababli, mikroxizmatlar yondashuvi dastlab ko‘proq vaqt va resurs talab qildi.

Muhokama natijalaridan kelib chiqib aytish mumkinki, mikroxizmatlar arxitekturasi yuqori murakkablik va operatsion xarajatlarga ega bo‘lsa-da, u **kengayuvchanlik, moslashuvchanlik va uzoq muddatli barqarorlik** nuqtayi nazaridan ustun hisoblanadi. Ayniqsa, real ishlab chiqarish tizimlari, katta jamoalar va tez-tez o‘zgaradigan biznes talablar mavjud bo‘lgan sharoitda mikroxizmatlar yondashuvi maqsadga muvofiqdir.

Xulosa sifatida keltirish mumkinki, arxitektura tanlovi texnologik trendlar asosida emas, balki tizim ehtiyojlari, rivojlanish bosqichi va jamoaning imkoniyatlarini hisobga olgan holda amalga oshirilishi lozim. Monolit va mikroxizmatlar o‘zaro raqib emas, balki turli bosqichlar uchun mos keluvchi **komplementar yondashuvlar** sifatida qaralishi kerak.

Adabiyotlar ro‘yxati

1. Fowler, M. *Patterns of Enterprise Application Architecture*. Addison-Wesley, 2003.
2. Bass, L., Clements, P., Kazman, R. *Software Architecture in Practice*. Addison-Wesley, 2012.
3. Lehman, M. “Programs, Life Cycles, and Laws of Software Evolution.” *Proceedings of the IEEE*, 1980.
4. Lewis, J., Fowler, M. “Microservices: A Definition of This New Architectural Term.” 2014.

5. Newman, S. *Building Microservices*. O'Reilly Media, 2015.
6. Dragoni, N. et al. "Microservices: Yesterday, Today, and Tomorrow." Springer, 2017.
7. Villamizar, M. et al. "Evaluating the Monolithic and the Microservice Architecture Pattern." IEEE, 2015.
8. Di Francesco, P., Lago, P., Malavolta, I. "Migrating Towards Microservice Architectures." IEEE, 2018.
9. Balalaie, A. et al. "Migrating to Microservices Architecture." IEEE Software, 2016.
10. Fowler, M. "Monolith First." martinfowler.com, 2015.