



## DASTURIY TA'MINOT ARXITEKTURASINI ISHLAB CHIQUISH.

Ismailov Alisher Shokirovich

Mahmudova Madina Bahodir qizi

Toshkent Davlat Iqtisodiyot Universiteti

<https://doi.org/10.5281/zenodo.15395894>

### ARTICLE INFO

Received: 28th April 2025

Accepted: 30<sup>th</sup> April 2025

Published: 12<sup>th</sup> May 2025

### KEYWORDS

*Scalability (Miqyoslashuvchanlik) , Maintainability (Qo'llab-quvvatlashga layoqlilik) , Security (Xavfsizlik) , Arxitektura loyihalash , Arxitektura baholash , Dasturiy tizimlar , Dasturiy ta'minot arxitekturasini.*

### ABSTRACT

*Ushbu maqola dasturiy ta'minot arxitekturasini ishlab chiqishning zamonaviy tendensiyalari, usullari va tamoyillarini o'rganadi. Unda dasturiy tizimlar uchun mustahkam, moslashuvchan va samarali arxitekturalarni loyihalash jarayoni batafsil ko'rib chiqiladi. Arxitektura uslublari (mikroservislar, monolit, serverless va boshqalar), loyihalash naqshlari (design patterns), sifat atributlari (scalability, maintainability, security), shuningdek, arxitektura qarorlarini qabul qilish va ularni baholash usullari tahlil etiladi. Ishning maqsadi dasturiy ta'minot arxitektorlari va ishlab chiquvchilariga murakkab dasturiy tizimlarni muvaffaqiyatli loyihalash va amalga oshirish uchun amaliy bilim va ko'nikmalarni berishdan iborat.*

Dasturiy ta'minot arxitekturasini - bu murakkab dasturiy tizimning asosiy tuzilishi, uning tarkibiy qismlari, ular o'rtasidagi munosabatlar va ularni boshqaruvchi tamoyillar hamda ko'rsatmalarni belgilab beruvchi muhim bosqichdir. Xuddi binoning loyihasi uning mustahkamligi va funktsionalligini ta'minlaganidek, yaxshi ishlab chiqilgan dasturiy ta'minot arxitekturasini ham tizimning muvaffaqiyatli rivojlanishi, oson boshqarilishi, kengaytirilishi va talablarga moslashuvchanligining poydevori hisoblanadi. Ushbu jarayon dasturiy ta'minotni yaratishning dastlabki bosqichlaridan biri bo'lib, keyingi barcha rivojlanish jarayonlariga sezilarli ta'sir ko'rsatadi.

Dasturiy ta'minot arxitekturasining asosiy tushunchalari:

**Tarkibiy qismlar (Components):** Dasturiy tizimning mustaqil, alohida bo'laklari. Bular modullar, xizmatlar, kutubxonalar yoki boshqa ijro etiladigan birliklar bo'lishi mumkin. Har bir komponent ma'lum bir vazifani bajaradi va boshqa komponentlar bilan aniq belgilangan interfeyslar orqali o'zaro ta'sirlashadi.

**Aloqalar (Connectors):** Tarkibiy qismlar o'rtasidagi o'zaro ta'sir mexanizmlari. Bular ma'lumot uzatish protokollari, funktsiya chaqiruvlari, xabar almashinuvi yoki boshqa aloqa usullari bo'lishi mumkin. Arxitektura aloqalarning turi va xususiyatlarini aniqlaydi.

**Konfiguratsiya (Configuration):** Tizimning tarkibiy qismlari va ularning aloqalari tashkil etilish usuli. Arxitektura tizimning umumiy tuzilishini va uning qismlari qanday joylashganligini belgilaydi.

Tamoyillar (Principles): Arxitektura loyihalashda qo'llaniladigan asosiy qoidalar va yo'nalishlar. Bular, masalan, yagona mas'uliyat tamoyili (Single Responsibility Principle), ochiq/yopiq tamoyili (Open/Closed Principle) va boshqalar bo'lishi mumkin.

Asosiy arxitektura uslublari:

Monolit arxitektura: Barcha funktsionallik bitta katta kod bazasida jamlangan. Bu uslub kichik va oddiy loyihalar uchun mos bo'lishi mumkin, ammo tizim o'sishi bilan uni qo'llab-quvvatlash va yangilash qiyinlashadi.

Mikroservislar arxitekturasi: Tizim kichik, mustaqil xizmatlarga bo'linadi, har biri o'z jarayonida ishlaydi va yengil protokollar (odatda HTTP API) orqali o'zaro aloqa qiladi. Bu uslub katta va murakkab loyihalar uchun moslashuvchanlik, mustaqil joylashtirish va texnologik xilma-xillikni ta'minlaydi.

Serverless arxitektura: Dasturiy ta'minotni serverlarni boshqarish zaruratisiz ishga tushirish imkonini beradi. Provyayder resurslarni dinamik ravishda taqsimlaydi va foydalanishga qarab to'lov amalga oshiriladi. Funktsiya sifatida xizmat (Function as a Service - FaaS) va orqa-end sifatida xizmat (Backend as a Service - BaaS) kabi turlari mavjud.

Event-driven arxitektura: Tizim komponentlari voqealarga (events) javob berish orqali o'zaro ta'sirlashadi. Bu uslub real vaqt rejimida ishlaydigan va yuqori darajada ajratilgan tizimlar uchun foydalidir.

Layered (Ko'p qatlamli) arxitektura: Tizim funktsionalligi bir nechta mantiqiy qatlamlarga ajratiladi (masalan, taqdimot qatlami, biznes-mantiq qatlami, ma'lumotlarga kirish qatlami). Har bir qatlam ma'lum bir mas'uliyatga ega va faqat o'ziga yaqin qatlamlar bilan o'zaro aloqa qiladi.

Loyihalash naqshlari (Design Patterns):

Loyihalash naqshlari dasturiy ta'minotni loyihalashda takrorlanuvchi muammolarning tasdiqlangan yechimlaridir. Ular arxitektura darajasida ham (masalan, Model-View-Controller, Publish-Subscribe) va dizayn darajasida ham (masalan, Factory, Singleton) qo'llanilishi mumkin. Loyihalash naqshlaridan foydalanish kodning qayta ishlatilishini oshiradi, uni tushunarliroq va qo'llab-quvvatlashga osonroq qiladi.

Sifat atributlari (Quality Attributes):

Sifat atributlari dasturiy tizimning non-funktsional talablarini ifodalaydi. Arxitektura tanlashda va loyihalashda ushbu atributlarga erishish muhim ahamiyatga ega. Ba'zi muhim sifat atributlari:

Unumdorlik (Performance): Tizimning ma'lum bir yuk ostida tez va samarali ishlash qobiliyati.

Miqyoslashuvchanlik (Scalability): Tizimning foydalanuvchilar soni yoki yuk ortishi bilan ishlash qobiliyatini saqlab qolish yoki oshirish imkoniyati.

Ishonchlilik (Reliability): Tizimning kutilmagan xatolarsiz uzoq vaqt davomida to'g'ri ishlash qobiliyati.

Qo'llab-quvvatlashga layoqlilik (Maintainability): Tizimni o'zgartirish, tuzatish va yangilash qanchalik osonligi.

Xavfsizlik (Security): Tizimning ruxsatsiz kirish, ma'lumotlarni o'g'irlash yoki buzishdan himoyalanganligi.

Testlashga layoqlilik (Testability): Tizimning turli xil sinovlarni o'tkazish uchun qanchalik qulayligi.

Foydalanishga qulaylik (Usability): Tizimning foydalanuvchilar uchun qanchalik oson va intuitiv ekanligi.

Arxitektura qarorlarini qabul qilish va baholash:

Arxitektura qarorlarini qabul qilish murakkab jarayon bo'lib, turli xil omillarni hisobga olishni talab qiladi, jumladan, biznes talablari, texnologik cheklovlar, jamoaning tajribasi va byudjet.

Arxitektura loyihasi ishlab chiqilgandan so'ng, uni baholash muhimdir. Arxitektura baholash usullari (masalan, Architecture Tradeoff Analysis Method - ATAM) potentsial risklarni aniqlash, turli arxitektura yechimlarining afzalliklari va kamchiliklarini taqqoslash va eng mos variantni tanlashga yordam beradi.

Dasturiy ta'minot arxitekturasini ishlab chiqish - bu dasturiy ta'minot tizimining tarkibiy qismlarini, ularning o'zaro aloqalarini va tashqi muhit bilan munosabatlarini belgilash jarayonidir. Yaxshi ishlab chiqilgan arxitektura dasturiy ta'minotning quyidagi sifatlarini ta'minlashga yordam beradi:

Ishonchlilik: Tizimning kutilganidek ishlashi va xatolarga bardosh berishi.

Scalability (Masshtablilik): Tizimning ortib borayotgan yuklamalarga bardosh bera olishi.

Maintainability (Qulaylik bilan xizmat ko'rsatish): Tizimni o'zgartirish, yangilash va xatolarni tuzatishning osonligi.

Flexibility (Moslashuvchanlik): Tizimning yangi talablarga moslasha olishi.

Performance (Ishlash tezligi): Tizimning resurslardan samarali foydalanishi va tezkor javob berishi.

Security (Xavfsizlik): Tizimning ruqsatsiz kirishdan himoyalanganligi.

Dasturiy ta'minot arxitekturasini ishlab chiqishda quyidagi asosiy bosqichlar mavjud:

Talablarni aniqlash: Mijozning ehtiyojlari va tizimning funksional va nofunksional talablari aniqlanadi.

Arxitektura uslubini tanlash: Loyiha talablariga eng mos keladigan arxitektura uslubi tanlanadi (masalan, mikroservislar, monolit, klient-server va boshqalar).

Tizimni modullarga bo'lish: Tizim mantiqiy va mustaqil bo'lgan kichikroq qismlarga (modullarga) bo'linadi.

Interfeyslarni belgilash: Modullar o'rtasidagi o'zaro aloqa usullari va interfeyslari aniqlanadi.

Texnologiyalarni tanlash: Loyihani amalga oshirish uchun zarur bo'lgan dasturlash tillari, ma'lumotlar bazalari, frameworklar va boshqa texnologiyalar tanlanadi.

Arxitektura diagrammasini chizish: Tizimning tuzilishini vizual tarzda ifodalovchi diagrammalar yaratiladi.

Prototip yaratish (ba'zan): Arxitekturaning ayrim qismlari yoki asosiy funksiyalari prototip sifatida ishlab chiqiladi.

Arxitekturani baholash: Ishlab chiqilgan arxitektura talablarga javob berishi va sifat atributlarini ta'minlashi baholanadi.

Hujjatlashtirish: Arxitektura qarorlari, dizayn tamoyillari va tizimning tuzilishi hujjatlashtiriladi.

Dasturiy ta'minot arxitekturasining turli xil uslublari mavjud bo'lib, ularning har biri o'zining afzalliklari va kamchiliklariga ega. Ba'zi mashhur arxitektura uslublari:

Monolit arxitektura: Barcha komponentlar bitta katta dastur sifatida joylashgan.

Mikroservislar arxitekturasi: Tizim kichik, mustaqil xizmatlarga bo'linadi, ular o'zaro tarmoq orqali aloqa qiladi.

Klient-server arxitekturasi: Klientlar serverdan resurslar yoki xizmatlarni so'raydi.

Qatlamli arxitektura: Tizim bir nechta qatlamlarga bo'linadi (masalan, prezentatsiya qatlami, biznes-logika qatlami, ma'lumotlar qatlami).

Voqealarga asoslangan arxitektura: Komponentlar voqealar orqali o'zaro aloqa qiladi.

Dasturiy ta'minot arxitekturasini ishlab chiqish - bu murakkab va muhim jarayon bo'lib, loyihaning muvaffaqiyatiga katta ta'sir ko'rsatadi. To'g'ri tanlangan va yaxshi ishlab chiqilgan arxitektura dasturiy ta'minotning sifatini oshiradi va uni kelajakdagi o'zgarishlarga tayyorlaydi.

Ushbu maqola raqamli dunyoda dasturiy ta'minot har qanday tashkilot va shaxs hayotining ajralmas qismiga aylandi. Bank operatsiyalari, elektron tijorat platformalari, kommunikatsiya vositalari va boshqa ko'plab muhim tizimlar ishonchli va samarali dasturiy ta'minotga tayanadi. Bunday murakkab va keng qamrovli dasturiy tizimlarning muvaffaqiyati ularning asosida yotgan mustahkam va to'g'ri tanlangan arxitekturaga bog'liqdir. Dasturiy ta'minot arxitekturasi tizimning asosiy tuzilishi, uning tarkibiy qismlari, ular o'rtasidagi munosabatlar va ularni boshqaruvchi tamoyillarni belgilaydi. Yaxshi loyihalashtirilgan arxitektura dasturiy ta'minotning sifatini oshiradi, uni o'zgarishlarga moslashuvchan qiladi, ishlab chiqish jarayonini tezlashtiradi va xarajatlarni kamaytiradi. Aksincha, noto'g'ri tanlangan yoki e'tiborsiz qoldirilgan arxitektura tizimning barqarorligi, unumdorligi va rivojlanish imkoniyatlarini cheklab qo'yishi mumkin. Shu sababli, dasturiy ta'minot arxitekturasini ishlab chiqish zamonaviy dasturiy injiniringning muhim va dolzarb yo'nalishlaridan biri hisoblanadi. Ushbu ish dasturiy ta'minot arxitekturasining asosiy tushunchalari, zamonaviy uslublari, loyihalash naqshlari va sifat atributlarini o'rganishga qaratilgan. Maqsad dasturiy ta'minot arxitektorlari va ishlab chiquvchilariga amaliyotda qo'llashlari mumkin bo'lgan bilim va ko'nikmalarni berishdir.

### **Xulosa**

Ushbu loyiha davomida dasturiy ta'minot arxitekturasini ishlab chiqishning fundamental tushunchalari, zamonaviy uslublari, muhim loyihalash naqshlari va sifat atributlari atroflicha ko'rib chiqildi. Dasturiy tizimlarning muvaffaqiyatli faoliyati va uzoq muddatli rivojlanishi uchun to'g'ri tanlangan va yaxshi loyihalashtirilgan arxitekturaning ahamiyati ta'kidlandi. Bugungi kunda tez sur'atlar bilan o'zgarib borayotgan texnologiyalar va biznes talablari sharoitida dasturiy ta'minot arxitekturasiga bo'lgan ehtiyoj tobora ortib bormoqda. Mikroservislar, serverless kabi yangi arxitektura uslublari tizimlarning moslashuvchanligi, miqyoslashuvchanligi va mustaqil rivojlanish imkoniyatlarini kengaytirmoqda. Shu bilan birga, loyihalash naqshlaridan oqilona foydalanish va sifat atributlariga e'tibor qaratish dasturiy ta'minotning ishonchliligi, xavfsizligi va qo'llab-quvvatlashga layoqliligini ta'minlashda muhim rol o'ynaydi. Dasturiy ta'minot arxitekturasini ishlab chiqish nafaqat texnik jarayon, balki biznes maqsadlari va foydalanuvchi ehtiyojlarini hisobga olgan holda strategik qarorlar qabul qilish san'atidir. Dasturiy ta'minot arxitektori zamonaviy texnologiyalar, arxitektura uslublari va loyihalash tamoyillarini chuqur bilishi, shuningdek, muammolarni tahlil qilish va optimal yechimlarni topish qobiliyatiga ega bo'lishi zarur. Kelgusida dasturiy ta'minot arxitekturasining evolyutsiyasi davom etishi, sun'iy intellekt, bulutli texnologiyalar va boshqa innovatsiyalarning integratsiyasi orqali yanada takomillashuvi kutilmoqda.

1. Isoqjonov, B., & Ismailov, A. (2025). PYTHONDA MODULLI DASTURLASH. Наука и технология в современном мире, 4(5), 61-63.
2. Ismoilov, A., Hamidov, M., & Sodiqov, N. (2025). PYTHON DASTURLASH TILIDA FAYLLAR BILAN ISHLASH. Академические исследования в современной науке, 4(12), 29-34.
3. Ismoilov, A., & Halimjonov, S. (2025). PYTHON DASTURLASH TILIDA TOPLAMLAR VA FAYLLAR. Наука и технология в современном мире, 4(5), 44-47.
4. Serobiddinova, M., & Ismailov, A. (2024, December). ALGORITMLARNI TAHLIL QILISHNING NAZARIY VA AMALIY JIHATLARI. In INTERNATIONAL CONFERENCE ON INTERDISCIPLINARY SCIENCE (Vol. 1, No. 12, pp. 339-344).
5. Tursunov, Z., To'ychiyev, B., & SH, I. A. (2024, October). KOMPONENTA TUSHUNCHASI VA XUSUSIYATLARI. In INTERNATIONAL CONFERENCE ON ADVANCE SCIENCE AND TECHNOLOGY (Vol. 1, No. 10, pp. 39-45).
6. Ne'matov, D., Sodiqov, Q., & Ismoilov, A. (2024, October). STANDART ALGORITMLAR. In INTERNATIONAL CONFERENCE ON ADVANCE SCIENCE AND TECHNOLOGY (Vol. 1, No. 10, pp. 54-58).
7. Abdumutaliyev, A., Nemadjonov, F., & Ismailov, A. (2024, October). KONTEYNERLAR BILAN ISHLAYDIGAN ALGORITMLAR. In INTERNATIONAL CONFERENCE ON ADVANCE SCIENCE AND TECHNOLOGY (Vol. 1, No. 10, pp. 32-38).
8. Umarjanov, S., Choriyev, M., & Ismailov, A. (2024, October). STL KUTUBXONALARI. In INTERNATIONAL CONFERENCE ON ADVANCE SCIENCE AND TECHNOLOGY (Vol. 1, No. 10, pp. 46-53).

INNOVATIVE  
ACADEMY