ISSN 2181-3187

API GATEWAY(AG) MIKROSERVISLAR TO'PLAMIDAGI IMKONIYATLARI

G.Rayimjonov
Erkin tadqiqotchi

Annotatsiya. Hozirgi kunda mikroservislar va klient-server arxitekturasi, kompleks tizimlarni shakllantirish, yaxshilash va rivojlantirishda keng qo'llaniladigan arxitekturalar hisoblanadi. Arxitekturalarni tanlashda, dasturlar va tizimlarning afzalliklarini va talablarini tushunish va aniqlash muhim jarayon. Ushbu ishda mikroservislar va klient-server arxitekturalarining tahlili keltirilgan.

Kalit so'zlar: API Gateway, Server, API, HTTP, arxitektura, web, Klient server, Microservices

Tasavvur qiling online do'kon yaratmoqchisiz. Ilovangizda quyidagi imkoniyatlar mavjud:

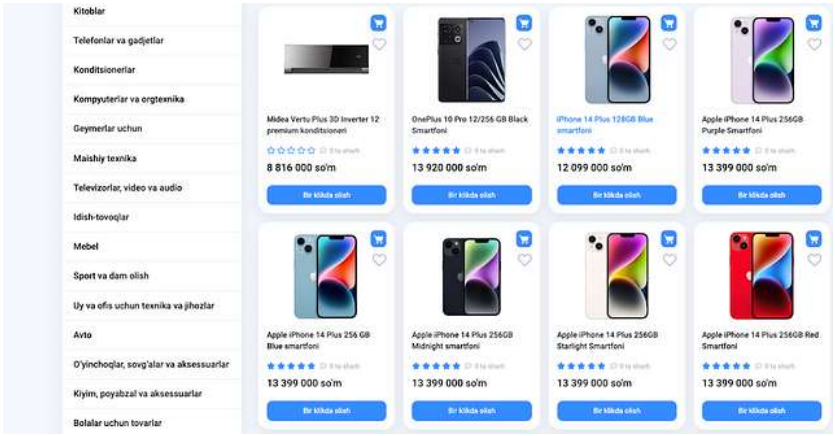
- Buyurtma berish
- Mahsulot haqida ma'lumotlar olish
- Mahsulot ro'yxatini olish

Shu xususiyatar asosida bizda bir nechta servislar(mikroservices) mavjud:

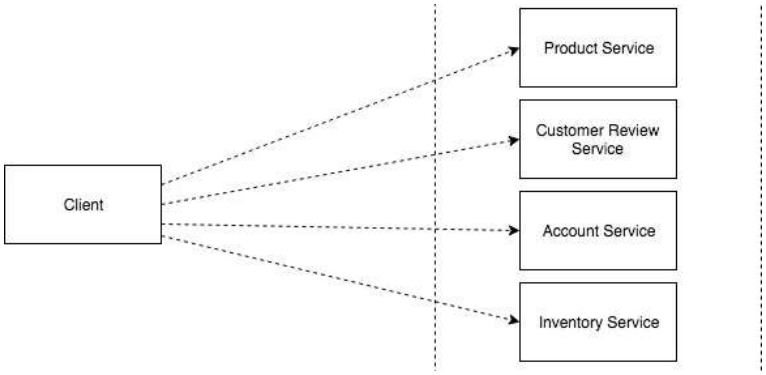
- Product Service
- Order Service
- Pricing Service
- Account Service
- Customer Review Service
- Inventory Service

Ilovangizdan foydalanish davomida foydalanuvchi bir necha servislardan foydalanadi. Agar siz asosiy oynaga kirib biror bir mahsulotga qidiruv bersangiz, mahsulotlar ro'yxatini chiqarib beradi. Bu ro'yxat shakllanish jarayonida yuqoridagi servislarga murojaat qilib mahsulot nomi, narxi, mahsulot tarixi va hokazolarni list ko'rinishida taqdim etadi. (quyidagi rasm)

ISSN 2181-3187



Shunday qilib mahsulotni ko’rsatadigan kod **product servis, account servis, customer review servis lardan** ma’lumotlarni olib bizga ko’rsatadi.



Bu grafikdagi klient ilovasi **API kompasitor** vazifasini bajaradi. U bir nechta servis larni chaqiradi va birlashtiradi.

Bu holatda maummo nimada?

Yondashuv oqilona bo’lib ko’rinsada unda jiddiy muammolar mavjud.

1. **User ma’lumotlarini olish uchun birnechta api ga request jo’natish** Foydalanuvchi ma’lumotini olish uchun bir nechta so’rovlarni amalga oshirish kerak va foydalanuvchi so’rovlarini ketma-ketlikda bajarish kerak. Bu ketma-ketlikni amalga oshirish uchun **api kompozitsiyon kodini** yozish talab qilinadi va bu potensial murakkab vazifa.

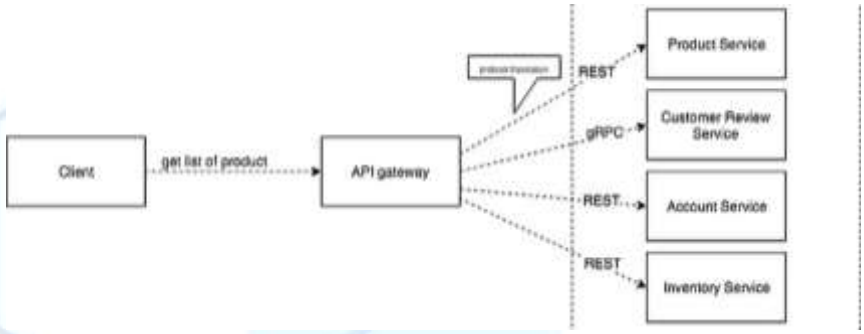
2. **Servis lar o’zgarishidagi muammolar(Lack Of Encapsulation)** Ma’lumki servislarga to’g’ridan to’g’ri kirishda ko’plab muammolar mavjud. Application rivojlanib borar ekan, bazida api larni o’zgartirishga to’g’ri keladi. Natijada o’zgargan api lardan foydalanayotgan mijozlarda(web, andorid, iOS) larda api bilan bo’gliq muammolar paydo bo’ladi.

3. **Mos kelmaydigan protokol(Unfriendly Protocol).** Bazi servis lar **gRPC** yoki **AMQP** messaging protokollar ishlatilishi mumkin. Bu ichki servis lar uchun juda zo’r yechim bo’lishi mumkin. Lekin mobile klient yoki veb klientlar uchun anchgina muammo bo’lishi mumkin. Yoki ba’zi bir protokol mexanizmlarini klient platformalarida moslashtirish qiyin bo’lishi mumkin.

ISSN 2181-3187

Bu holatda qanday yechim qilish mumkin? degan tabiiy savol tug'ilishi mumkun. Endi bu muammoni yechimi haqida gaplashsak.

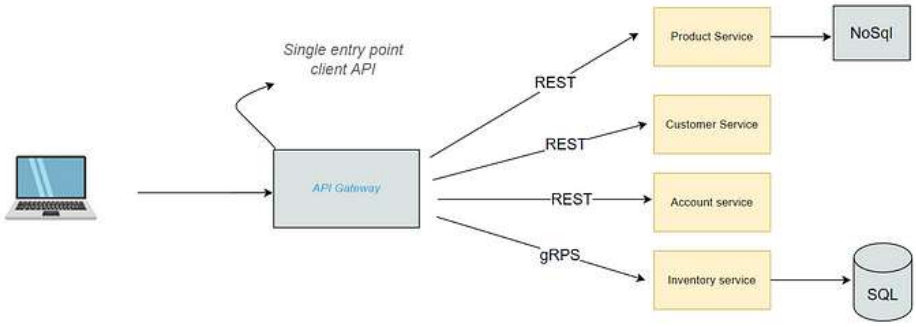
Yechim:



API Gateway(AG) — bu mikroservislar to'plami uchun yagona kirish nuqtasi vazifasini bajaradigan servis.

API Gateway klientdan kelgan request(so'rov)larni qabul qiladi, ularni tegishli mikroservis larga yo'naltiradi va ularni javobini mijozga qaytaradi.

API Gateway marshrutlash, **autentifikatsiya** va **rate limiting** kabi vazifalar uchun javobgardir. Bu mikroservislarga o'zlarining individual vazifalariga e'tibor qaratish imkonini beradi va tizimning umumiy ishlashi va mashtablashni(scalability) yaxshilaydi.



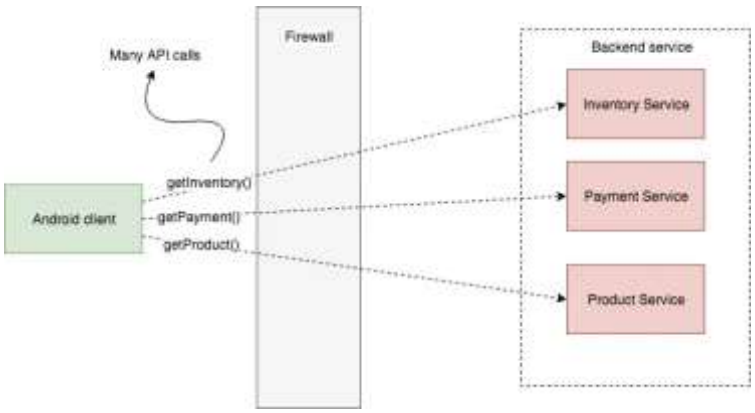
Request Routing

API Gatewayning asosiy funktsiyalaridan biri so'rovlarni marshrutlashdir. API Gatewayning so'rovlarni mos keladigan servislarga yo'naltirish orqali ba'zi API operatsiyalarini amalga oshiradi. So'rovni qabul qilganda, API Gateway requestni qaysi servisga yo'naltirish kerakligini ko'rsatadigan marshrutlash xaritasiga murojaat qiladi.

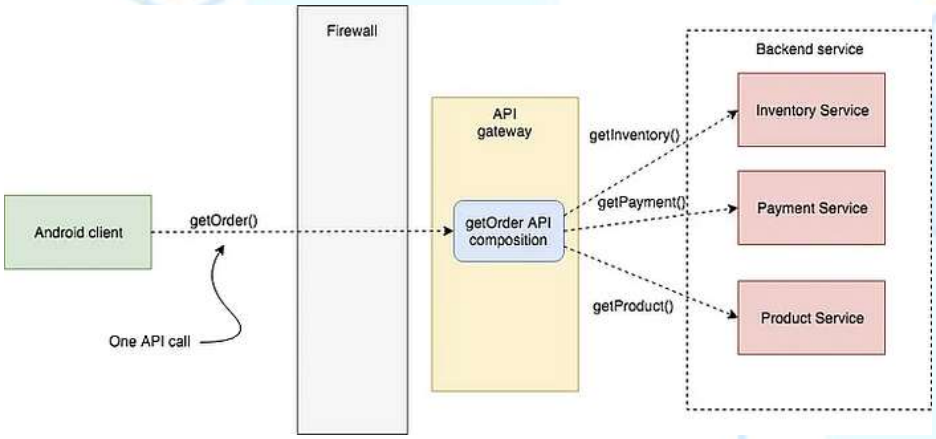
API Composition

API Gateway API tarkibini ham ta'minlaydi. Men buni bazi bir misol yordamida tushuntiraman.

ISSN 2181-3187



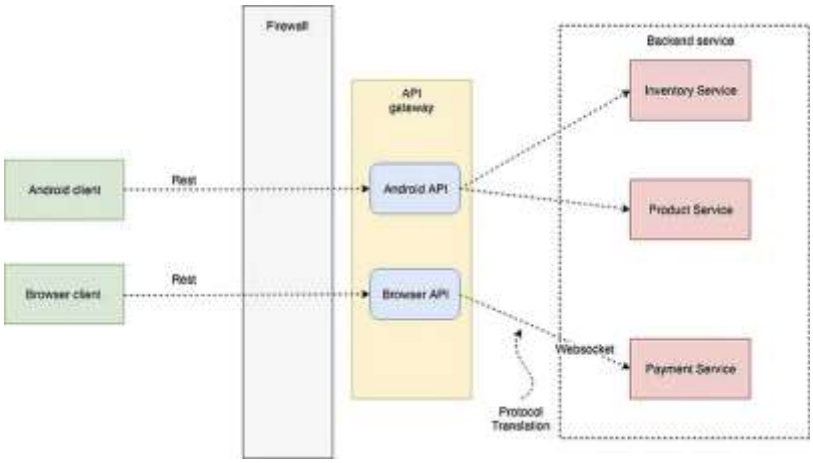
Yuqoridagi rasmda ko’rsatilganidek, Android klient bir nechta API so’rovlarini amalga oshiryapti.



Yuqoridagi rasmda ko’rsatilganidek, API Gateway API tarkibini taqdim etadi, bu esa android klientga bitta API so’rovi yordamida ma’lumotlarni samarali tarzda olish imkonini beradi.

Protocol Translation

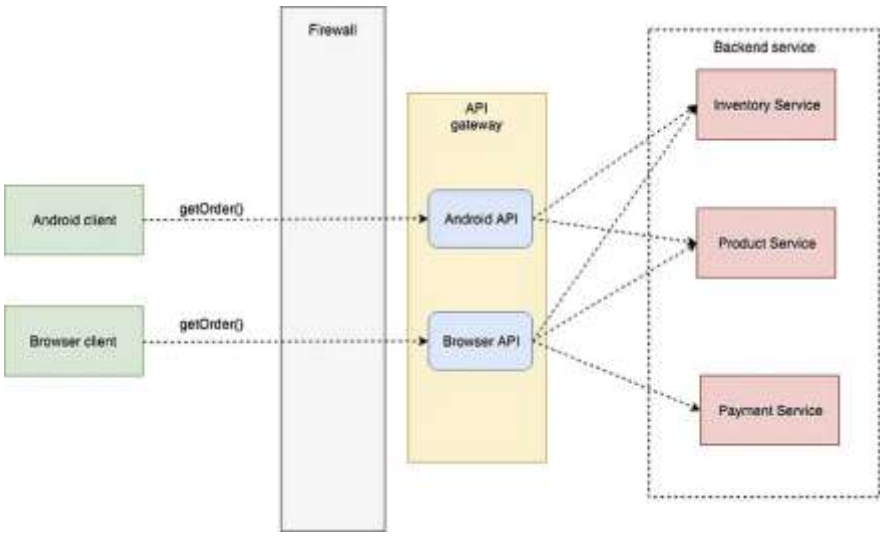
API Gateway shuningdek, protokol tarjimasini ham ta’minlaydi. Application servislari ichki protokollar yig’indisidan , jumladan REST va gRPCdan foydalansa ham, u tashqi mijozlarga RESTful API taqdim etishi mumkin. Zarur bo’lganda, ba’zi API operatsiyalarini amalga oshirish RESTful tashqi API va ichki gRPC asoslangan API o’rtasida ma’lumotlarni almashina oladi.



API Gateway har bir mijozga xos va mos API lar bilan ta’minlaydi

ISSN 2181-3187

Bitta yagona API bilan bog'liq muammo shundaki, turli mijozlar ko'pincha turli talablarga ega. Masalan, getOrder API operatsiyasi mahsulot ma'lumotlarini, to'lov ma'lumotlarini va inventar ma'lumotlarini qaytaradi. Ba'zi hollarda, barcha mijozlar barcha ma'lumotlarga muhtoj emas. Aytaylik, mobil mijozga faqat ma'lumotlarning bir qismi kerak. Bu holatda Yechim API Gatewayni har bir mijozga o'z APIsi bilan ta'minlaydi. Masalan, API Gateway getOrder Android, iOS va brauzer mijozi uchun turli xil APIlarni taqdim etishi mumkin.



API Gateway Patternning asosiy xususiyatlari

API Gateway **bir nechta afzalliklarni** beradi . Mijoz so'rovini bajarish va ichki mikroservislarga yo'naltirishimiz sababli, biz **API Gateway** ni ba'zi foydali xususiyatlardan foydalanishimiz mumkin bo'ladi .

Xususiyatlarni ko'rib chiqaylik.

Routing: API Gateway mijozlardan so'rovlarni qabul qiladi va ularni tegishli mikroservisga yo'naltiradi. Bu mijozlarga bitta kirish nuqtasi orqali turli xil mikroservislarga kirish imkonini beradi, bu esa umumiy tizim dizaynini soddalashtiradi.

Security: API Gateway mijozlarni autentifikatsiya qilish va mikroservislar uchun kirishni boshqarish siyosatini amalga oshirish uchun ishlatilishi mumkin. Bu mikroservislarga faqat vakolatli mijozlar kirishini ta'minlashga yordam beradi va ruxsatsiz kirishning oldini olishga yordam beradi.

Transforming requests and responses : API Gateway turli mijozlarning so'rovlarini qondirish yoki turli xil backend arxitekturasiga mos kelish uchun kiruvchi so'rovlarni va chiquvchi javoblarni backendda o'zgartirishi mumkin.

Rate limiting: Siz API Gateway bilan mijozning mikroservislarga kirishini cheklashingiz mumkin. Bu xizmat hujumlarini bartaraf etishi va boshqa turdagi zararli xatti-harakatlarning oldini olishga yordam beradi.

ISSN 2181-3187

Load balancing: API Gateway kiruvchi so‘rovlarni mikroservisning bir nechta namunalari(instances) o‘rtasida taqsimlashi mumkin, bu tizimda ko‘proq so‘rovlarni boshqarish imkonini beradi va uning samaradorligini oshiradi.

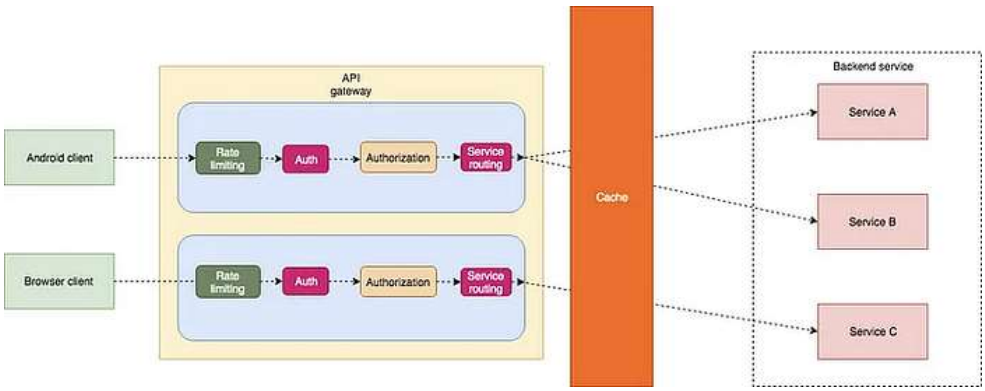
Caching: API Gateway mikroservislar javoblarni keshlashi mumkin va buning natijasida mikroservislarga yuborilishi kerak bo‘lgan so‘rovlar sonini kamaytiradi va tizimning umumiy ish faoliyatini yaxshilaydi.

Serverless execution: API Gateway boshqa servislar bilan integratsiyalasha oladi.

Circuit breaker: API Gateway kaskadli nosozliklardan himoyalanish va tizimingiz chidamliligini oshirishga yordam beradigan *breaker patternlarni* amalga oshirish uchun ishlatilishi mumkin.

Reverse proxy: API Gateway kiruvchi so‘rovlarni so‘rov yo‘li yoki boshqa mezonlar asosida tegishli backend xizmatiga yo‘naltiruvchi teskari **proksi-server** sifatida ishlashi mumkin.

API versioning: API Gateway API versiyasini amalga oshirish uchun ishlatilishi mumkin, bu sizga APIning bir nechta versiyasini saqlash va bir versiyadan ikkinchisiga o‘tishni boshqarish imkonini beradi.



API Gateway ishonchli bo’lishi kerak. Buning uchun load balancer oqali gateway ni bir nechta nusxalarini(replicalari) ishga tushirish kerak. Agar bitta gateway o’chib qolsa yoki nimadur bo’lsa, load balancer so’rovlarni boshqa instancega(gateway) yo’naltiradi.

REFERENCES

1. Qulmatov Qurvonali Zokirali o'g'li, . 2Olimjonov O. O. o'g'li . (2023). MIKROSERVISLAR VA KLIENT-SERVER ARXITEKTURALARINING TAHLILI. https://doi.org/10.5281/zenodo.7860908
2. Building Microservices: Designing Fine-Grained Systems by Sam Newman Paperback [64-108 p].
3. Monolith to Microservices: Refactoring Approaches compared: Transforming Applications to could-ready Software Architectures by Jonas Fritzsch /Apr 24, 2018 [64-80 p]

ISSN 2181-3187

4. Building Event-Driven Microservices: Leveraging Organizational Data at Scale 1st Edition by Adam Bellemare 2020 [189-192 p]
5. https://docs.webmethods.io/api/10.11.0/webmethods_api_cloud__api_gateway_user_s_guide/chapter1/#gsc.tab=0