Sadržaj

Predgovor ..... **xvii**

Deo I. Osnova

1. Šta su to mikroservisi? **3**

Prvi pogled na mikroservise ____ 3
Ključni koncepti mikroservisa ____ 6
Mogućnost nezavisnog raspoređivanja ____ 6
Modelovanje prema poslovnom domenu ____ 7
Vlasnici vlastitog stanja ____ 8
Veličina ____ 9
Fleksibilnost ____ 10
Usklađivanje arhitekture i organizacije ____ 10
Monolit ____ 14
Jednoprocesni monolit ____ 15
Modularni monolit ____ 15
Distribuirani monolit ____ 17
Monoliti i nadmetanje isporuke ____ 17
Prednosti monolita ____ 17
Podržavajuća tehnologija ____ 18
Agregacija evidencije i distribuirano praćenje ____ 19
Kontejneri i Kubernetes ____ 20
Striming ____ 21
Javni oblak i bez servera ____ 21
Prednosti mikroservisa ____ 22
Heterogenost tehnologije ____ 22
Robustnost ____ 23

Skaliranje 24
Lakoća raspoređivanja 25
Organizaciono usaglašavanje 25
Mogućnost kombinovanja 25
Bolne tačke mikroservisa 26
Iskustva programera 26
Preopterećenje tehnologije 26
Cena 27
Izveštavanje 27
Nadgledanje i rešavanje grešaka 28
Bezbednost 29
Testiranje 29
Latencija 30
Konzistentnost podataka 30
Da li da koristim mikroservise? 30
Za koga bi mogli da ne valjaju 31
Gde dobro odgovaraju 32
Rezime 33
2. Kako modelovati mikroservise 35
Upoznajte MusicCorp 35
Šta je dobra granica za mikroservis? 36
Sakrivanje informacija 36
Kohezija 37
Povezanost 38
Međusobno delovanje povezanosti i kohezije 38
Vrste povezanosti 39
Domenska povezanost 41
Posrednička povezanost 43
Povezanost preko zajedničkih prostora 46
Povezanost preko sadržaja 49
Tek toliko dizajna vođenog domenom 51
Zajednički jezik 52
Agregat 53
Ograničeni kontekst 55
Preslikavanje agregata i ograničenih konteksta na mikroservise 58
Diskusija o događajima 59
Slučaj dizajna zasnovanog na domenu za mikroservise 61
Alternative za granice poslovnog domena 62

Nepostojanost 62
Podaci 64
Tehnologija 65
Organizacioni 66
Mešanje modela i izuzeci 68
Rezime 69
3. Podela monolita 71
Imajte cilj 71
Inkrementalna migracija 72
Monolit je retko neprijatelj 73
Opasnosti preuranjenog razlaganja 73
Šta najpre podeliti? 74
Razlaganje po slojevima 76
Najpre kôd 77
Najpre podaci 78
Korisni šabloni razlaganja 79
Šablon Fikus davitelj 79
Paralelno izvršavanje 80
Preklopnik funkcija 80
Problemi razlaganja podataka 81
Performanse 81
Integritet podataka 84
Transakcije 84
Alatke 85
Baza podataka za izveštavanje 85
Rezime 86
4. Stilovi komunikacije mikroservisa 89
Od unutar procesa do između procesa 89
Performanse 90
Menanje interfejsa 91
Obrada grešaka 91
Tehnologija za komunikaciju među procesima: toliko izbora 93
Stilovi komunikacije mikroservisa 93
Mešanje i usklađivanje 95
Šablon: sinhrono blokirajući 95
Prednosti 96
Nedostaci 96
Gde da se koristi 96

Šablon: asinhrono neblokirajući 98
Prednosti 99
Nedostaci 99
Gde da se koristi 101
Šablon: komunikacija preko zajedničkih podataka 101
Implementacija 102
Prednosti 103
Nedostaci 103
Gde da se koristi 104
Šablon: komunikacija zahtev-odgovor 104
Implementacija: sinhroni i asinhroni 106
Gde da se koristi 108
Šablon: komunikacija vođena događajima 108
Implementacija 110
Šta je to događaj? 111
Gde da se koristi 115
Nastavite oprezno 116
Rezime 117

Deo II. Implementacija

5. Implementiranje komunikacije mikroservisa 121
Traženje idealne tehnologije 121
Olakšati kompatibilnost unazad 121
Eksplicitan interfejs 122
Neka vaš API ne bude veran nijednoj tehnologiji 122
Neka naš servis bude jednostavan za konzumente 122
Sakrijte unutrašnje detalje implementacije 123
Izbori tehnologije 123
Pozivi udaljenih procedura 123
REST 127
GraphQL 132
Brokeri poruka 135
Formati serijalizacije 140
Tekstualni formati 140
Binarni formati 141
Šeme 141
Strukturno i semantičko kršenje ugovora 142

Treba li koristiti šeme? 142
Upravljanje promenama među mikroservisima 143
Izbegavanje značajnih promena 144
Promene proširenjem 144
Tolerantno čitanje 145
Prava tehnologija 146
Eksplicitan interfejs 146
Rano hvatanje slučajnih značajnih promena 148
Upravljanje promenama koje izazivaju poremećaje 149
Istovremeno raspoređivanje 149
Koegzistiranje nekompatibilnih verzija mikroservisa 149
Emuliranje starog interfejsa 150
Koji pristup mi se više sviđa? 152
Društveni ugovor 152
Praćenje korišćenja 153
Ekstremne mere 154
Ne ponavljaj se – i opasnosti višekratnog korišćenja koda u svetu mikroservisa 154
Deljenje koda putem biblioteka 155
Otkrivanje servisa 157
DNS (Domain Name System) 157
Dinamički registri servisa 159
Ne zaboravite ljude! 161
Mreže servisa i API mrežni prolazi 161
API mrežni prolazi 163
Mreže servisa 165
A drugi protokoli? 169
Dokumentovanje servisa 169
Eksplicitne šeme 169
Sistem koji sam sebe opisuje 170
Rezime 172
6. Radni tok 175
Transakcije baza podataka 175
ACID transakcije 176
Još uvek ACID, ali nedovoljno atomičnosti? 177
Distribuirane transakcije — potvrđivanja u dve faze 179
Distribuirane transakcije — samo kažite ne 181
Sage 182
Režimi otkaza za sagu 184

Implementiranje saga 188
Sage nasuprot distribuiranim transakcijama 194
Rezime 194
7. Građenje 197
Kratak uvod u neprekidnu integraciju 197
Da li zaista vršite neprekidnu integraciju? 198
Modeli grananja 199
Cevovodi gradnje i neprekidno isporučivanje 200
Alati 203
Kompromisi i okruženja 203
Kreiranje artefakta 204
Preslikavanje izvornog koda i gradnje u mikroservise 205
Jedan džinovski repozitorijum, jedna džinovska gradnja 205
Šablon: jedan repozitorijum po mikroservisu (tzv. Multirepo) 207
Šablon: Monorepo 210
Koji pristup da prihvatim? 216
Rezime 217
8. Raspoređivanje 219
Od logičkog ka fizičkom 219
Više instanci 220
Baza podataka 221
Okruženja 225
Principi raspoređivanja mikroservisa 228
Izolovano izvršavanje 228
Fokus na automatizaciji 231
Infrastruktura kao kôd (IAC) 232
Raspoređivanje bez zastoja rada 233
Upravljanje željenim stanjem 234
Opcije raspoređivanja 237
Fizičke mašine 238
Virtuelne mašine 239
Kontejneri 241
Aplikacioni kontejneri 246
Platforma kao servis (PaaS) 248
Funkcija kao servis (FaaS) 249
Koja opcija raspoređivanja je prava za vas? 256
Kubernetes i orkestracija kontejnera 259
Slučaj za orkestraciju kontejnera 259

Uprošćen prikaz Kubernetes koncepata 260
Više zakupaca i federacija 262
The Cloud Native Computing Federation 264
Platforme i prenosivost 265
Helm, Operator i CRD-ji, O Bože! 266
I Knative 267
Budućnost 267
Treba li ga koristiti? 268
Postupna isporuka 269
Odvajanje raspoređivanja od izdanja 269
Napred na postupnu isporuku 270
Preklopnik funkcija 270
„Canary" izdanje 271
Paralelno izvršavanje 272
Rezime 273
9. Testiranje 275
Vrste testova 276
Domet testa 278
Jedinični testovi 279
Servisni testovi 280
Testovi sa kraja na kraj 281
Kompromisi 282
Implementiranje servisnih testova 283
Lažiranje ili zamenjivanje 283
Pametniji zamenski servis 284
Implementiranje (tih nezgodnih) testova sa kraja na kraj 285
Nekonzistentni i osetljivi testovi 286
Ko piše te testove sa kraja na kraj? 287
Koliko treba da traje izvršavanje testa sa kraja na kraj? 289
Veliko nagomilavanje 290
Metaverzija 290
Nedostatak mogućnosti nezavisnog testiranja 291
Treba li izbegavati testove sa kraja na kraj? 291
Testovi ugovora i ugovori vođeni konzumentima 292
Konačna reč 295
Iskustvo programera 296
Od predprodukcionog do produkcionog testiranja 297
Vrste testiranja u produkciji 297

Omogućavanje bezbednog testiranja u produkciji 298
Prosečno vreme do popravke i prosečno vreme između dva kvara? 298
Međufunkcijsko testiranje 299
Testovi performansi 300
Testovi robustnosti 302
Rezime 302
10. Od nadgledanja do opservabilnosti 305
Poremećaj, panika i pometnja 305
Jedan mikroservis, jedan server 306
Jedan mikroservis, više servera 307
Više servisa, više servera 308
Opservabilnost ili nadgledanje 309
Stubovi opservabilnosti? Samo polako 310
Gradivni blokovi za opservabilnost 311
Agregacija dnevnika 312
Agregacija metrike 321
Distribuirano praćenje 324
Ide li nam dobro? 326
Upozoravanje 329
Semantičko nadgledanje 332
Testiranje u produkciji 334
Standardizacija 337
Biranje alatki 337
Demokratski 338
Lako se integriše 338
Pruža kontekst 338
U realnom vremenu 339
Pogodan za vaše razmere 339
Ekspert u mašini 340
Početak 341
Rezime 342
11. Bezbednost 345
Osnovni principi 346
Princip najmanjih privilegija 347
Odbrana u dubini 347
Automatizacija 349
Ugradnja bezbednosti u proces isporuke 349
Pet funkcija sajber bezbednosti 350

Prepoznati 351
Zaštititi 352
Otkriti 352
Reagovati 353
Oporaviti 353
Osnove bezbednosti aplikacija 354
Ovlašćenja 354
Zakrpe 360
Rezervno kopiranje 363
Ponovna izgradnja 364
Implicitno poverenje i nulto poverenje 365
Implicitno poverenje 365
Nulto poverenje 366
To je spektar 367
Obezbeđivanje podataka 368
Podaci u tranzitu 369
Podaci u mirovanju 371
Autentifikacija i ovlašćivanje 374
Autentifikacija servis-servis 375
Autentifikacija ljudi 375
Uobičajene implementacije jednokratnog prijavljivanja 375
Mrežni prolaz za jednokratno prijavljivanje 376
Fino raspodeljeno ovlašćivanje 378
Problem zbunjenog zastupnika 379
Centralizovano uzvodno ovlašćivanje 380
Decentralizovanje ovlašćivanja 381
JSON Web Tokeni 381
Rezime 385
12. Samoodrživost 387
Šta je to samoodrživost? 387
Robustnost 388
Elastičnost 389
Elegantna proširivost 390
Trajna prilagodljivost 390
I mikroservisna arhitektura 391
Otkazi su svuda 391
Šta je to previše? 392
Degradiranje funkcionalnosti 394

Šabloni stabilnosti 395
Tajmauti 397
Ponovni pokušaji 399
Pregrade 400
Prekidači 401
Izolovanje 404
Redundantnost 405
Posrednički softver 405
Idempotentnost 406
Rasprostiranje rizika 407
Teorema CAP 408
Žrtvovanje konsiztentnosti 410
Žrtvovanje dostupnosti 410
Žrtvovanje otpornosti na razdvajanje? 411
AP ili CP? 411
Ipak to nije sve ili ništa 412
I realni svet 412
Inženjering haosa 413
Dani igre 414
Produkcioni eksperimenti 414
Od robustnosti i dalje 415
Krivica 415
Rezime 417
13. Skaliranje 419
Četiri ose skaliranja 419
Vertikalno skaliranje 420
Horizontalno dupliciranje 422
Podela podataka na particije 426
Funkcionalna dekompozicija 430
Kombinovanje modela 432
Počnite od malog 433
Keširanje 435
Za performanse 436
Za skaliranje 436
Za robustnost 436
Gde da keširamo 437
Poništavanje 442
Zlatno pravilo keširanja 447

Ažurnost nasuprot optimizacije 447
Trovanje keša: upozoravajuća priča 448
Automatsko skaliranje 449
Opet od početka 450
Rezime 451

Deo III. Ljudi

14. Korisnički interfejsi 455
Prema digitalnom 456
Modeli nadležnosti 456
Pobude za namenske čeone timove 458
Ka timovima usklađenim sa tokom 459
Deljenje specijalista 460
Obezbeđenje doslednosti 461
Rad na tehničkim izazovima 462
Šablon: Monolitni čeoni prikaz 463
Kada se koristi 464
Šablon: Mikro čeoni prikaz 464
Implementacija 465
Kada se koristi 465
Šablon: Dekompozicija na osnovu stranice 466
Gde se koristi 468
Šablon: Dekompozicija na osnovu vidžeta 468
Implementacija 470
Kada se koristi 472
Ograničenja 473
Šablon: Centralni agregacioni prolaz 474
Nadležnost 476
Različite vrste korisničkih interfejsa 476
Više briga 478
Kada se koristi 478
Šablon: Pozadina za čeoni prikaz 479
Koliko BFF-ova? 480
Višekratno korišćenje i BFF-ovi 483
BFF-ovi za stoni veb i nadalje 485
Kada se koristi 487
GraphQL 487

Hibridan pristup 488
Rezime 489
15. Organizacione strukture 491
Labavo povezane organizacije 491
Konvejev zakon 492
Dokazivanje 493
Veličina tima 494
Tumačenje Konvejevog zakona 495
Mali timovi, velika organizacija 496
O autonomnosti 497
Jaka i kolektivna nadležnost 499
Jaka nadležnost 499
Kolektivna nadležnost 500
Na nivou tima i na nivou organizacije 501
Odmeravanje modela 502
Timovi osposobljavanja 502
Zajednice prakse 504
Platforma 505
Deljeni mikroservisi 508
Teško je podeliti 508
Unakrsne promene 508
Uska grla isporuke 509
Interni otvoreni kôd 510
Uloga glavnih izvršilaca 510
Zrelost 511
Alati 511
Priključivi, modularni mikroservisi 511
Pregledi promena 514
Servis siroče 517
Studijski primer: realestate.com.au 518
Geografska distribucija 520
Konvejev zakon u obrnutom smeru 521
Ljudi 522
Rezime 523
16. Razvojni arhitekta 525
Šta znači ime? 525
Šta je to softverska arhitektura? 527
Omogućavanje promena 528

Razvojna vizija arhitekte 529
Definisanje granica sistema 530
Društveni pojam 533
Udobnost 534
Principijelan pristup 535
Strateški ciljevi 536
Principi 536
Prakse 537
Kombinovanje principa i praksi 537
Primer iz realnog sveta 537
Usmeravanje evolutivne arhitekture 538
Arhitektura u organizaciji usklađenoj sa tokom 539
Izgradnja tima 541
Zahtevani standard 542
Nadgledanje 542
Interfejsi 543
Bezbednost arhitekture 543
Upravljanje i utabani put 543
Uzori 544
Skrojen šablon mikroservisa 545
Utaban put u velikim razmerama 546
Tehnički dug 546
Rukovanje izuzecima 547
Rezime 548

Pogovor: Sve zajedno 551

Glosar 563

Indeks 569

Predgovor

Mikroservisi predstavljaju pristup distribuiranim sistemima koji podstiče upotrebu sitno granuliranih servisa koji mogu nezavisno da se menjaju, raspoređuju i izdaju. Za organizacije koje idu ka slabije povezanim sistemima, sa autonomnim timovima koji isporučuju funkcionalnost usmerenu na korisnika, mikroservisi neverovatno dobro funkcionišu. Osim toga, mikroservisi nam nude veliki broj opcija za izgradnju sistema, što nam pruža mnogo fleksibilnosti koja omogućava da naš sistem može da se menja prema potrebama korisnika.

Ipak, mikroservisi nisu bez značajnih nedostataka. Kao distribuiran sistem, oni uvode mnoštvo složenosti, od kojih su mnoge možda nove čak i za iskusnog programera.

Iskustvo ljudi širom sveta, pored pojave novih tehologija, ima dubok uticaj na način na koji se mikroservisi koriste. U ovoj knjizi su prikupljene ove ideje, uz konkretne primere iz realnog sveta, što će vam pomoći da utvrdite da li su mikroservisi pravo rešenje za vas.

Ko treba da čita ovu knjigu

Opseg knjige *Izgradnja mikroservisa* je širok, pošto je domet arhitekture mikroservisa takođe širok. Kao takva, ova knjiga bi trebalo da privuče ljude koje zanimaju aspekti dizajna, programiranja, raspoređivanja, testiranja i održavanja sistema. Oni među vama koji su se već ukrcali na putovanje ka sitnije granuliranim arhitekturama, bilo za grinfild aplikaciju ili kao deo razlaganja postojećeg, monolitnijeg sistema, naći će mnogo praktičnih saveta koji će im pomoći. Ova knjiga će takođe pomoći onima među vama koje zanima oko čega se digla sva ta halabuka, da biste mogli da utvrdite da li su mikroservisi prava stvar za vas.

Zašto sam napisao ovu knjigu

S jedne strane, napisao sam ovu knjigu jer sam hteo da proverim da li su informacije iz prvog izdanja još uvek ažurne, tačne i korisne. Napisao sam prvo izdanje zato što su postojale stvarno zanimljive ideje koje sam hteo da podelim. Imao sam sreću da sam bio u poziciji da imam vremena i podršku za pisanje prvog izdanja, i da sam bio u stanju da to radim sa prilično nepristrasnog stanovišta jer nisam radio za velikog

tehnološkog prodavca. Ja nisam prodavao rešenje, pa sam se nadao da ne *prodajem* ni mikroservise — prosto mi se ideja činila očaravajuća i privlačilo me da raspakujem koncept mikroservisa i pronalazim načine da ih šire podelim.

Kada stvarno pogledam suštinu, napisao sam drugo izdanje iz dva glavna razloga. Najpre, imao sam osećaj da ću to sada bolje uraditi. Više sam naučio i nadam se da sam malo napredovao kao pisac. Ali sam ovo drugo izdanje napisao i zato što sam odigrao jednu malu ulogu u pomaganju da ove ideje postanu vodeće, i kao takav imam dužnost da se pobrinem da budu predstavljene na razuman, uravnotežen način. Mikroservisi su za mnoge postali podrazumevan izbor arhitekture. To je nešto što mi se čini da je teško opravdati, pa sam želeo šansu da objasnim zašto je tako.

Ova knjiga nije *za* mikrosevise, niti je *protiv* mikroservisa. Samo želim da budem siguran da sam pravilno istražio kontekst u kojem ove ideje dobro funkcionišu i izneo probleme koje mogu da izazovu.

Šta se promenilo posle prvog izdanja?

Prvo izdanje *Izgradnje mikroservisa* sam napisao za približno godinu dana, počeo sam na početku 2014, a knjiga je izdata februara 2015. godine. To je bilo na početku priče o mikroservisima, bar u smislu svesti šire industrije o tom terminu. Od tada su mikroservisi postali vodeći na način koji nisam mogao da predvidim. Sa tim porastom dolazi i mnogo širi skup iskustava kao oslonac, i više tehnologija za istraživanje.

Pošto sam nakon prvog izdanja radio sa više timova, počeo sam da usavršavam svoje razmišljanje u vezi sa nekim idejama o mikroservisima. U nekim slučajevima je to značilo da su ideje koje su bile tek na periferiji mog razmišljanja, kao što je sakrivanje informacija, postale jasnije kao fundamentalni koncepti koje treba bolje istaći. U drugim područjima, nova tehnologija je predstavila i nova rešenja i nove probleme za naše sisteme. Kad sam video koliko ljudi hrli ka Kubernetesu u nadi da će on rešiti sva njihova pitanja sa arhitekturama mikroservisa, to mi je svakako dalo povoda za razmišljanje.

Pored toga, napisao sam prvo izdanje *Izgradnje mikroservisa* da bih pružio ne samo objašnjenje mikroservisa, već i širi pogled na to kako ovaj pristup arhitekturi menja aspekte razvijanja softvera. Zato, dok sam dublje sagledavao aspekte bezbednosti i samoodrživosti, utvrdio sam da želim da se vratim i proširim te teme koje su sve važnije za razvoj savremenog softvera.

Tako, u ovom drugom izdanju provodim više vremena na eksplicitnim primerima kako bih bolje objasnio te ideje. Svako poglavlje je preispitano i svaka rečenica pregledana. Od prvog izdanja nije ostalo mnogo u smislu konkretnog teksta, ali su sve ideje još uvek tu. Pokušao sam da jasnije iznesem vlastita mišljenja, i dalje priznajući da često postoji više načina da se problem reši. To je značilo proširenje diskusije o komunikaciji među procesima, koja je sada podeljena na tri poglavlja. Takođe posvećujem više vremena

razmatranju posledica tehnologija kao što su kontejneri, Kubernetes, i bez servera; zbog toga sada imamo zasebna poglavlja o izgradnji i o raspoređivanju.

Nadao sam se da ću napraviti knjigu približno istih dimenzija kao prvo izdanje, ali tako da nađem načina da unesem više ideja. Kao što vidite, nisam postigao svoj cilj— ovo izdanje je veće! Ali mislim da sam uspeo da budem jasnji u načinu na koji izlažem te ideje.

Organizacija knjige

Knjiga je pre svega organizovana u formatu zasnovanom na temama. Strukturisao sam i pisao knjigu kao da ćete je čitati od početka do kraja, ali naravno, možda ćete želeti da skočite na konkretne teme koje vas najviše zanimaju. Ako rešite da zaronite u konkretno poglavlje, možda će vam Glosar pri kraju knjige biti od pomoći u objašnjavanju novih ili nepoznatih pojmova. Što se tiče terminologije, ja naizmenično koristim izraze mikroservis i servis u celoj knjizi; možete da smatrate da se oba termina odnose na istu stvar, ukoliko eksplicitno ne navedem nešto drugo. Neke od ključnih saveta iz knjige takođe sumiram u Pogovoru ako *zaista* prosto skočite na kraj — ali imajte na umu da ćete u tom slučaju propustiti *mnogo* detalja!

Glavni deo knjige je podeljen na tri zasebna dela — Osnova, Implementacija i Ljudi. Pogledajmo šta obuhvata svaki od tih delova.

Deo I, *Osnova*

U ovom delu, detaljno opisujem neke osnovne ideje koje stoje u osnovi mikroservisa.

Poglavlje 1, Šta su to mikroservisi?

> Ovo je opšti uvod u mikroservise u kojem ukratko opisujem jedan broj tema koje proširujem kasnije u knjizi.

Poglavlje 2, Kako modelovati mikroservise

> U ovom poglavlju istražujem značaj koncepata kao što su sakrivanje informacija, povezanost, kohezija i dizajn zasnovan na domenu u traženju pravilnih granica za naše mikroservise.

Poglavlje 3, Podela monolita

> Ovo poglavlje sadrži neke smernice o tome kako da se uzme postojeća monolitna aplikacija i podeli na mikroservise.

Poglavlje 4, Stilovi komunikacije mikroservisa

> Poslednje poglavlje u ovom delu sadrži opis različitih vrsta komunikacije mikroservisa, uključujući asinhrone i sinhrone pozive i stil saradnje zahtev--odgovor i onaj vođen događajima.

Deo II, *Implementacija*

Prelazeći od koncepata višeg nivoa na detalje implementacije, u ovom delu razmatramo tehnike i tehnologije koje mogu da vam pomognu da izvučete maksimum iz mikroservisa.

Poglavlje 5, Implementiranje komunikacije mikroservisa

> U ovom poglavlju dublje razmatramo konkretne tehnologije koje se koriste za implementaciju komunikacije među mikroservisima.

Poglavlje 6, Radni tok

> Ovo poglavlje nudi poređenje saga i distribuiranih transakcija i objašnjava kako su one korisne u modelovanju poslovnih procesa koje uključuje više mikroservisa.

Poglavlje 7, Građenje

> U ovom poglavlju se objašnjava mapiranje iz mikroservisa u repozitorijume i gradnje.

Poglavlje 8, Raspoređivanje

> U ovom poglavlju ćete naći opise mnoštva opcija za raspoređivanje mikroservisa, uključujući kontejnere, Kubernetes i FaaS.

Poglavlje 9, Testiranje

> Ovde govorim o izazovima testiranja mikroservisa, uključujući pitanja koja slede iz testiranja sa kraja na kraj, i kako mogu da pomognu ugovori vođeni konzumentima i testiranje u produkciji.

Poglavlje 10, Od nadgledanja do opservabilnosti

> Ovo poglavlje pokriva prelazak sa fokusiranja na aktivnosti statičkog nadgledanja na šire razmišljanje o tome kako da se unapredi opservabilnost mikroservisnih arhitektura, uz neke konkretne preporuke u vezi sa alatima.

Poglavlje 11, Bezbednost

> Mikroservisne arhitekture mogu da povećaju površinu izloženu napadima ali nam takođe daju više prilike za temeljnu odbranu. U ovom poglavlju ispitujemo to balansiranje.

Poglavlje 12, Samoodrživost

> Ovo poglavlje nudi širi pogled na pojam samoodrživosti i ulogu koju mikroservisi mogu da odigraju u unapređivanju samoodrživosti naših aplikacija.

Poglavlje 13, Skaliranje

U ovom poglavlju opisujem četiri ose skaliranja i pokazujem kako one mogu da se kombinuju za skaliranje mikroservisne arhitekture.

Deo III, *Ljudi*

Ideje i tehnologija ne znače ništa bez ljudi i organizacije da ih podrže.

Poglavlje 14, Korisnički interfejsi

Od udaljavanja od namenskih čeonih timova pa do upotrebe BFF-ova i GraphQL-a, u ovom poglavlju se istražuje kako mikroservisi i korisnički interfejsi mogu da sarađuju.

Poglavlje 15, Organizacione strukture

Ovo pretposlednje poglavlje usredsređeno je na to kako timovi usklađeni sa tokom i timovi za osposobljavanje mogu da rade u kontekstu mikroservisne arhitekture.

Poglavlje 16, Razvojni arhitekta

Mikroservisne arhitekture nisu statične, pa vaš pogled na arhitekturu sistema možda mora da se promeni — tema koja se u ovom poglavlju detaljno istražuje.

Konvencije koje se koriste u ovoj knjizi

U ovoj knjizi se koriste sledeće tipografske konvencije:

Kurziv

Označava nove izraze, URL-ove, email adrese, imena fajlova, i ekstenzije fajlova.

`Konstantna širina`

Koristi se za listinge programa, kao i unutar pasusa za referenciranje na elemente programa kao što su imena promenljivih ili funkcija, baza podataka, tipova podataka, promenljivih okruženja, iskazi i ključne reči.



Ovaj element znači savet ili predlog.



Ovaj element znači opštu napomenu.



Ovaj element znači upozorenje ili oprez.

O'Reilly Onlajn učenje



Već preko 40 godina, *O'Reilly Media* pruža tehnološku i poslovnu obuku, znanje i uvid kao pomoć kompanijama na putu do uspeha.

Naša jedinstvena mreža eksperata i inovatora deli svoje znanje i stručnost preko knjiga, članaka, konferencija i naše platforme za onlajn učenje. O'Reilly-jeva platforma za onlajn učenje daje vam pristup na zahtev do kurseva obuke uživo, detaljnim putevima učenja, interaktivnim okruženjima za kodiranje i velikoj zbirci tekstova i videa od O'Reilly-ja i više od 200 drugih izdavača. Za više informacija, molim posetite *http://oreilly.com*.

Naši kontakti

Molim da komentare i pitanja o ovoj knjizi uputite izdavaču:

O'Reilly Media, Inc.
1005 Gravenstein Highway North
Sebastopol, CA 95472
800-998-9938 (u SAD ili Kanadi)
707-829-0515 (međunarodno ili lokalno)
707-829-0104 (fax)

Imamo veb stranicu za ovu knjigu na kojoj se nalaze greške, primeri, i neke dodatne informacije. Može joj se pristupiti na *https://oreil.ly/Building_Microservices2*.

Email *bookquestions@oreilly.com* za komentare i tehnička pitanja o ovoj knjizi.

Za novosti i informacije o našim knjigama i kursevima, posetite *http://oreilly.com*.

Nađite nas na Facebooku: *http://facebook.com/oreilly*

Pratite nas na Twitteru: *http://twitter.com/oreillymedia*

Gledajte nas na YouTubu: *http://youtube.com/oreillymedia*

Priznanja

Neprestano sam zapanjen podrškom koju dobijam od svoje porodice, posebno od moje supruge, Lindy Stephens, koja toliko toga podnosi sa neverovatnom ljubaznošću. Reći da ova knjiga ne bi postojala bez nje bilo bi potcenjivanje, čak iako mi ona uvek ne veruje kad to kažem. Ovo je za nju. Takođe je i za mog oca, Jacka, Josie, Kane, i za širi klan Gilmanco Staynes.

Veliki deo ove knjige je napisan tokom globalne pandemije, koja dok pišem ova priznanja još uvek traje. Možda ne znači mnogo, ali želim da se zahvalim Nacionalnoj zdravstvenoj službi, ovde u UK, i svim ljudima na celom svetu koji nas čuvaju radeći na vakcinama, lečeći bolesne, isporučujući hranu i pomažući na hiljade drugih načina koje ne bih shvatio. To je takođe za sve vas.

Ovo drugo izdanje ne bi postojalo bez prvog, pa bih želeo da ponovo zahvalim svima koji su mi pomogli tokom izazovnog procesa pisanja moje prve knjige. Tu su tehnički recenzenti Ben Christensen, Martin Fowler i Venkat Subramaniam; James Lewis, kojem zahvaljujem za naše mnoge prosvećujuće razgovore; O'Reilly-jev tim: Brian MacDonald, Rachel Monaghan, Kristen Brown i Betsy Waliszewski; i izvrsne povratne informacije koje su slali čitaoci Anand Krishnaswamy, Kent McNeil, Charles Haynes, Chris Ford, Aidy Lewis, Will Thames, Jon Eaves, Rolf Russell, Badrinath Janakiraman, Daniel Bryant, Ian Robinson, Jim Webber, Stewart Gleadow, Evan Bottcher, Eric Sword i Olivia Leonard. I hvala Mikeu Loukidesu, valjda, što me je uopšte uvukao u ovo čudo!

Za drugo izdanje, Martin Fowler se ponovo vratio kao tehnički recenzent i pridružili su mu se Daniel Bryant i Sarah Wells, koji su velikodušno poklonili svoje vreme i povratne informacije. Takođe želim da zahvalim Nickyju Wrightsonu i Alexandru von Zitzerwitzu na pomoći da se tehnička revizija prevede preko završne linije. Na O'Reillyjevom frontu, ceo proces je nadgledala moja fenomenalna urednica, Nicole Taché, bez koje bih sigurno izgubio razum; Melissa Duffield, koja izgleda bolje od mene upravlja mojim radom; Deb Baker, Arthur Johnson i ostali iz produkcionog tima (Žao mi je, ne znam sva vaša imena, ali hvala vam!); i Mary Treseler, za upravljanje brodom u nekim teškim vremenima.

Pored toga, drugo izdanje je imalo velike koristi od pomoći i uvida brojnih ljudi, uključujući (bez određenog redosleda) Dave Coombes i tim u Tyro-u, Dave Halsey i tim u Money Supermarketu, Tom Kerkhove, Erik Doernenburg, Graham Tackley, Kent Beck, Kevlin Henney, Laura Bell, Adrian Mouat, Sarah Taraporewalla, Uwe Friedrichse, Liz Fong-Jones, Kane Stephens, Gilmanco Staynes, Adam Tornhill, Venkat Subramaniam, Susanne Kaiser, Jan Schaumann, Grady Booch, Pini Reznik, Nicole Forsgren, Jez Humble, Gene Kim, Manuel Pais, Matthew Skelton, i South Sydney Rabbitohs. Najzad, hteo bih da zahvalim fenomenalnim čitaocima verzije za rani pristup knjizi, koji su dali neprocenjive povratne informacije; to su Felipe de Morais, Mark Gardner, David Lauzon, Assam Zafar, Michael Bleterman, Nicola Musatti, Eleonora Lester, Nathan DiMauro, Daniel Lemke, Soner Eker, Ripple Shah, Joel Lim i Himanshu Pant. I na kraju, pozdrav Jasonu Isaacsu.

Najveći deo ove knjige sam pisao tokom cele 2020. godine i u prvoj polovini 2021. Za veći deo pisanja sam koristio Visual Studio Code na macOS-u, mada sam u retkim slučajevima koristio još i Working Copy na iOS-u. OmniGraffle je korišćen za pravljenje svih dijagrama u knjizi. AsciiDoc je upotrebljen za formatiranje knjige i u celini je bio odličan, a sa O'Reillyjevim lancem alata Atlas postigao je čaroliju da se na drugom kraju pojavi knjiga.

DEO I

Osnova

POGLAVLJE 1

Šta su to mikroservisi?

Mikroservisi su za ovih preko pola decenije od kad sam pisao prvo izdanje ove knjige postali sve popularniji izbor arhitekture. Ne mogu da tvrdim da sam ja zaslužan za potonju eksploziju popularnosti, ali žurba da se upotrebi arhitektura mikroservisa znači da se mnoge ideje koje sam ranije opisao sada isprobavaju i testiraju, nove ideje ulaze u kombinaciju istovremeno dok ranije prakse gube na popularnosti. Zato je ponovo vreme da se istraži suština arhitekture mikroservisa i istaknu osnovni koncepti koji omogućavaju rad mikroservisa.

Ova knjiga je kao celina zamišljena da predoči širok pogled na uticaj koji mikroservisi imaju na razne aspekte isporuke softvera. Za početak, ovo poglavlje se osvrće na osnovne ideje na koje se mikroservisi oslanjaju, prethodne veštine koje su nas dovele dovde i neke razloge zbog kojih su ove arhitekture u tako širokoj upotrebi.

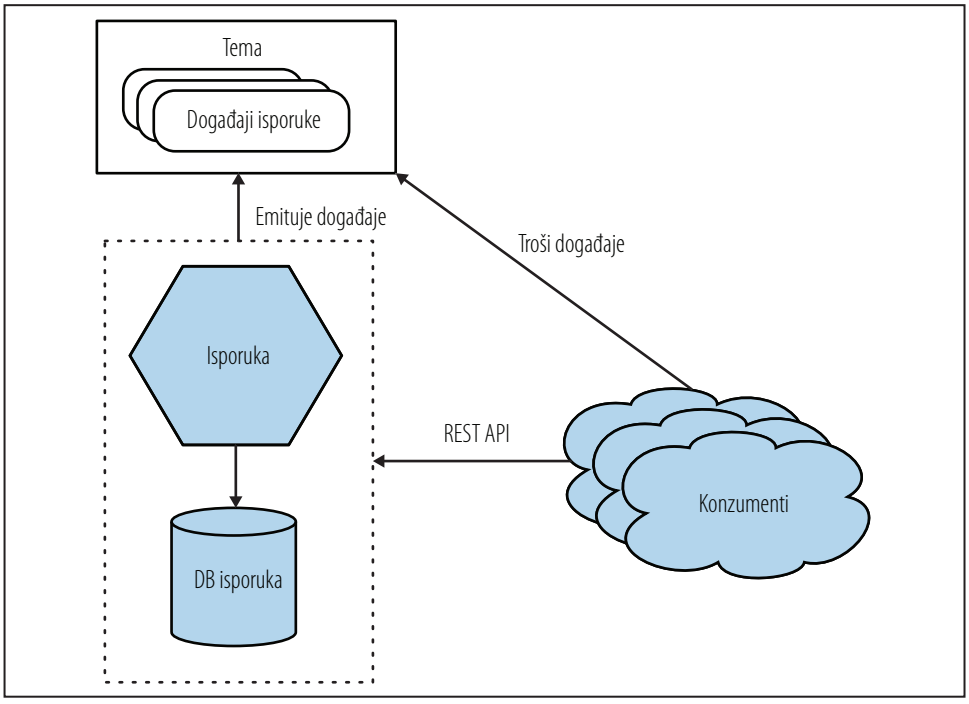
Prvi pogled na mikroservise

Mikroservisi su servisi koji mogu pojedinačno da se isporučuju i modelovani su oko nekog poslovnog domena. Servis enkapsulira jednu funkcionalnost i pruža je na raspolaganje drugim servisima putem mreža – od tih gradivnih blokova se pravi složeniji sistem. Jedan mikroservis bi mogao da predstavlja inventar, drugi upravljanje porudžbinama, a još neki špediciju, ali svi zajedno mogu da čine kompletan sistem e-trgovine. Mikroservisi su izbor arhitekture fokusirane na davanje više opcija za rešavanje problema koji su pred vama.

Oni su jedna *vrsta* arhitekture orijentisane na servise, premda imaju čvrste stavove o povlačenju granica među servisima i o tome gde je najvažnija mogućnost nezavisnog uvođenja. Oni nisu vezani ni za koju tehnologiju, što predstavlja jednu od prednosti koje nude.

Spolja, pojedinačan mikroservis se tretira kao crna kutija. U njemu se nalazi poslovna funkcionalnost na jednoj krajnjoj tački mreže ili na više njih (na primer, red čekanja ili REST API, kao što je prikazano na Slici 1-1), preko bilo kojih najpogodnijih protokola. Konzumenti, bilo da je reč o drugim mikroservisima ili su to druge vrste

programa, pristupaju toj funkcionalnosti preko tih krajnjih tačaka mreže. Unutrašnji detalji implementacije (kao što su tehnologija u kojoj je servis napisan ili način na koji se podaci čuvaju) su potpuno sakriveni od spoljnog sveta. To znači da arhitekture mikroservisa u većini slučajeva izbegavaju upotrebu deljenih baza podataka; umesto toga, svaki mikroservis enkapsulira vlastitu bazu podataka gde je potrebno.



Slika 1-1. Mikroservis izlaže svoju funkcionalnost preko REST API-ja i preko teme

Mikroservisi se pridržavaju koncepta sakrivanja informacija[1]. *Sakrivanje informacija* znači da se što je god više moguće informacija krije unutar komponente i što je manje moguće izlaže putem spoljnih interfejsa. Tako se omogućava jasno razdvajanje onog što može lako da se menja od onog što je teže promeniti. Implementacija koja je sakrivena od spoljnih učesnika može slobodno da se menja sve dok se mrežni interfejsi koje mikroservis izlaže ne menjaju na način koji nije kompatibilan unazad. Promene unutar granica mikroservisa (kao što je prikazano na Slici 1-1) ne bi smele da utiču na uzvodnog potrošača, što omogućava nezavisno isporučivanje funkcionalnosti. Ovo je bitno da bi se omogućilo da se na našim mikroservisima radi izolovano i da se oni isporučuju na zahtev. Kad imate jasne, stabilne granice servisa koje se ne menjaju kada

1 Ovaj koncept je najpre skicirao David Parnas u "Information Distribution Aspects of Design Methodology", *Obrada informacija: Zbornik radova Kongresa IFIP 1971* (Amsterdam: Severna Holandija, 1972), 1:339–44

se promeni unutrašnja implementacija, dobijate sisteme koji imaju slabiju povezanost (engl. *looser coupling*) i jaču koheziju.

Dok govorimo o sakrivanju detalja unutrašnje implementacije, bilo bi sa moje strane neuviđavno da ne pomenem obrazac Šestougaona arhitektura (engl. *Hexagonal Architecture*), koji je prvi detaljno opisao Alistair Cockburn[2]. Ovaj obrazac opisuje značaj odvajanja unutrašnje implementacije od spoljnih interfejsa, uz zamisao da biste možda hteli da sa istom funkcionalnošću komunicirate preko različitih vrsta interfejsa. Ja svoje mikroservise crtam kao šestouglove delimično zato da bi se oni razlikovali od „normalnih" servisa, ali takođe i kao odavanje pošte ovom prethodnom majstorstvu.

Da li su arhitektura orijentisana na servise i mikroservisi dve različite stvari?

Arhitektura orijentisana na servise (*SOA – Service-oriented architecture*) je pristup dizajnu u kojem više servisa sarađuje na obezbeđenju određenog krajnjeg skupa funkcionalnosti. (Ovde *servis* obično znači sasvim zaseban proces operativnog sistema). Komunikacija među tim servisima se vrši pozivima preko mreže, a ne putem poziva metoda unutar granica procesa.

SOA se pojavila kao pristup za borbu protiv izazova velikih, monolitnih aplikacija. Ovaj pristup ima za cilj unapređenje višekratne upotrebe softvera; dve ili više aplikacija za krajnjeg korisnika, na primer, mogle bi da koriste iste servise. SOA ima za cilj da olakša održavanje ili ponovno pisanje softvera, pošto teoretski možemo jedan servis da zamenimo drugim bez ičijeg znanja, pod uslovom da se semantika servisa previše ne menja.

SOA je u suštini razumna zamisao. Međutim, uprkos mnogim naporima, nedostaje dobar konsenzus o tome kako da se SOA uradi *dobro*. Po mom mišljenju, veliki deo industrije propušta da dovoljno holistički posmatra problem i ponudi ubedljivu alternativu narativu koji iznose razni prodavci u tom domenu.

Mnogi problemi koji se zameraju pristupu SOA su u stvari problemi sa stvarima kao što su komunikacioni protokoli (npr., SOAP), posrednički komercijalni softver, nedostatak uputstava o granularnosti servisa, ili pogrešna uputstva o pravilnoj podeli sistema i izboru mesta na kojima se sistem deli. Cinik bi mogao da primeti da su prodavci prihvatili (i u nekim slučajevima usmeravali) pokret SOA kao način da se proda više proizvoda, i takvi isti proizvodi su na kraju podrili cilj SOA. Video sam mnogo primera arhitekture SOA u kojima su se timovi trudili da servisi budu što manji, ali im je sve ipak bilo povezano sa istom bazom podataka, pa su morali sve da raspoređuju zajedno. Orijentisano na servise? Da. Ali to nisu mikroservisi.

Pristup mikroservisa je potekao od upotrebe u realnom svetu, tako što uzima naše bolje razumevanje sistema i arhitekture da dobro vršimo arhitekturu SOA. Trebalo

2 Alistair Cockburn, "Hexagonal Architecture," 4. januar, 2005, *https://oreil.ly/NfvTP*