

Spis treści

przedmowa	xiii
podziękowania	xv
o książce	xvii
o autorach	xx
o ilustracji na okładce	xxi
CZĘŚĆ 1 STAN RZECZY	1
1 Projektowanie i uruchamianie mikroserwisów	3
1.1. Czym jest aplikacja mikroserwisowa?	4
Skalowanie przez dekompozycję	6
Kluczowe zasady	8
Kto używa mikroserwisów?	11
Dlaczego mikroserwisy to dobry wybór?	12
1.2. Co w mikroserwisach jest wyzwaniem?	14
Wyzwania związane z projektowaniem	15
Wyzwania związane z działaniem	17
1.3. Cykl rozwojowy mikroserwisu	18
Projektowanie mikroserwisów	19
Wdrażanie mikroserwisów	21
Obserwowanie mikroserwisów	24
1.4. Odpowiedzialna i operacyjnie świadoma kultura inżynierska	25
2 Mikroserwisy w SimpleBanku	27
2.1. Czym zajmuje się SimpleBank?	28
2.2. Czy mikroserwisy są dobrym wyborem?	29
Ryzyko i inercja w oprogramowaniu finansowym	30
Zmniejszanie tarc i zapewnienie wartości dodanej	31
2.3. Budowanie nowej funkcjonalności	31
Identyfikacja mikroserwisów przez modelowanie domeny	32
Współpraca usług	35
Choreografia usług	36
2.4. Udostępnianie usług na zewnątrz	38
2.5. Wdrażanie funkcjonalności na produkcję	39
Kontrolowane pod względem jakości i zautomatyzowane wdrożenia	41
Odporność	42

Przejrzystość	42
2.6. Skalowanie rozwoju mikroserwisów	44
Techniczna różnorodność	44
Izolacja	45
2.7. Co dalej?	46
CZĘŚĆ 2 PROJEKT	49
3 Architektura aplikacji mikroserwisowej	51
3.1. Architektura jako całość	52
Od monolitów do mikroserwisu	52
Rola architekta	54
Zasady architektoniczne	54
Cztery warstwy aplikacji mikroserwisowej	55
3.2. Platforma mikroserwisu	56
Mapowanie platformy uruchomieniowej	57
3.3. Usługi	58
Zdolności	58
Agregacja i usługi wyższego szczebla	59
Ścieżki krytyczne i niekrytyczne	60
3.4. Komunikacja	61
Kiedy używać komunikatów synchronicznych	61
Kiedy używać komunikatów asynchronicznych	62
Wzorce komunikacji asynchronicznej	63
Lokalizowanie innych usług	64
3.5. Granica aplikacji	66
Bramy API	68
Backends for frontends	68
Bramy consumer-driven	69
3.6. Klienci	71
Monolity frontendowe	71
Mikrofrontendy	72
4 Projektowanie nowych funkcjonalności	74
4.1. Nowa funkcjonalność w SimpleBanku	75
4.2. Określanie zakresu według zdolności biznesowych	77
Zdolności i modelowanie domeny	77
Tworzenie strategii inwestycyjnej	79
Zagnieżdżone konteksty i usługi	85
Wyzwania i ograniczenia	86
4.3. Określanie zakresu według przypadków użycia	86
Składanie zleceń związanych ze strategią inwestycyjną	87
Działania i magazyny	92
Orkiestracja i choreografia	93
4.4. Ograniczanie zakresu według zmienności	94
4.5. Zdolności techniczne	96
Wysyłanie powiadomień	96

Kiedy używać zdolności technicznych	97
4.6. Postępowanie z niejednoznacznością	98
Zacznijmy od gruboziarnistych usług	99
Przygotujmy się na przyszłą dekompozycję	99
Wycofywanie i migracja	100
4.7. Własność usług w firmach	100
5 Transakcje i zapytania w mikroserwisach	105
5.1. Spójne transakcje w aplikacjach rozproszonych	106
Dlaczego nie użyć transakcji rozproszonych?	107
5.2. Komunikacja oparta na zdarzeniach	108
Zdarzenia i choreografia	109
5.3. Sagi	111
Choreografowane sagi	112
Orkiestrowane sagi	115
Przeplatane sagi	117
Wzorce spójności	118
Event sourcing	119
5.4. Zapytania w rozproszonym świecie	120
Przechowywanie kopii danych	122
Oddzielanie zapytań i poleceń	123
Wyzwania związane z CQRS	125
Analizy i raportowanie	127
5.5. Literatura uzupełniająca	128
6 Projektowanie niezawodnych usług	129
6.1. Definiowanie niezawodności	130
6.2. Co może pójść źle?	132
Źródła awarii	133
Awarie kaskadowe	136
6.3. Projektowanie niezawodnej komunikacji	139
Ponowne próby	140
Plany awaryjne	143
Przeterminowania	145
Bezpieczniki	147
Komunikacja asynchroniczna	150
6.4. Maksymalizacja niezawodności usługi	150
Równoważenie obciążenia i stan usługi	150
Limity użycia	152
Walidacja niezawodności i odporność na błędy	153
6.5. Bezpieczeństwo przede wszystkim	156
Frameworki	157
Service mesh	158
7 Budowanie uniwersalnego frameworka mikroserwisowego	160
7.1. Szkielet mikroserwisu	161
7.2. Jaki jest cel szkieletu mikroserwisu?	164

Zmniejszone ryzyko	165
Szybsze rozpoczynanie	165
7.3. Projektowanie szkieletu	166
Wykrywanie usług	168
Obserwowalność	172
Równoważenie i limitowanie	178
7.4. Badanie funkcjonalności zaimplementowanych z użyciem szkieletu	181
7.5. Czy heterogeniczność nie była jedną z obietnic mikroserwisów?	183
CZĘŚĆ 3 WDRAŻANIE	187
8 Wdrażanie mikroserwisów	189
8.1. Dlaczego wdrożenie jest ważne?	190
Stabilność i dostępność	191
8.2. Środowisko produkcyjne mikroserwisów	191
Cechy środowiska produkcyjnego mikroserwisów	192
Automatyzacja i szybkość	193
8.3. Wdrażanie usługi, szybka ścieżka	193
Uruchomienie usługi	194
Budowanie maszyny wirtualnej	194
Uruchamianie wielu instancji usługi	196
Dodawanie równoważenia obciążenia	199
Czego się nauczyliśmy?	200
8.4. Budowanie artefaktów usługi	201
Co jest w artefakcie?	202
Niezmienialność	203
Typy artefaktów usługi	204
Konfiguracja	209
8.5. Modele hostowania usług	210
Jedna usługa per host	210
Wiele statycznych usług per host	210
Wiele zarządzanych usług per host	211
8.6. Wdrażanie usług bez przestojów	212
Wdrożenia kanarkowe i rolling deploy w GCE	213
9 Wdrażanie z użyciem kontenerów i programów zarządzających	217
9.1. Konteneryzowanie usługi	218
Praca z obrazami	219
Budowanie naszego obrazu	221
Uruchamianie kontenerów	224
Przechowywanie obrazu	226
9.2. Wdrażanie do klastra	228
Projektowanie i uruchamianie kapsuł	229
Równoważenie obciążenia	232
Krótkie spojrzenie pod maskę	233
Kontrola działania	236
Wdrażanie nowej wersji	238

Wycofywanie	244
Łączenie wielu usług	244
10 Budowanie strumienia dostarczania dla mikroserwisów	246
10.1. Nudne wdrażanie	247
Strumień wdrażania	247
10.2. Budowanie strumienia w Jenkinsie	249
Konfigurowanie strumienia budowania	250
Budowanie obrazu	254
Uruchamianie testów	255
Publikowanie artefaktów	257
Wdrażanie do środowiska staging	258
Środowiska staging	261
Wdrażanie do produkcji	262
10.3. Budowanie strumienia z krokami wielokrotnego zastosowania	265
Proceduralne versus deklaratywne budowanie strumieni	266
10.4. Techniki wdrażania o niskim oddziaływaniu i wydawanie funkcjonalności	267
Dark launch	267
Flagi funkcjonalności	268
CZĘŚĆ 4 OBSERWOWALNOŚĆ I WŁASNOŚĆ	271
11 Budowanie systemu monitorowania	273
11.1. Kompleksowy stos monitorowania	274
Dobry monitoring jest warstwowy	275
Złote sygnały	276
Typy metryk	277
Rekomendowane praktyki	278
11.2. Monitorowanie SimpleBanku za pomocą Prometheusa	279
Konfigurowanie infrastruktury zbierania metryk	279
Zbieranie metryk dotyczących infrastruktury - RabbitMQ	286
Instrumentacja składania zleceń w SimpleBanku	289
Konfigurowanie alarmów	291
11.3. Ustawianie sensownych i praktycznych alarmów	295
Kto potrzebuje wiedzieć, że coś nie działa?	296
Objawy, nie przyczyny	297
11.4. Obserwowanie całej aplikacji	297
12 Używanie dzienników i śladów do zrozumienia zachowania	300
12.1. Zrozumienie zachowania między wieloma usługami	301
12.2. Generowanie spójnych, uporządkowanych dzienników czytelnych dla człowieka	304
Przydatne informacje do uwzględnienia we wpisach dziennika	304
Struktura i czytelność	305
12.3. Konfigurowanie infrastruktury logowania w SimpleBanku	307
Rozwiązanie oparte na ELK i Fluentd	308

Konfigurowanie systemu do rejestrowania	310
Konfigurowanie tego, co będzie zbierane w dziennikach	312
Znajdowanie igły w stogu siana	316
Rejestrowanie właściwych informacji	317
12.4. Siedzenie interakcji między usługami	317
Korelowanie żądań: ślady i spany	318
Konfigurowanie śledzenia w usługach	320
12.5. Wizualizowanie śladów	324
13 Budowanie zespołów mikroserwisowych	329
13.1. Budowanie efektywnych zespołów	330
Prawo Conwaya	331
Zasady efektywnych zespołów	332
13.2. Modele zespołów	334
Grupowanie funkcyjne	334
Grupowanie przekrojowe	336
Ustalanie granic zespołu	338
Infrastruktura, platforma i produkt	340
Kto jest pod telefonem?	341
Dzielenie się wiedzą	343
13.3. Zalecane praktyki dla zespołów mikroserwisowych	344
Czynniki zmian w mikroserwisach	345
Rola architektury	346
Jednorodność a elastyczność techniczna	348
Model otwartego oprogramowania	348
Przeglądy projektu	350
Żywa dokumentacja	351
Odpowiadanie na pytania dotyczące aplikacji	352
13.4. Dodatkowe materiały	352
dodatek Instalacja Jenkinsa w Minikube'ie	355
indeks	363