

Spis treści

Przedmowa	9
1. Tylko tyle mikrousług, ile potrzeba	13
Czym są mikrousługi?	13
Możliwość niezależnego instalowania	14
Modelowane na podstawie dziedziny biznesowej	14
Mikrousługi są właścicielem swoich danych	17
Jakie korzyści mogą dawać mikrousługi?	18
Jakie problemy są powodowane przez mikrousługi?	18
Interfejsy użytkownika	19
Technologie	19
Wielkość	20
Właściciele	21
System monolityczny	23
Jednoprocesowe systemy monolityczne	23
Rozproszony system monolityczny	25
Systemy typu czarna skrzynka od niezależnych dostawców	25
Problemy związane z systemami monolitycznymi	25
Zalety systemów monolitycznych	26
O powiązaniu i spójności	26
Spójność	28
Powiązanie	28
Tylko tyle DDD, ile potrzeba	37
Agregat	37
Ograniczony kontekst	38
Odwzorowywanie agregatów i ograniczonych kontekstów na mikrousługi	39
Dalsza lektura	39
Podsumowanie	40

2. Planowanie migracji	41
Zrozumieć cel	41
Trzy kluczowe pytania	42
Dlaczego możesz zdecydować się na mikrouслуги?	43
Zwiększenie autonomii zespołu	43
Skrócenie czasu wprowadzania funkcji na rynek	44
Ekonomiczne skalowanie systemu pod kątem obciążenia	45
Zwiększanie stabilności	46
Skalowanie liczby programistów	47
Wprowadzanie nowej technologii	48
Kiedy wprowadzanie mikrouслуг może być złym pomysłem?	49
Niesprecyzowana dziedzina	50
Startupy	50
Oprogramowanie instalowane przez klienta a oprogramowanie zarządzane	51
Brak dobrego powodu!	52
Wady i zalety	52
Zachęcanie innych do wspólnej podróży	53
Zmienianie organizacji	54
Uświadamianie pilności wprowadzenia zmian	54
Budowanie koalicji kierującej wprowadzaniem zmian	55
Opracowywanie wizji i strategii	56
Komunikowanie wizji zmian	56
Dawanie pracownikom uprawnień do szeroko zakrojonych działań	57
Uzyskiwanie krótkoterminowych sukcesów	58
Konsolidowanie korzyści i wprowadzanie nowych zmian	58
Utrwalanie nowego podejścia w kulturze firmy	58
Znaczenie stopniowej migracji	59
Ważne jest to, co w środowisku produkcyjnym	60
Koszt wprowadzania zmian	60
Odwracalne i nieodwracalne decyzje	60
Lepsze miejsca do eksperymentów	61
Od czego więc zacząć?	62
Podejście DDD	62
Jak szczegółowy powinien być model?	63
Event storming	63
Korzystanie z modelu dziedziny do określania priorytetów	64
Model mieszany	65
Reorganizacja zespołów	67
Zmiany w strukturach	67
Nie istnieją uniwersalne rozwiązania	68

Wprowadzanie zmian	70
Nowe umiejętności	71
Skąd wiadomo, czy zmiany przynoszą dobre efekty?	74
Regularne punkty kontrolne	75
Wskaźniki ilościowe	75
Wskaźniki jakościowe	76
Unikanie efektu utopionych kosztów	76
Otwartość na nowe podejścia	77
Podsumowanie	77
3. Podział systemu monolitycznego	79
Modyfikować system monolityczny czy nie?	79
Wycinać i wklejać czy pisać od nowa?	80
Refaktoryzacja systemu monolitycznego	80
Wzorce migracji	82
Wzorzec „figowiec dusiciel”	82
Jak działa ten wzorzec?	83
Gdzie stosować ten wzorzec?	84
Przykład: odwrotny pośrednik HTTP	86
Dane?	89
Możliwości związane z pośrednikiem	89
Zmiana protokołów	92
Przykład: FTP	95
Przykład: przechwytywanie komunikatów	96
Inne protokoły	99
Inne przykłady stosowania wzorca „figowiec dusiciel”	99
Zmienianie działania kodu przy przenoszeniu funkcji	99
Wzorzec: składanie interfejsu użytkownika	100
Przykład: składanie strony	100
Przykład: składanie widżetów	101
Przykład: mikrofrontendy	104
Gdzie stosować ten wzorzec?	105
Wzorzec: rozgałęzianie z użyciem abstrakcji	105
Jak działa ten wzorzec?	106
Zapewnianie rezerwowego mechanizmu	112
Gdzie używać tego wzorca?	113
Wzorzec: równoległe uruchamianie	113
Przykład: porównywanie cen kredytowych instrumentów pochodnych	114
Przykład: oferty w agencji Homegate	115
Techniki sprawdzania poprawności	116
Używanie szpiegów	116

Biblioteka Scientist z serwisu GitHub	117
Ukryte udostępnianie i udostępnianie próbne	118
Gdzie stosować ten wzorzec?	118
Wzorzec: współdziałający dekorator	118
Przykład: zarządzanie programem lojalnościowym	119
Gdzie stosować ten wzorzec?	120
Wzorzec: przechwytywanie zmian w danych	120
Przykład: wydawanie kart lojalnościowych	120
Implementowanie przechwytywania zmian w danych	121
Gdzie stosować ten wzorzec?	124
Podsumowanie	124
4. Podział baz danych	125
Wzorzec: współdzielona baza danych	125
Wzorce radzenia sobie	126
Gdzie stosować ten wzorzec?	126
Ale to niemożliwe!	127
Wzorzec: widoki bazodanowe	128
Baza danych jako publiczny kontrakt	128
Prezentowane widoki	129
Ograniczenia	131
Własność	131
Gdzie stosować ten wzorzec?	131
Wzorzec: usługa opakowująca bazę danych	131
Gdzie stosować ten wzorzec?	133
Wzorzec: interfejs „baza danych jako usługa”	134
Implementowanie mechanizmu odwzorowywania danych	135
Porównanie z widokami	136
Gdzie stosować ten wzorzec?	136
Przekazywanie własności	136
Wzorzec: system monolityczny udostępniający agregaty	137
Wzorzec: zmiana właściciela danych	139
Synchronizacja danych	140
Wzorzec: synchronizowanie danych z użyciem aplikacji	142
Etap 1. Masowa synchronizacja danych	142
Etap 2. Synchronizowanie zapisu, odczyt z dawnego schematu	143
Etap 3. Synchronizacja zapisu, odczyt z nowego schematu	144
Gdzie stosować ten wzorzec?	144

Wzorzec: stopniowa synchronizacja	145
Synchronizowanie danych	148
Przykład: zamówienia w firmie Square	149
Gdzie stosować ten wzorzec?	153
Podział bazy danych	153
Fizyczny i logiczny podział baz danych	153
Co dzielić najpierw — bazę danych czy kod?	155
Najpierw podział bazy danych	155
Najpierw podział kodu	159
Jednoczesny podział bazy danych i kodu	163
Co więc należy podzielić w pierwszej kolejności?	163
Przykłady dotyczące podziału schematu	164
Wzorzec: tabela pomostowa	164
Wzorzec: przenoszenie relacji klucza obcego do kodu	166
Transakcje	178
Transakcje ACID	178
Nadal ACID, ale bez atomowości?	179
Zatwierdzanie dwuetapowe	181
Transakcje rozproszone — po prostu powiedz „nie”	183
Sagi	184
Rodzaje błędów w sagach	185
Implementowanie sag	188
Sagi a transakcje rozproszone	194
Podsumowanie	195
5. Rosnące problemy	197
Więcej usług, więcej kłopotów	197
Własność przy dużej liczbie usług	198
Jak ten problem może się ujawnić?	199
Kiedy ten problem może wystąpić?	200
Potencjalne rozwiązania	200
Zmiany naruszające zgodność	200
Jak ten problem może się ujawnić?	201
Kiedy ten problem może wystąpić?	201
Potencjalne rozwiązania	201
Generowanie raportów	204
Kiedy ten problem może wystąpić?	205
Potencjalne rozwiązania	205

Monitorowanie i rozwiązywanie problemów	206
Kiedy te problemy mogą wystąpić?	207
Jak objawiają się problemy?	207
Potencjalne rozwiązania	207
Komfort pracy programisty na lokalnej maszynie	211
Jak ten problem może się ujawnić?	211
Kiedy ten problem może wystąpić?	211
Możliwe rozwiązania	212
Uruchamianie zbyt wielu rzeczy	212
Jak ten problem może się ujawnić?	212
Kiedy ten problem może wystąpić?	213
Możliwe rozwiązania	213
Testy end-to-end	214
Jak ten problem może się ujawnić?	214
Kiedy ten problem może wystąpić?	215
Możliwe rozwiązania	215
Optymalizacja globalna i lokalna	217
Jak ten problem może się ujawnić?	217
Kiedy ten problem może wystąpić?	217
Możliwe rozwiązania	218
Stabilność i odporność	219
Jak ten problem może się ujawnić?	219
Kiedy ten problem może wystąpić?	219
Możliwe rozwiązania	220
Osieroczone usługi	220
Jak ten problem może się ujawnić?	221
Kiedy ten problem może wystąpić?	221
Możliwe rozwiązania	221
Podsumowanie	222
6. Słowo na zakończenie	225
A Bibliografia	227
B Indeks wzorców	229