

Przedmowa	13
 1. Dlaczego mikrousługi oparte na zdarzeniach?	 17
Czym są mikrousługi oparte na zdarzeniach?	17
Wprowadzenie do projektowania dziedzinowego i kontekstów ograniczonych	19
Wykorzystywanie modeli dziedzin i kontekstów ograniczonych	20
Dopasowywanie kontekstów ograniczonych do wymagań biznesowych	20
Struktury komunikacji	22
Struktury komunikacji biznesowej	22
Struktury komunikacji implementacyjnej	22
Struktury komunikacji danych	23
Prawo Conwaya i struktury komunikacji	23
Struktury komunikacji w tradycyjnych systemach obliczeniowych	25
Opcja 1. Utworzenie nowej usługi	25
Opcja 2. Dodanie funkcjonalności do istniejącej usługi	26
Wady i zalety obu opcji	26
Kontynuacja scenariusza z zespołami	27
Sprzeczne naciski	28
Struktury komunikacji oparte na zdarzeniach	28
Zdarzenia są podstawą komunikacji	28
Strumienie zdarzeń zapewniają jedno źródło prawdy	28
Konsumenty przeprowadzają własne modelowanie i kwerendowanie	29
Komunikacja danych zostaje usprawniona w całej organizacji	29
Dostępne dane wspierają zmiany w komunikacji biznesowej	30
Asynchroniczne mikrousługi oparte na zdarzeniach	30
Przykładowy zespół korzystający z mikrousług opartych na zdarzeniach	31
Mikrousługi synchroniczne	31
Wady mikrousług synchronicznych	32
Korzyści ze stosowania mikrousług synchronicznych	33
Podsumowanie	34

2. Podstawy mikrousług opartych na zdarzeniach	35
Budowanie topologii	35
Topologia mikrousługi	35
Topologia biznesowa	36
Zawartość zdarzenia	37
Struktura zdarzenia	37
Zdarzenie bez klucza	38
Zdarzenie encyjne	38
Zdarzenie z kluczem	38
Materializowanie stanu ze zdarzeń encyjnych	39
Definicje i schematy danych zdarzenia	41
Zasada pojedynczego źródła zapisu mikrousługi	41
Mikrousługi wyposażone w broker zdarzeń	41
Przechowywanie i serwowanie zdarzeń	42
Uwzględnianie czynników dodatkowych	43
Porównanie brokerów zdarzeń i brokerów komunikatów	44
Konsumowanie zdarzeń z niemutowalnego dziennika	45
Zapewnianie jednego źródła prawdy	47
Zarządzanie mikrousługami na dużą skalę	47
Umieszczanie mikrousług w kontenerach	48
Umieszczanie mikrousług w maszynach wirtualnych	48
Zarządzanie kontenerami i maszynami wirtualnymi	48
Podatek od mikrousług	49
Podsumowanie	50
 3. Komunikacja i kontrakty danych	 51
Kontrakty danych opartych na zdarzeniach	51
Używanie bezpośrednich schematów jako kontraktów	52
Komentarze definicji schematu	52
W pełni funkcjonalna ewolucja schematu	53
Obsługa generatora kodu	54
Przełomowe zmiany schematu	55
Wybór formatu zdarzenia	57
Projektowanie zdarzeń	57
Mów prawdę, całą prawdę i tylko prawdę	58
Używaj pojedynczej definicji zdarzenia na strumień	58
Używaj najwęższych typów danych	58
Zachowuj specjalizację zdarzeń	59
Minimalizuj rozmiar zdarzeń	62
Angażuj potencjalne konsumenty w projektowanie zdarzeń	62
Unikaj zdarzeń jako semaforów lub sygnałów	62
Podsumowanie	63

4. Integracja architektur opartych na zdarzeniach z istniejącymi systemami	65
Czym jest uwalnianie danych?	66
Kompromisy związane z uwalnianiem danych	66
Konwersja uwolnionych danych na zdarzenia	68
Wzorce uwalniania danych	69
Frameworki uwalniania danych	70
Uwalnianie danych oparte na zapytaniach	70
Ładowanie zbiorcze	70
Ładowanie przyrostowe na podstawie znacznika czasu	71
Ładowanie na podstawie autoinkrementowanego identyfikatora	71
Kwerendowanie niestandardowe	71
Aktualizacja przyrostowa	71
Zalety aktualizacji opartej na zapytaniach	72
Wady aktualizacji opartej na zapytaniach	72
Uwalnianie danych oparte na dziennikach CDC	73
Zalety korzystania z dzienników magazynu danych	75
Wady korzystania z dzienników magazynu danych	75
Uwalnianie danych oparte na tablicach skrzynki nadawczej	75
Kwestie związane z wydajnością	77
Izolowanie wewnętrznych modeli danych	77
Zapewnianie kompatybilności schematu	79
Przechwytywanie danych o zmianach za pomocą wyzwalaczy	82
Dokonywanie zmian definicji danych w przechwytywanych zbiorach danych	85
Obsługa zmian definicji danych „po fakcie”	
dla wzorców opartych na zapytaniach i dziennikach CDC	86
Obsługa zmian definicji danych dla wzorców opartych	
na przechwytywaniu tablicy danych o zmianach	86
Zlewanie danych o zdarzeniach do magazynów danych	87
Wpływ zlewania i pozyskiwania danych na prowadzenie działalności biznesowej	87
Podsumowanie	89
5. Podstawy przetwarzania opartego na zdarzeniach	91
Tworzenie topologii bezstanowych	92
Transformacje	92
Rozgałęzianie i scalanie strumieni	93
Repartycjonowanie strumieni zdarzeń	93
Przykład: repartycjonowanie strumienia zdarzeń	94
Współpartycjonowanie strumieni zdarzeń	95
Przykład: współpartycjonowanie strumienia zdarzeń	95

Przypisywanie partycji do instancji konsumenta	96
Mechanizm przypisywania partycji	96
Przypisywanie partycji współpartycjonowanych	96
Strategie przypisywania partycji	97
Odzyskiwanie sprawności po awariach bezstanowej instancji przetwarzania	99
Podsumowanie	99
6. Deterministyczne przetwarzanie strumieni	101
Determinizm w przepływach pracy opartych na zdarzeniach	102
Znaczniki czasu	102
Synchronizacja rozproszonych znaczników czasu	103
Przetwarzanie zdarzeń ze znacznikami czasu	104
Planowanie zdarzeń i przetwarzanie deterministyczne	105
Niestandardowe dyspozytory zdarzeń	106
Przetwarzanie na podstawie czasów: zdarzenia, przetwarzania i pozyskania	106
Wyodrębnianie znacznika czasu przez konsumenta	106
Wywołania typu żądanie-odpowiedź wysyłane do systemów zewnętrznych	107
Znaki wodne	107
Znaki wodne w przetwarzaniu równoległym	108
Czas strumienia	109
Czas strumienia w przetwarzaniu równoległym	110
Zdarzenia w niewłaściwej kolejności i zdarzenia opóźnione	111
Zdarzenia opóźnione ze znakami wodnymi i czasem strumienia	112
Przyczyny i skutki występowania zdarzeń w niewłaściwej kolejności	113
Funkcje zależne od czasu i okna czasowe	114
Obsługa zdarzeń opóźnionych	117
Porównanie ponownego przetwarzania i przetwarzania w czasie zbliżonym do rzeczywistego	118
Okresowe awarie i opóźnione zdarzenia	119
Problemy z łącznością producenta (brokera zdarzeń)	119
Podsumowanie i dalsza lektura	121
7. Stanowe przesyłanie strumieniowe	123
Magazyny stanu i materializowanie stanu ze strumienia zdarzeń	123
Rejestrowanie stanu w strumieniu zdarzeń dziennika zmian	124
Materializacja stanu w wewnętrznym magazynie stanu	125
Materializacja stanu globalnego	126
Zalety korzystania z wewnętrznego stanu	126
Wady korzystania z wewnętrznego stanu	128
Skalowanie i przywracanie wewnętrznego stanu	128

Materializacja stanu w zewnętrznym magazynie stanu	131
Zalety zewnętrznego stanu	132
Wady zewnętrznego stanu	132
Skalowanie i odzyskiwanie w przypadku zewnętrznych magazynów stanu	133
Porównanie odbudowywania i migrowania magazynów stanu	135
Odbudowa	135
Migracja	136
Transakcje i przetwarzanie efektywnie raz	136
Przykład: usługa rachunkowości zapasów	137
Przetwarzanie efektywnie raz za pomocą transakcji klient-broker	137
Przetwarzanie efektywnie raz bez transakcji klient-broker	139
Podsumowanie	144
8. Tworzenie przepływów pracy za pomocą mikrouslug	145
Wzorzec Choreografia	146
Prosty przykład choreografii opartej na zdarzeniach	147
Tworzenie i modyfikowanie choreografowanego przepływu pracy	148
Monitorowanie choreografowanego przepływu pracy	148
Wzorzec Orkiestracja	149
Prosty przykład orkiestracji opartej na zdarzeniach	149
Prosty przykład orkiestracji wywołań bezpośrednich	151
Porównanie orkiestracji bezpośrednich wywołań i orkiestracji opartej na zdarzeniach	151
Tworzenie i modyfikowanie przepływu pracy orkiestracji	152
Monitorowanie przepływu pracy orkiestracji	153
Transakcje rozproszone	153
Transakcje choreografowane: wzorzec Saga	153
Transakcje orkiestrowane	155
Kompensacyjne przepływy pracy	157
Podsumowanie	158
9. Mikrouslugi wykorzystujące funkcję jako usługę	159
Projektowanie rozwiązań opartych na funkcjach jako mikrouslug	159
Zapewnianie ścisłej przynależności do kontekstu ograniczonego	159
Zatwierdzanie przesunąć dopiero po zakończeniu przetwarzania	160
Mniej znaczy więcej	161
Wybór dostawcy FaaS	161
Budowanie mikrouslug na podstawie funkcji	161
Zimny start i ciepłe starty	163

Uruchamianie funkcji za pomocą wyzwalaczy	163
Wyzwalanie oparte na nowych zdarzeniach: nasłuchiwanie strumienia zdarzeń	163
Wyzwalanie oparte na opóźnieniu grupy konsumentów	165
Wyzwalanie według harmonogramu	166
Wyzwalanie przy użyciu webhooksów	166
Wyzwalanie oparte na zdarzeniach dotyczących zasobów	167
Wykonywanie pracy biznesowej za pomocą funkcji	167
Utrzymywanie stanu	167
Funkcje wywołujące inne funkcje	168
Wzorzec komunikacji opartej na zdarzeniach	168
Wzorzec bezpośrednich wywołań	169
Zakończenie działania i zamknięcie	172
Dostrajanie funkcji	172
Alokacja wystarczających zasobów	172
Parametry wsadowego przetwarzania zdarzeń	173
Skalowanie rozwiązań FaaS	173
Podsumowanie	174
10. Mikrousługi BPC	175
Gdzie sprawdzają się mikrousługi BPC?	175
Integracja z istniejącymi i starszymi systemami	176
Stanowa logika biznesowa, która nie jest zależna od kolejności zdarzeń	177
Gdy warstwa danych wykonuje większość pracy	178
Niezależne skalowanie przetwarzania i warstwy danych	178
Hybrydowe aplikacje BPC z zewnętrznym przetwarzaniem strumieni	179
Przykład: użycie zewnętrznego frameworku przetwarzania strumieni do łączenia strumieni zdarzeń	180
Podsumowanie	181
11. Mikrousługi frameworków ciężkich	183
Krótka historia ciężkich frameworków	184
Wewnętrzne działanie ciężkich frameworków	185
Korzyści i ograniczenia	186
Opcje konfiguracji klastra i tryby wykonywania	188
Użycie usługi hostowanej	188
Budowanie własnego pełnego klastra	188
Tworzenie klastrów z integracją z systemem CMS	189
Tryby zatwierdzania aplikacji	190
Tryb sterownika	190
Tryb klastra	191
Obsługa stanu i używanie punktów kontrolnych	191

Skalowanie aplikacji i obsługa partycji strumienia zdarzeń	192
Skalowanie aplikacji podczas jej działania	193
Skalowanie aplikacji przez jej ponowne uruchomienie	196
Automatycznie skalujące się aplikacje	196
Odzyskiwanie sprawności po awarii	196
Kwestie wielodzierżawności	197
Języki i składnia	197
Wybór frameworku	198
Przykład: tworzenie okna sesji kliknięć i wyświetleń	198
Podsumowanie	201
12. Mikrouslugi frameworków lekkich	203
Zalety i ograniczenia	203
Lekkie przetwarzanie	204
Obsługa stanu i używanie dzienników zmian	204
Skalowanie aplikacji i odzyskiwanie sprawności po awarii	205
Tasowanie zdarzeń	205
Przypisywanie stanu	206
Replikacja stanu i aktywne repliki	206
Wybór lekkiego frameworku	207
Apache Kafka Streams	207
Apache Samza: tryb osadzony	207
Języki i składnia	208
Łączenie strumień-tablica-tablica: wzorzec Wzbogacanie	208
Podsumowanie	212
13. Integracja mikrouslug opartych na zdarzeniach z mikrouslugami typu żądanie-odpowiedź	213
Obsługa zdarzeń zewnętrznych	213
Zdarzenia generowane autonomicznie	214
Zdarzenia generowane reaktywnie	214
Obsługa automatycznie generowanych zdarzeń analitycznych	215
Integracja z zewnętrznymi interfejsami API żądanie-odpowiedź	216
Przetwarzanie i udostępnianie danych stanowych	218
Obsługa żądań w czasie rzeczywistym za pomocą wewnętrznych magazynów stanu	218
Obsługa żądań w czasie rzeczywistym za pomocą zewnętrznych magazynów stanu	221
Obsługa żądań w przepływie pracy opartym na zdarzeniach	224
Przetwarzanie zdarzeń dla interfejsów użytkownika	225
Mikrofrontendy w aplikacjach typu żądanie-odpowiedź	231
Zalety mikrofrontendów	233
Mikrouslugi oparte na kompozycji	233
Łatwe dostosowywanie do wymagań biznesowych	233

Wady mikrofrontendów	233
Potencjalnie niespójne elementy i style interfejsu użytkownika	234
Zmienna wydajność mikrofrontendu	234
Przykład: aplikacja do wyszukiwania i recenzowania wydarzeń	234
Podsumowanie	237
14. Narzędzia pomocnicze	239
System przypisywania mikrousług zespołom	239
Tworzenie i modyfikowanie strumienia zdarzeń	240
Znakowanie strumieni zdarzeń za pomocą metadanych	240
Kwoty	241
Rejestr schematów	241
Powiadomienia o tworzeniu i modyfikowaniu schematów	243
Zarządzanie przesunięciami	243
Uprawnienia i listy kontroli dostępu dla strumieni zdarzeń	244
Zarządzanie stanem i resetowanie aplikacji	245
Monitorowanie opóźnienia przesunięcia konsumenta	246
Zoptymalizowany proces tworzenia mikrousług	246
Kontrola zarządzania kontenerami	247
Tworzenie klastra i zarządzanie nim	247
Programowe uruchamianie brokerów zdarzeń	248
Programowe uruchamianie zasobów obliczeniowych	248
Replikacja danych zdarzeń między klastrami	249
Programowe uruchamianie narzędzi	249
Śledzenie zależności i wizualizacja topologii	249
Przykład topologii	251
Podsumowanie	253
15. Testowanie mikrousług opartych na zdarzeniach	255
Ogólne zasady testowania	255
Funkcje topologii testów jednostkowych	256
Funkcje bezstanowe	256
Funkcje stanowe	256
Testowanie topologii	257
Testowanie ewolucji i zgodności schematów	258
Testowanie integracyjne mikrousług opartych na zdarzeniach	258
Lokalne testy integracyjne	259
Tworzenie tymczasowego środowiska w ramach wykonywania kodu testowego	261
Tworzenie tymczasowego środowiska zewnętrznie względem kodu testowego	262
Integracja usług hostowanych przy użyciu opcji atrapy i symulatora	263
Integracja usług zdalnych, które nie mają opcji lokalnych	263

Pełne zdalne testy integracyjne	264
Programowe tworzenie tymczasowego środowiska testów integracyjnych	265
Testowanie przy użyciu środowiska współdzielonego	267
Testowanie przy użyciu środowiska produkcyjnego	268
Wybór strategii w pełni zdalnych testów integracyjnych	269
Podsumowanie	270
16. Wdrażanie mikrousług opartych na zdarzeniach	271
Zasady wdrażania mikrousług	271
Architektoniczne komponenty wdrażania mikrousług	272
Systemy ciągłej integracji, ciągłego dostarczania i ciągłego wdrażania	272
Systemy zarządzania kontenerami i powszechnie dostępny sprzęt	273
Podstawowy wzorzec Wdrożenie z Pełnym Zatrzymaniem	274
Wzorzec Aktualizacja Krocząca	275
Wzorzec Przełomowa Zmiana Schematu	276
Ostateczna migracja za pośrednictwem dwóch strumieni zdarzeń	277
Zsynchronizowana migracja do nowego strumienia zdarzeń	278
Wzorzec Wdrożenie Niebiesko-Zielone	279
Podsumowanie	280
17. Zakończenie	281
Warstwy komunikacyjne	281
Dziedziny biznesowe i konteksty ograniczone	282
Współużytkowanie narzędzi i infrastruktury	282
Uschematyzowane zdarzenia	282
Wyzwolenie danych i jedno źródło prawdy	283
Mikrousługi	284
Opcje implementacji mikrousług	284
Testowanie	285
Wdrażanie	285
Kilka słów na zakończenie	286