

Przedmowa	13
Wstęp	15
Wprowadzenie	21

CZĘŚĆ I. Opracowywanie, budowanie i testowanie API 33

1. Opracowywanie, budowanie i określanie API	35
Przykład opracowywania API uczestnika	35
Wprowadzenie do stylu REST	36
Oparte na przykładzie wprowadzenie do technologii REST i HTTP	36
Model dojrzałości Richardsona	37
Wprowadzenie do API zdalnego wywoływania procedur	38
Krótkie wprowadzenie do GraphQL	39
Struktura i standardy API REST	40
Kolekcje i stronicowanie	41
Filtrowanie kolekcji	42
Obsługa błędów	42
Wskazówki dotyczące dokumentu typu ADR — wybór standardu API	43
Określanie API REST za pomocą OpenAPI	44
Praktyczne zastosowanie specyfikacji OpenAPI	44
Generowanie kodu	45
Weryfikacja OpenAPI	45
Przykłady i imitacje	46
Wykrywanie zmian	46
Wersjonowanie API	47
Wersjonowanie semantyczne	48
Specyfikacja OpenAPI i wersjonowanie	48
Implementacja RPC za pomocą gRPC	49

Modelowanie zmian i wybór formatu API	51
Usługi z wysokim poziomem ruchu sieciowego	52
Ogromne ilości wymienianych danych	52
Korzyści związane z wydajnością działania HTTP/2	52
Stare formaty	53
Wskazówki dotyczące dokumentu typu ADR — modelowanie wymiany danych	53
Wiele specyfikacji	54
Czy istnieje złoty środek?	54
Wyzwania związane z połączeniem specyfikacji	55
Podsumowanie	56
2. Testowanie API	57
Użyty w tym rozdziale scenariusz systemu konferencyjnego	58
Strategie testowania	58
Kwadrant testów	59
Piramida testów	61
Wskazówki dotyczące dokumentu typu ADR — strategie testowania	63
Testowanie kontraktu	63
Dlaczego często preferowane jest testowanie kontraktu?	64
Jak jest implementowany kontrakt?	64
Wskazówki dotyczące dokumentu typu ADR — testowanie kontraktu	69
Testowanie komponentu API	70
Testowanie kontraktu kontra testowanie komponentu	71
Przykład użycia testowania komponentu do weryfikacji sposobu działania	71
Testowanie integracji API	72
Używanie szkieletów serwerów — kiedy i dlaczego?	73
Wskazówki dotyczące dokumentu typu ADR — testy integracji	74
Konteneryzowanie komponentów testowych — biblioteka Testcontainers	75
Przykład zastosowania biblioteki Testcontainers do weryfikacji integracji	75
Testy typu E2E	77
Automatyzacja weryfikacji E2E	77
Typy testów E2E	78
Wskazówki dotyczące dokumentu typu ADR — testowanie typu E2E	79
Podsumowanie	80

CZĘŚĆ II. API zarządzania ruchem sieciowym **81**

3. Bramy API — zarządzanie przychodzącym ruchem sieciowym	83
Czy brama API to jedyne rozwiązanie?	83
Wskazówki dotyczące dokumentu typu ADR — proxy, mechanizm równoważenia obciążenia lub brama API	84
Przykład udostępnienia konsumentom usługi uczestnika	85

Czym jest brama API?	85
Jaką funkcjonalność może zaoferować brama API?	86
Gdzie zostaje wdrożona brama API?	87
Jak brama API integruje się z innymi technologiami w położeniu brzegowym?	88
Dlaczego warto używać bramy API?	88
Zmniejszenie poziomu powiązania przez użycie wzorca adaptera/fasady między frontendem a backendem	90
Uproszczenie sposobu użycia przez agregację i tłumaczenie usług backendu	90
Ochrona API przed nadużyciami dzięki wykorzystaniu mechanizmu wykrywania zagrożeń i ich łagodzeniu	92
Zrozumienie, w jaki sposób może być używane API (monitorowanie)	93
Zarządzanie API jako produktem poprzez zarządzanie cyklem życiowym API	93
Zarabianie na API dzięki użyciu mechanizmów zarządzania kontem, rozliczeniami i płatnościami	95
Nowoczesna historia bram API	95
Od lat dziewięćdziesiątych ubiegłego wieku — sprzętowe mechanizmy równoważenia obciążenia	96
Od lat dwutysięcznych — programowe mechanizmy równoważenia obciążenia	96
Pierwsza dekada XXI wieku — kontrolery dostarczania aplikacji	97
Druga dekada XXI wieku — pierwsza generacja bram API	98
Rok 2015 — druga generacja bram API	99
Taksonomia obecnych bram API	100
Tradycyjne bramy korporacyjne	101
Mikrousługi i mikrobramy	101
Bramy infrastruktury typu service mesh	101
Porównanie typów bram API	102
Przykład ewolucji systemu konferencyjnego z użyciem bramy API	103
Instalacja stosu Ambassador Edge Stack w Kubernetes	104
Konfigurowanie mapowania ze ścieżek dostępu adresów URL do usług backendu	105
Konfiguracja mapowania z użyciem routingu bazującego na hoście	106
Wdrażanie bramy API — poznanie niepowodzeń i radzenie sobie z nimi	106
Brama API jako pojedynczy punkt awarii	107
Wykrywanie i usuwanie problemów	107
Rozwiązywanie problemów i radzenie sobie z incydentami	108
Łagodzenie ryzyka	108
Najczęściej pojawiające się problemy podczas implementacji bramy API	109
Pętla zwrotna bramy API	109
Brama API jako korporacyjna magistrała usług	110
Bramy API aż do końca	110

Wybór bramy API	110
Określenie wymagań	110
Samodzielne opracowanie rozwiązania kontra zakup gotowego	111
Wskazówki dotyczące dokumentu typu ADR — wybór bramy API	111
Podsumowanie	112

4. Infrastruktura typu service mesh i zarządzanie ruchem sieciowym między usługami	115
Czy infrastruktura typu service mesh to jedyne rozwiązanie?	115
Wskazówki dotyczące dokumentu typu ADR — czy należy zastosować infrastrukturę typu service mesh?	116
Przykład wyodrębnienia funkcjonalności sesji do nowej usługi	117
Czym jest infrastruktura typu service mesh?	118
Jaką funkcjonalność dostarcza infrastruktura typu service mesh?	120
Gdzie jest wdrażana infrastruktura typu service mesh?	121
Jak infrastruktura typu service mesh integruje się z innymi topologiami sieci?	121
Dlaczego warto używać infrastruktury typu service mesh?	123
Pełna kontrola nad routowaniem usługi, niezawodnością i zarządzaniem ruchem sieciowym	124
Niewidoczne monitorowanie	127
Wymuszenie bezpieczeństwa, np. szyfrowanie transmisji, uwierzytelnianie i autoryzacja	128
Obsługa komunikacji międzyfunkcyjnej w różnych językach programowania	128
Oddzielenie przychodzącego ruchu sieciowego od zarządzania ruchem sieciowym między usługami	129
Ewolucja infrastruktury typu service mesh	130
Wczesna historia i motywy	131
Wzorce implementacji	132
Taksonomia infrastruktury typu service mesh	139
Przykład użycia infrastruktury typu service mesh na potrzeby związane z routowaniem, monitorowaniem i zapewnieniem bezpieczeństwa	139
Routing za pomocą Istio	140
Monitorowanie ruchu sieciowego za pomocą Linkerd	141
Segmentacja sieci za pomocą narzędzia Consul	143
Wdrażanie infrastruktury typu service mesh — zrozumienie awarii i zarządzanie nimi	146
Infrastruktura typu service mesh jako pojedynczy punkt awarii	146
Najczęściej pojawiające się trudności podczas implementacji infrastruktury typu service mesh	146
Infrastruktura typu service mesh jako korporacyjna magistrala usług	146
Infrastruktura typu service mesh jako brama	147
Zbyt wiele warstw sieciowych	147

Wybór infrastruktury typu service mesh	147
Określenie wymagań	147
Samodzielne opracowanie rozwiązania kontra zakup gotowego	148
Lista rzeczy do sprawdzenia podczas wyboru infrastruktury typu service mesh	149
Podsumowanie	149

Część III. Funkcjonowanie API i jego zabezpieczanie 151

5. Wdrażanie i wydawanie API	153
Rozdzielenie wdrożenia i wydania	154
Przykład włączania funkcjonalności	154
Zarządzanie ruchem sieciowym	156
Przykład modelowania wydań w systemie konferencyjnym	157
Cykl życiowy API	157
Mapowanie strategii wydań na cykl życiowy	158
Wskazówki dotyczące dokumentu typu ADR — rozdzielenie operacji wdrożenia i wydania dzięki użyciu zarządzania ruchem sieciowym i techniki włączania funkcjonalności	159
Strategie wydań	159
Wydania kanarkowe	159
Odbicie lustrzane ruchu sieciowego	161
Niebieski-zielony	162
Przykład przeprowadzania wdrożenia za pomocą narzędzia Argo Rollouts	164
Monitorowanie pod kątem sukcesu i identyfikowanie niepowodzeń	167
Trzy filary monitorowania	167
Ważne wskaźniki dla API	168
Odczytywanie sygnałów	169
Decyzje związane z efektywnymi wydaniem oprogramowania	170
Buforowanie odpowiedzi	170
Propagowanie nagłówka na poziomie aplikacji	171
Rejestrowanie danych, aby ułatwić debugowanie	171
Rozważenie użycia sprawdzonej platformy	171
Wskazówki dotyczące dokumentu typu ADR — sprawdzone platformy	172
Podsumowanie	172
6. Bezpieczeństwo operacyjne — model zagrożeń dla API	175
Przykład zastosowania metody OWASP w API uczestnika	176
Ryzyko związane z niezabezpieczeniem zewnętrznego API	177
Modelowanie zagrożeń	178
Myśl jak atakujący	179

Jak odbywa się modelowanie zagrożeń?	180
Krok 1. — określ cele	180
Krok 2. — zbierz właściwe informacje	181
Krok 3. — rozłóż system na czynniki	181
Krok 4. — określ zagrożenia	182
Krok 5. — oceń ryzyko związane z zagrożeniami	193
Krok 6. — przeprowadź weryfikację	195
Podsumowanie	195
7. Uwierzelnianie i autoryzacja API	197
Uwierzelnianie	197
Uwierzelnianie użytkownika końcowego z wykorzystaniem tokenów	198
Uwierzelnianie między systemami	199
Dlaczego nie należy łączyć kluczy i użytkowników?	200
OAuth2	201
Rola serwera autoryzacji i interakcje API	201
JSON Web Token (JWT)	202
Terminologia i mechanizmy grantów OAuth2	204
Wskazówki dotyczące dokumentu typu ADR — czy należy rozważyć użycie OAuth2?	205
Grant kodu autoryzacji	206
Token odświeżania	210
Grant danych uwierzelniających klienta	210
Dodatkowe granty OAuth2	211
Wskazówki dotyczące dokumentu typu ADR — wybór używanego grantu OAuth2	212
Zasięg OAuth2	212
Wymuszenie autoryzacji	214
Wprowadzenie do OIDC	215
SAML 2.0	216
Podsumowanie	217

CZĘŚĆ IV. Architektura ewolucyjna z użyciem API **219**

Rozdział 8. Przeprojektowanie aplikacji do architektury bazującej na API	221
Dlaczego używać API do ewolucji systemu?	221
Tworzenie użytecznych abstrakcji — większa spójność	222
Definiowanie granic domeny — promowanie luźnego powiązania	223
Przykład pokazujący określenie granic domeny uczestnika	224
Opcje architekuralne stanu końcowego	224
Monolit	225
Architektura zorientowana na usługi	225
Mikrousługi	226
Funkcje	226

Zarządzanie procesem ewolucyjnym	227
Określenie celu	227
Używanie funkcji przystosowania	227
Podział systemu na moduły	229
Utworzenie API jako „szwów” dla rozszerzenia	231
Określanie zmiany punktów wzmocnienia w systemie	231
Ciągłe wdrażanie i weryfikacja	232
Wzorce architektoniczne dla wykorzystujących API systemów ewoluujących	232
Wzorzec „strangler fig”	232
Wzorce fasady i adaptera	234
Tort API	234
Określanie potencjalnych możliwości i trudnych miejsc	235
Problemy związane z uaktualnieniami i obsługą techniczną	235
Problemy związane z wydajnością działania	236
Przerwanie zależności — wysoce powiązane API	236
Podsumowanie	237
9. Używanie infrastruktury API do ewolucji w kierunku platform chmury	239
Przykład przeniesienia usługi uczestnika do chmury	239
Wybór strategii migracji do chmury	240
Retain lub Revisit	241
Rehost	242
Replatform	242
Repurchase	243
Refactor/Rearchitect	243
Retire	243
Przykład zmiany platformy dla usługi uczestnika i przeniesienie jej do chmury	244
Rola zarządzania API	244
Ruch sieciowy typu północ – południe kontra ruch sieciowy	
typu wschód – zachód — zanikanie granic zarządzania ruchem sieciowym	246
Rozpoczęcie od położenia brzegowego, a następnie przejście do wnętrza sieci	246
Przekraczanie granic — routing między sieciami	246
Od architektury bazującej na strefach do sieci o zerowym zaufaniu	247
Architektura bazująca na strefach	247
Nie ufaj nikomu i sprawdzaj	249
Rola infrastruktury typu service mesh w architekturze	
bazującej na zerowym zaufaniu	250
Podsumowanie	253
10. Podsumowanie	255
Spojrzenie wstecz na naszą podróż	255
API, prawo Conwaya i Twoja organizacja	261
Poznajemy rodzaje decyzji	262

Wybieganie w przyszłość	262
Komunikacja asynchroniczna	262
HTTP/3	263
Platforma typu service mesh	263
Co dalej? Jak nadal poznawać architekturę API?	263
Nieustanne doskonalenie podstaw	264
Śledzenie nowości w branży	264
Radary, kwadranty i raporty dotyczące trendów	264
Poznawanie najnowszych praktyk i przykładów	265
Uczenie się przez działanie	266
Uczenie się przez nauczanie	266