

Inhalt

Geleitwort des Herausgebers	XVII
Vorwort	XIX
1 Serviceorientierung	1
1.1 Konzeptionelle Ebene	1
1.1.1 Betriebswirtschaftliche Sicht	1
1.1.2 Technische Sicht	2
1.1.3 Was ist ein Service?	3
1.2 Technische Realisierung	4
1.2.1 SOAP	4
1.2.2 Web Service Description Language (WSDL)	6
1.2.3 Universal Description, Discovery and Integration	7
1.2.4 WS-I	7
1.2.5 WS-*	8
1.2.6 RESTfui Web Services als Gegenbewegung zu SOAP	8
1.2.7 POX-Services und Web APIs	11
1.2.8 SOAP- und REST im Vergleich	12
1.3 WCF vs. ASP.NET Web API	13
2 WCF im Überblick	15
2.1 Architektur	15
2.2 Standard-Bindings	16
2.3 Hosting von Services	18
2.4 Erste Schritte mit WCF	19
2.4.1 Erstellen eines Web-Service-Projektes	19
2.4.2 Web-Service mit Client konsumieren	27
2.4.3 Mit Laufzeit-Proxy auf Service zugreifen	31
2.4.4 Service zur Verwendung von ws2007HttpBinding konfigurieren	32
2.4.5 NetTcpBinding und Self-Hosting	33

3	Services mit WCF erstellen	39
3.1	Verträge	39
3.1.1	Serviceverträge	39
3.1.2	Datenverträge	40
3.1.3	Nachrichtenverträge	43
3.1.4	SOAP-Binding festlegen	44
3.2	Instanziierung von Services	44
3.2.1	PerCall	45
3.2.2	Single	46
3.2.3	PerSession	46
3.3	Nebenläufigkeit	51
3.4	Asynchrone Service-Operationen	51
3.5	WCF konfigurieren	52
3.5.1	WCF deklarativ konfigurieren	52
3.5.2	WCF programmatisch konfigurieren	55
3.5.3	In IIS gehostete Services programmatisch konfigurieren (ab .NET 4.5)	55
3.5.4	Benutzerdefinierte Bindings	57
3.5.5	Einschränkungen für Bindings festlegen	58
3.5.6	Drosselung	59
3.5.7	Port-Sharing bei TCP-basierten Services	60
3.5.8	Konfiguration des Proxy-Servers	61
3.6	Metadaten	62
3.7	Services diagnostizieren	64
3.7.1	Protokollierung konfigurieren	64
3.7.2	Leistungsindikatoren	66
3.8	One-Way-Operationen	67
3.9	Duplex-Operationen	67
3.9.1	Unterstützte Bindings	68
3.9.2	Implementierung von Duplex-Szenarien	69
3.9.3	Konfigurieren von Duplex-Szenarien	70
3.9.4	Callbacks mit WebSockets (ab .NET 4.5)	72
3.9.5	Grenzen von Callbacks	73
3.10	UDP und Multicasts (ab .NET 4.5)	74
3.11	Umgang mit binären Daten	75
3.11.1	MTOM	76
3.11.2	Streaming	80
3.12	Fehlerbehandlung und FaultContracts	83
3.13	ASP.NET-Kompatibilität	85
3.14	Zuverlässige Sitzungen	87
3.14.1	Verlässliche Sitzungen konfigurieren	88
3.14.2	Verlässliche Sitzungen erzwingen	89
3.14.3	Idempotente Services als Alternative zu WS-ReliableMessaging	89
3.15	Transaktionale Services	90
3.15.1	Zwei-Phasen-Commit	90
3.15.2	Plug-in für WS-AtomicTransactions einrichten	90

3.15.3	Distributed Transaction Coordinator (DTC) einrichten	91
3.15.4	Transaktionen konfigurieren und nutzen	92
3.15.5	Transaktionsisolationslevel	94
3.15.6	Transaktion am Client starten	95
3.15.7	Transaktionen und Sessions	95
3.16	Queued Services	96
3.16.1	Microsoft Message Queuing Services (MSMQ)	96
3.16.2	Implementierung von Queued Services	97
3.16.3	Queued Services konfigurieren	98
3.17	REST-basierte Services mit WCF	100
3.17.1	REST-Services mit WCF implementieren und konsumieren	101
3.17.2	Antwortformat dynamisch festlegen	104
3.17.3	Hilfe-Seiten (Help Pages)	105
3.17.4	ASP.NET Cache Profiles	107
3.17.5	REST-Services über ASP.NET-Routen	108
3.17.6	Ausnahmen auf HTTP-Statuscodes abbilden	108
3.17.7	Conditional GET und ETag-Unterstützung	109
3.18	WCF und Windows 8	110
4	Sicherheit von WCF-Diensten	111
4.1	Überblick über WCF-Security	111
4.1.1	Transport- und Nachrichtensicherheit	111
4.1.2	Festlegen der zu verwendenden Credentials	113
4.1.3	Authentifizierung und Autorisierung	114
4.1.4	Service-Identitäten	116
4.1.5	Verschlüsseln und Signieren	116
4.2	Windows-Security	118
4.2.1	Impersonation	118
4.2.2	Kerberos vs. NTLM	119
4.3	Web-Security mit SSL und IIS	121
4.3.1	IIS für die Verwendung von SSL konfigurieren	121
4.3.2	Konfiguration des Service	125
4.3.3	Aufruf der Service-Operation	126
4.3.4	Benutzerdefinierte Authentifizierung und Autorisierung	126
4.4	Web-Security mit SSL ohne IIS	133
4.5	Nachrichtenbasierte Sicherheit	134
4.5.1	Zertifikate einrichten	134
4.5.2	Konfiguration	135
4.5.3	Aufruf der Service-Operation	137
4.6	Client-Zertifikate	137
4.7	Federated und Claims-based Security mit Windows Identity Foundation (WIF)	139
4.7.1	Architektur	140
4.7.2	Web-Service-Standards	141
4.7.3	Implementierung in .NET 4.5	141
4.7.4	Implementierung für Versionen vor 4.5	155

5	Lose Kopplung mit WCF	167
5.1	Routing (System.ServiceModel.Routing)	167
5.1.1	Architektur	168
5.1.2	Routerarten	169
5.1.3	Beispielanwendung (Routing Testclient)	170
5.1.4	Filter	171
5.1.5	Erstellen des Routers mit WCF 4.x	174
5.1.6	Konfiguration eines Routers	174
5.1.7	Router für das Routing-Testszenario	175
5.1.8	Entwicklung des Clients	180
5.1.9	Dynamische Filter entwickeln	181
5.1.10	Dynamische Filter mit eigener Filbertabelle	183
5.1.11	Leistungsverlust durch Routing	187
5.2	Discovery	188
5.2.1	Dienst mit Discovery	188
5.2.2	Client mit Discovery	189
5.2.3	Mögliche Services mittels Scopes einschränken	190
5.2.4	Clients für die Verwendung mit Discovery deklarativ konfigurieren ..	191
5.2.5	Ankündigungen (Announcements)	192
5.3	Ereignisse und Publish-/Subscribe-Szenarien	194
5.3.1	Service-Verträge	194
5.3.2	Implementierung eines Publish-/Subscribe-Service	195
5.3.3	Konfiguration	197
5.3.4	Implementierung des zu benachrichtigenden Clients	198
5.3.5	Weiterführende Überlegungen	199
5.4	ESB-Lösungen mit WCF entwickeln	199
5.4.1	Catch-All-Verträge	199
5.4.2	Nachrichtentransformation	200
5.5	Freie ESB-Implementierung	204
6	ASP.NET Web API	205
6.1	REST-Services mit ASP.NET Web API erstellen	205
6.1.1	Parameter und Rückgabewerte	206
6.1.2	Dynamische Parameter	208
6.1.3	REST-Services konfigurieren	209
6.1.4	REST-Services mit Fiddler testen	210
6.1.5	Mehr Kontrolle über HTTP	211
6.1.6	REST-Services über HttpClient konsumieren	214
6.1.7	Hilfe-Seiten	217
6.2	Tracing	221
6.2.1	Standard-Implementierung von ITraceWriter	221
6.2.2	Eigenen TraceWriter implementieren	222
6.3	OData-Unterstützung	223
6.4	Self-Hosting mit Web APIs	223
6.5	Querschnittsfunktionen an globalen Stellen platzieren	224

6.5.1	Querschnittsfunktionen mit Message-Handler implementieren	225
6.5.2	Handler mit HttpClient verwenden	226
6.5.3	Querschnittsfunktionen mit Filter realisieren	228
6.6	Erweiterte Konfigurationsmöglichkeiten	232
6.6.1	Benutzerdefinierte Routen	232
6.6.2	Controller-basierte Konfiguration	232
6.6.3	Routen-basierte Konfiguration	233
6.7	Deklaratives Validieren von Parametern	234
6.7.1	Verwenden von Data-Attributen	234
6.7.2	Auswerten von Validierungsattributen	236
6.7.3	Benutzerdefinierte Validierungsattribute	237
6.8	Benutzerdefinierte Formate unterstützen	238
6.8.1	Formatter implementieren	238
6.8.2	Formatter serverseitig registrieren und testen	240
6.8.3	Formatter mit HttpClient verwenden	240
6.8.4	Binäre Serialisierung mit BSON	241
6.9	Serialisierung beeinflussen	241
6.9.1	JSON-Serializer konfigurieren	242
6.9.2	XML-Serializer konfigurieren	243
6.9.3	Eigenschaften von der Serialisierung ausnehmen	243
6.9.4	Zirkuläre Referenzen serialisieren	243
6.10	Streaming	246
6.10.1	Action-Methoden für Streaming vorbereiten	246
6.10.2	Streaming in Self-Hosting-Szenarien konfigurieren	247
6.10.3	Streaming für IIS konfigurieren	247
6.10.4	Streams über HttpClient verwenden	249
6.11	Fortschritt ermitteln	250
6.12	Web API und HTML-Formulare	251
6.12.1	Einfache Formular-Felder übermitteln	251
6.12.2	Dateiupload via HTML-Formular	252
6.13	ASP.NET Web API erweitern	254
6.13.1	Abhängigkeiten auflösen mit benutzerdefiniertem DependencyResolver	254
6.13.2	Zusätzliche Assemblies mit AssemblyResolver laden	255
6.13.3	Service-Operationen über HttpActionSelector auswählen	256
6.13.4	Controller über HttpControllerSelector auswählen	257
6.13.5	Methodenparameter auf benutzerdefinierte Weise mit HttpParameterBinding	259
7	ASP.NET Web API Security	263
7.1	Verschlüsselte Übertragung	263
7.1.1	SSL mit IIS	263
7.1.2	SSL in Self-Hosting-Szenarien	263
7.1.3	Diskussion über Nachrichtensicherheit	264
7.2	Authentifizierung und Autorisierung	264

7.2.1	Operationen absichern	264
7.2.2	HTTP-Authentifizierung mit IIS	266
7.2.3	HTTP-Authentifizierung in Seif-Hosting-Szenarien	267
7.2.4	Benutzer mit HttpClient authentifizieren	268
7.2.5	Benutzerdefinierte Security mit Handler	268
7.2.6	Benutzerdefinierte Security mit HTTP-Module	270
7.2.7	Mit Client-Zertifikaten arbeiten	273
7.3	Single Sign-On mit OAuth2 und DotNetOpenAuth	277
7.3.1	OAuth2	277
7.3.2	OAuth2 und REST-Services	278
7.3.3	Implementieren eines Authorization Servers mit DotNetOpenAuth	278
7.3.4	Client für OAuth2-Szenario	284
7.3.5	Service mittels OAuth2 absichern	285
8	ASP.NET SignalR	287
8.1	Long-Polling	287
8.2	Web Sockets	288
8.3	ASP.NET SignalR	288
8.4	PersistentConnection	289
8.4.1	Erste Schritte mit SignalR und PersistentConnection	289
8.4.2	Lifecycle-Methoden	290
8.4.3	URL-Mapping für PersistentConnection	291
8.4.4	Einfacher Client für eine PersistentConnection	291
8.4.5	Einfacher JavaScript-Client für eine PersistentConnection	292
8.4.6	Daten über serverseitige Prozesse an Connection senden	294
8.5	Hubs	294
8.5.1	Methoden und Callbacks mit SignalR und Hubs	294
8.5.2	URL-Mapping für Hubs	296
8.5.3	Lifecycle-Methoden	296
8.5.4	Hubs konsumieren	296
8.5.5	Hubs über JavaScript konsumieren	299
8.5.6	Gruppen	302
8.5.7	Hubs über serverseitige Prozesse benachrichtigen	303
8.6	Pipeline-Modules für Querschnittsfunktionen	303
8.7	SignalR konfigurieren	305
8.8	SignalR skalieren	305
9	Hosting von WCF- und Web API-Diensten	307
9.1	Hosting in Windows-Systemdiensten	307
9.1.1	Erstellen eines Windows-Systemdienstes	308
9.1.2	Ergänzen eines WCF-Dienstes in einem Systemdienst	310
9.1.3	Ergänzen eines Web API-Dienstes in einem Systemdienst	311
9.1.4	Trick für das Debugging eines Systemdienstes	311
9.1.5	Vorbereiten der Installation eines Systemdienstes	314
9.1.6	Installation eines Systemdienstes	315

9.2	Hosting im Internet Information Server (IIS)/Windows Server AppFabric	317
9.2.1	Von IIS über WAS zu AppFabric	317
9.2.2	IIS-Sites	318
9.2.3	IIS-Anwendungen	324
9.2.4	IIS-Anwendungspools	325
9.2.5	Autostart	331
9.3	WCF-Projekte für den IIS	334
9.3.1	.svc-Dateien	335
9.3.2	Test des Dienstes	335
9.3.3	Abruf der Metadaten	336
9.3.4	Konfigurationsdatei	337
9.3.5	WCF ohne SVC-Datei	339
9.3.6	Eigene ServiceHostFactory	339
9.4	ASP.NET Web API-Projekte für den IIS	340
9.5	Installieren von WCF- und Web API-Diensten im IIS	340
9.5.1	Manuelles Verbreiten von Diensten (XCopy-Deployment)	340
9.5.2	Verbreiten mit der Funktion „Build/Publish“ in Visual Studio direkt auf einen IIS	341
9.5.3	Verbreiten von Diensten mit dem IIS Web Deployment Tool (MSDeploy)	342
9.6	Konfiguration und Monitoring mit den „AppFabric“-Erweiterungen	352
9.6.1	Installation	353
9.6.2	AppFabric-Ansichten	354
9.6.3	Überwachungsfunktionen	356
9.6.4	Weitere Konfigurationsmöglichkeiten	358
10	Service Bus und Access Control für Windows Azure und Windows Server	361
10.1	Überblick	361
10.2	Namespace einrichten	362
10.3	Relaying	363
10.3.1	Relay-Bindings	364
10.3.2	Service über Relaying bereitstellen	364
10.3.3	Service über Relaying konsumieren	367
10.4	Lose Kopplung mit Queues und Topics	368
10.4.1	Zugriff auf Queues und Topics	370
10.4.2	Transaktionen	370
10.4.3	Sitzungen	371
10.4.4	Metadaten, Filtern und Stamping	371
10.4.5	Queues über die .NET-API ansprechen	371
10.4.6	Queues programmatisch erzeugen	372
10.4.7	Nachrichten an Queues senden	373
10.4.8	Nachrichten aus Queues abrufen	374
10.4.9	Mit Sitzungen arbeiten	376
10.4.10	Aus Dead Letter Queue lesen	377

10.4.11 Topics und Abonnenten einrichten	378
10.4.12 Nachrichten unter Verwendung von Topics senden und abrufen	380
10.4.13 Queues über WCF ansprechen	382
10.4.14 WCF-Sitzungen realisieren	385
10.4.15 Topics via WCF abonnieren	386
10.5 Windows Azure Access Control	387
10.5.1 AC konfigurieren	387
10.5.2 Service mit AC absichern	389
10.5.3 Azure Service Bus mit AC absichern	393
10.6 Windows Server Service Bus	394
10.6.1 Namespaces administrieren	395
10.6.2 Auf Queues und Topics zugreifen	395
10.6.3 Zugriff auf Queues und Topics einschränken	396
11 Datenbasierte Services	397
11.1 Was sind datenbasierte Services?	397
11.2 Services und ADO.NET Entity Framework	397
11.2.1 Vom DataSet zum Entity Framework	397
11.2.2 Grundlagen der Self-Tracking Entities	399
11.2.3 Beispianwendung	403
11.3 ADO.NET Entity Framework Code First (Code Only)	412
11.3.1 Die drei Ebenen des ADO.NET Entity Framework	412
11.3.2 Vorbereitungen	413
11.3.3 Entitätsklassen	413
11.3.4 Kontextklasse	418
11.3.5 Nutzen der Kontextklasse	419
11.3.6 Einsatzbeispiel	420
11.3.7 Konvention vor Konfiguration	423
11.3.8 Codebasierte Konfiguration	425
11.3.9 Schemamigrationen	426
11.3.10 Entity Framework Power Tools	427
11.3.11 Code Only und Webservices	429
11.4 Kritik an der Implementierung eines datenbasierten Service mit WCF	430
11.5 WCF Data Services (Open Data Protocol)	431
11.5.1 Lizenz und Standardisierung	431
11.5.2 Bezug der WCF Data Services	431
11.5.3 Andere Bibliotheken	432
11.5.4 Rahmenbedingungen	433
11.5.5 OData-Beispiel	433
11.5.6 Architektur	435
11.5.7 Abfragesyntax	436
11.5.8 Einen WCF Data Service erstellen	437
11.5.9 Zugriffsrechte	437
11.5.10 Fehlermeldungen aktivieren	439
11.5.11 Einen WCF Data Service im Webbrowser testen	440
11.5.12 Abruf der Metadaten	442

11.5.13 Einen WCF Data Service mit Fiddler testen	443
11.5.14 Einen .NET-basierten Client erstellen	443
11.5.15 Tipps und Tricks	448
11.6 WCF RIA Services	457
11.6.1 Architektur der RIA Services	458
11.6.2 Einen RIA Service erstellen	459
11.6.3 Einen RIA Service nutzen	463
11.7 Datenbasierte REST-Services mit ASP.NET Web API und dem Open Data Protocol (OData)	466
11.7.1 Daten mit OData flexibel abfragen	466
11.7.2 Mögliche OData-Abfragen einschränken	468
11.7.3 OData-Abfragen global aktivieren	470
11.7.4 OData-Abfragen manuell auswerten	470
11.7.5 Daten mit OData verwalten	471
11.8 Vergleich und Fazit	475
12 Workflows und Workflow Services	479
12.1 Überblick	479
12.2 Visual Studio-Projektvorlagen	480
12.3 Sequenzielle Workflows	480
12.3.1 Sequenzielle Workflows erstellen	480
12.3.2 Sequenzielle Workflows ausführen	481
12.3.3 Kontrollfluss	482
12.4 Flussdiagramme	484
12.5 Zustandsautomaten (State Machines)	485
12.6 Transaktionen und Kompensation	487
12.6.1 Transaktionen	487
12.6.2 Kompensation	487
12.7 WCF Workflow Services	489
12.7.1 Workflow Services erstellen	489
12.7.2 Workflow Services testen	490
12.7.3 Workflow Services konfigurieren	491
12.7.4 Korrelation	492
12.7.5 Contract-First (ab .NET 4.5)	493
12.8 Benutzerdefinierte Aktivitäten	495
12.8.1 CodeActivity	495
12.8.2 AsyncCodeActivity	496
12.8.3 NativeActivity	497
12.8.4 Anpassen der Darstellung benutzerdefinierter Aktivitäten	500
12.9 Bookmarks	502
12.10 Persistenz	503
12.10.1 SQLWorkflowInstanceStore	503
12.10.2 Eigenschaften höherstufen	506
12.10.3 Höherstufen von Eigenschaften bei Verwendung von WCF Workflow Services	508

12.11	Versionisierung und Aktualisierung von Workflows (ab .NET 4.5)	510
12.11.1	Side-by-Side-Versionisierung	510
12.11.2	Dynamic Update	516
12.11.3	Dynamic Update mit WorkflowControlEndpoint	521
12.12	Ablaufverfolgung (Tracking)	523
12.12.1	Benutzerdefinierte Tracking Records erzeugen	523
12.12.2	Tracking Records abonnieren	524
12.12.3	Tracking Records einsehen	526
12.13	Workflow-Designer in eigenen Anwendungen hosten	527
12.14	WCF-Dienste und -Services in AppFabric hosten	528
12.14.1	Monitoring	529
12.14.2	Persistenz	531
12.14.3	Weitere Möglichkeiten	532
13	Das WCF-Erweiterungsmodell	533
13.1	Übersicht	533
13.1.1	Was sich erweitern lässt	533
13.1.2	Die Laufzeitumgebung der Applikation erweitern	534
13.2	Erweiterung des Sicherheitsmodells	534
13.2.1	Verantwortungsbereich der WCF Security Component	534
13.2.2	Das WebService-(WS-)Sicherheitsmodell	534
13.2.3	Implementierung der WebService-(WS-)Sicherheit	535
13.3	Erweiterung des Bindungssystems	539
13.3.1	Bindungen und Bindungselemente	540
13.3.2	Nachrichtenkanäle	540
13.3.3	Benutzerdefinierte Bindungen	541
13.4	Weitere Erweiterungsfunktionen	542
13.4.1	Erweiterung des Metadatenmodells	542
13.4.2	Erweiterungen der Serialisierung	543
14	WCF erweitern	545
14.1	ServiceHost und Dienstmodellebene	545
14.1.1	Aufbau der Dienstmodellebene	545
14.1.2	Erweiterung mittels Verhalten	546
14.1.3	Erweiterung von Clients	549
14.1.4	Erweiterung des Dispatchers	554
14.1.5	Erweiterbare Objekte	562
14.1.6	Erweitertes Hosting – die Klasse ServiceHostFactory	565
14.2	Erweiterung des Bindungsmodells	567
14.2.1	Konfigurieren mittels Bindungselementen	567
14.2.2	Bindungen selbst erstellen	568
14.2.3	Erstellen eines eigenen Bindungselements	570
14.3	Channel Layer- und Peer Channel-Erweiterungen	572
14.3.1	Funktionsweise	573
14.3.2	Benutzerdefinierte Übertragungskanäle	573
14.3.3	Benutzerdefinierte Nachrichtenencoder	583

14.4	Das Metadatensystem	585
14.4.1	Funktionsweise	586
14.4.2	Anwendung	586
14.5	Serialisierung und Encoder	591
14.5.1	Ersetzen von Datenverträgen	591
14.5.2	Eigene Formatierer	597
14.6	Erweiterung des Sicherheitsmodells	603
14.6.1	Aufbau der Sicherheitsarchitektur	603
14.6.2	Benutzerdefinierte Anmeldeinformation und Token	603
14.6.3	Benutzerdefinierte Nachrichtenverschlüsselung	621
Anhang - NuGet		631
Index		635