

Inhaltsverzeichnis

Jürgen Dunkel, Andreas Eberhart, Stefan Fischer, Carsten Kleiner, Arne
Koschel

Systemarchitekturen für Verteilte Anwendungen

Client-Server, Multi-Tier, SOA, Event-Driven Architectures, P2P, Grid,
Web 2.0

ISBN: 978-3-446-41321-4

Weitere Informationen oder Bestellungen unter

<http://www.hanser.de/978-3-446-41321-4>

sowie im Buchhandel.

Inhaltsverzeichnis

Teil I	Einführung	1
1	Motivation und Überblick	3
2	Softwarearchitekturen	7
2.1	Der Begriff „Softwarearchitektur“	7
2.2	Leitgedanken zur Strukturierung von Software	8
2.3	Kriterien für gute Softwarearchitekturen	9
2.4	Die Dimensionen verteilter Systeme	11
2.4.1	Verteilung und Kommunikation	12
2.4.2	Nebenläufigkeit	13
2.4.3	Persistenz	14
2.5	Existierende Softwarearchitekturen für verteilte Systeme	15
Teil II	Architekturen für verteilte Systeme	19
3	Client-Server-Architekturen	21
3.1	Architekturkonzept	21
3.1.1	Einführung	21
3.1.2	Eigenschaften des Client-Server-Modells	22
3.2	Realisierungsplattformen	25
3.2.1	WWW-Clients und -Server	25
3.2.2	Sockets	27
3.2.3	RPC am Beispiel Java Remote Method Invocation	28
3.2.4	Client und Datenbank-Server	30
3.3	Code-Beispiele	32
3.3.1	Sockets	32

3.3.2	RPC mit Java RMI	35
3.3.3	DB-Client und DB-Server	37
4	3- und N-Tier-Architekturen	41
4.1	Architekturkonzepte	42
4.1.1	Dreischichtige Architekturen	42
4.1.2	Mehrschichtige Architekturen	46
4.2	Realisierungsplattformen	49
4.2.1	Klassische Web 1.0-Anwendungsarchitekturen	49
4.2.2	Verteilte Objekte am Beispiel CORBA	51
4.2.3	JEE	55
4.2.4	.NET	63
4.3	Code-Beispiele	72
4.3.1	Klassische Web 1.0-Anwendungsarchitekturen	72
4.3.2	Verteilte Objekte am Beispiel CORBA: Code	75
4.3.3	JEE	78
4.3.4	.NET	83
5	SOA	89
5.1	Architekturkonzept	89
5.1.1	Motivation	89
5.1.2	Struktur von SOAs	90
5.2	Web Services	92
5.2.1	Motivation, Historie und Standardisierung	92
5.2.2	SOAP	94
5.2.3	WSDL	98
5.2.4	UDDI	100
5.2.5	WS-BPEL	100
5.2.6	WS-I	102
5.2.7	WS-*	102
5.2.8	Fragestellungen in der Praxis	103
5.2.9	Bewertung der Web Service Standards	104
5.3	Realisierungsplattformen	105
5.3.1	.NET	105
5.3.2	Apache Axis	107
5.3.3	Open Enterprise Service Bus	110

5.3.4	Oracle WS-BPEL Engine	112
5.4	Code-Beispiele	113
5.4.1	Java / Axis	113
5.4.2	.NET	115
5.4.3	WS-BPEL	116
6	Event-Driven Architecture (EDA)	119
6.1	Architekturkonzept	120
6.1.1	Ereignis-orientierte Softwarearchitektur	121
6.1.2	Complex Event Processing	124
6.1.3	EDA-Referenzarchitektur	133
6.1.4	Vorgehen bei der Entwicklung von EDA-Anwendungen	134
6.1.5	Aktueller Entwicklungsstand	135
6.2	Realisierungsplattformen	136
6.3	Code-Beispiele	137
7	Peer-to-Peer	141
7.1	Architekturkonzept	142
7.1.1	Was ist P2P?	142
7.1.2	Zentrale Architektur – Napster	145
7.1.3	Verteilte Architektur – Gnutella	146
7.1.4	Distributed Hash Tables	149
7.1.5	Chord	151
7.1.6	Split-Stream-Protokolle	153
7.1.7	Bedeutung und Einordnung von P2P-Netzen	155
7.2	Realisierungsplattformen	155
7.2.1	JXTA	156
7.2.2	Peer-to-Peer-Netze in der Praxis	158
8	Grid-Architekturen	161
8.1	Architekturkonzept	162
8.1.1	Allgemeines	163
8.1.2	Arten von Grids	165
8.1.3	OGSA	166
8.1.4	Weiterführende Literatur	167
8.2	Realisierungsplattformen	168

8.2.1	Konzeptionelle Realisierungen der OGSA	169
8.2.2	Unabhängige Implementierungen	174
8.2.3	Herstellerspezifische Implementierungen	177
8.3	Code-Beispiele	179
8.3.1	Globus Toolkit GT4	179
8.3.2	Amazon	180
9	Web 2.0 und Web-orientierte Architekturen	185
9.1	Architekturkonzept	187
9.1.1	Keep it Simple!	187
9.1.2	Hochskalierbare Systeme mit REST	188
9.1.3	AJAX: Neue Wege im Design von Web-basierten Benutzer- schnittstellen	189
9.1.4	JSON als leichtgewichtiger Ersatz für XML	195
9.1.5	Event-basierte Programmierung mit Feeds	196
9.1.6	Mashups: Daten- und Applikationsintegration im Browser	198
9.1.7	Architektonische Probleme bei Mashups und AJAX	199
9.2	Realisierungsplattformen	202
9.2.1	AJAX-Werkzeuge	202
9.2.2	UI Libs	203
9.2.3	Mashup IDEs	204
9.2.4	Alternative Clients	204
9.3	Code-Beispiele	206
9.3.1	REST Client in Java	206
9.3.2	JavaScript Mashup	208
Teil III Auswahl einer konkreten Architektur		211
10	Vergleichskriterien zur Architekturwahl	213
10.1	Anforderungen aus dem Softwarelebenszyklus	214
10.1.1	Analyse und Design	215
10.1.2	Entwicklung und Test	216
10.1.3	Betrieb	217
10.1.4	Management und Umfeld	218
10.1.5	Analyse der Architekturen	219
10.2	Anforderungen der Anwendungen	234

10.2.1	Grad an Interaktivität	235
10.2.2	Zahl der Teilnehmer	235
10.2.3	Ressourcenbedarf	236
10.2.4	Dynamik	236
10.2.5	Robustheitsanforderungen	236
10.2.6	Anwendungsgebiet	237
10.3	Zusammenfassung der Architekturbewertung	237
11	Verteilte Anwendungen: Fallbeispiele aus der Praxis	239
11.1	Fallbeispiele „Klassische Web-Anwendungsarchitekturen und Verteilte Objekte“	239
11.1.1	Klassische Web 1.0-Anwendungsarchitekturen	239
11.1.2	3-Tier Web- und verteilte Objekte-Anwendung mit CORBA: „UIS-Föderationsarchitektur“	240
11.2	Fallbeispiele „N-tier-Architekturen“	241
11.2.1	.NET: „3-Schicht-Anwendung vita.NET“	242
11.2.2	Java EE/J2EE: „Standard-Web-Anwendungen PetStore und Duke’s Bank“	245
11.3	Fallbeispiele „SOA“	247
11.3.1	SOA und Web Services: „Amazon.com“	248
11.3.2	SOA und ESB: „Einführung in einem mittelständischen Versicherungsunternehmen“	248
11.3.3	SOA, CORBA und J2EE: „Erfahrungen bei der Migration eines IMS-basierenden Kernbankenverfahrens in eine Serviceorientierte Architektur“	252
11.4	Fallbeispiele „Peer-to-Peer“	257
11.5	Fallbeispiele „Grid“	257
11.5.1	Huge Scale Grid: „Worldwide LHC Computing Grid (WLCG)“	257
11.5.2	Kleine Grids: „ViSoGrid“	259
11.6	Fallbeispiel Web 2.0: „Flickr“	261
Teil IV	Ausblick und Zusammenfassung	263
12	Künftige Entwicklungen	265
12.1	Software as a Service	265
12.2	Virtualisierung	266

12.3 Appliances	268
12.4 Cloud Computing	270
12.5 Semantic Web	271
12.6 Ubiquitous Computing	273
12.7 Ultra-Large-Scale Systems	274
13 Zusammenfassung	277
Literaturverzeichnis	285
Stichwortverzeichnis	287