

Table of contents

1 What is software architecture?	1
1.1 Software architecture as abstraction	2
1.2 Software architecture as blueprint	3
1.3 Software architecture terminology	5
1.4 What is software architecture?.....	7
1.5 What is "good" software architecture?	8
1.6 Software architecture vs. system architecture.....	10
References	11
2 MobiTeam: A design review case study	13
2.1 Case study: Design review of cell phone software	14
2.1.1 The design review process	17
2.1.2 Software support for distributed product development	19
2.2 Architectural concepts and abstractions	23
2.2.1 A first look on the architecture of MobiTeam.....	25
2.2.2 MobiTeam's software architecture	30
2.3 Concluding remarks.....	36
References	37
3 Architectural design.....	39
3.1 Model-driven development.....	39

3.2 What is software architecture?.....	43
3.2.1 Goals of an architectural design.....	45
3.2.2 Design of a product family.....	46
3.2.3 Structure of complex software systems.....	47
3.3 Views on a software system	48
3.3.1 Conceptual (logical) view	50
3.3.2 Module view	51
3.3.3 Process view.....	52
3.3.4 Physical view	52
3.3.5 Scenarios.....	53
3.3.6 Application of the „4+1“ views model.....	53
3.4 Software architecture description	54
3.4.1 Software architecture elements	54
3.4.2 Architectural descriptions	56
3.5 Acme – an architecture description language	57
3.5.1 Architectural structure.....	57
3.5.2 Properties of architecture elements	59
3.5.3 Design constraints.....	60
3.5.4 Program families.....	62
3.5.5 The Acme architectural design tool	63
3.5.6 Conclusions.....	64
3.6 Documenting a software architecture	64
References	66
4 Architectural principles, building blocks and patterns	69
4.1 Quality properties of a software architecture.....	69
4.2 Architectural styles.....	74
4.2.1 Data-centric software architectures.....	76
4.2.2 Data-flow architectures	77
4.2.3 Virtual machine architectures	78
4.2.4 Call-and-return architectures.....	79
4.2.5 Independent component architectures.....	80
4.2.6 Heterogeneous architectural styles.....	82
4.2.7 Application of architectural styles	83
4.3 Architectural styles and quality attributes	85
4.4 Architectural styles and patterns.....	89
4.4.1 Proxy.....	91
4.4.2 Broker	94
4.4.3 Master-slave.....	98
4.4.4 Client-dispatcher-server	101
4.4.5 Publisher-subscriber.....	104
4.4.6 Model-view-controller (MVC).....	106
4.5 Concluding remarks.....	110
References	111
5 Web services	113

5.1 Introduction	113
5.2 Simple Object Access Protocol (SOAP).....	117
5.2.1 Structure of a SOAP message	118
5.3 Web Service Description Language (WSDL).....	120
5.3.1 Structure of WSDL documents	121
5.3.2 Communication patterns	122
5.4 Universal Description, Discovery and Integration.....	125
5.4.1 Business registry	127
5.4.2 Business registrations.....	127
5.4.3 Service-type registration	128
5.4.4 Business information – businessEntity.....	128
5.4.5 Service information – businessService.....	129
5.4.6 Binding information – bindingTemplate.....	129
5.4.7 The tModel.....	130
5.4.8 Cooperation of UDDI and SOAP	131
5.5 Composition of Web services	135
5.5.1 Interaction patterns of basic web services.....	136
5.5.2 Interaction patterns of composed web services	138
5.6 Concluding remarks.....	141
References	141
6 Workflow and coordination	143
6.1 Introduction	143
6.2 Terms	144
6.3 Architectural views of workflow systems	145
6.4 Example systems	147
6.4.1 Groove.....	147
6.4.2 Caramba	150
6.5 Web services workflows.....	152
6.5.1 Business process execution language for web services.....	154
6.6 Concluding remarks.....	159
References	159
7 Peer-to-peer architectures	161
7.1 Introduction	161
7.2 Classification of the peer-to-peer approach.....	163
7.2.1 Peer-to-peer vs. Event based	164
7.2.2 Peer-to-peer vs. push.....	164
7.2.3 Peer-to-peer vs. mobile agents	165
7.2.4 Peer-to-peer vs. distributed databases	166
7.2.5 Application domains	167
7.3 The peer-to-peer model	168
7.3.1 Characteristic architectural parameters	169
7.3.4 Further criteria.....	171
7.4 Example peer-to-peer architectures	171
7.4.1 Napster	171

7.4.2 Gnutella.....	172
7.4.3 FastTrack	177
7.4.4 Freenet	177
7.4.5 P-Grid.....	183
7.4.6 Further approaches.....	190
7.5 Application domains.....	193
7.6 Concluding remarks.....	195
References	195
8 Component models.....	199
8.1 Introduction	199
8.2 Component technology.....	199
8.2.1 Components and objects	200
8.2.2 Distributed object models	200
8.3 Business-to-business architectures und component models	204
8.4 Architectures of J2EE and .NET	206
References	210
9 Security	211
9.1 Security in the software life-cycle	211
9.2 Cryptography basics	212
9.2.1 Hash functions	212
9.2.2 Symmetric and asymmetric encryption.....	213
9.2.3 Certificates and digital signature.....	214
9.2.4 Public key infrastructure	216
9.3 Security aspects	216
9.4 Architectural aspects	219
9.4.1 Confidential authenticated communication.....	220
9.4.2 Secure Socket Layer.....	222
9.4.3 Firewalls.....	224
9.5 Concluding remarks.....	226
References	227
10 Push and event architectures	229
10.1 Push systems.....	229
10.1.1 A component model for push systems	230
10.1.2 An application scenario for push systems	237
10.1.3 Possible applications	239
10.1.4 Concluding remarks	240
10.2 Event-based systems.....	240
10.2.1 Concepts of Event-based systems	241
10.2.2 Comparison of event-based systems and push systems	242
10.3 Concluding remarks.....	243
References	243
11 Electronic payment	245

11.1 E-payment scenario	246
11.2 Classification parameters.....	247
11.3 Payment instruments.....	248
11.3.1 Credit cards	248
11.3.2 Electronic money	250
11.3.3 Electronic checks	251
11.4 Security requirements	252
11.5 Example architectures.....	254
11.5.1 Secure Electronic Transaction.....	254
11.5.2 Millicent	255
11.6 Legal aspects	258
References	258
Index.....	259