

Table of Contents

Preface	1
Chapter 1: Introduction	7
New demands in enterprise systems	7
Modern way of realizing enterprise systems	8
Relevance of Java EE in modern systems	9
Java EE 8 update and roadmap	10
Java Community Process	11
What to expect in the book	12
Chapter 2: Designing and Structuring Java Enterprise Applications	15
The purpose of enterprise applications	15
What developers should focus on	16
Meeting customer's demands	16
Outer enterprise project structure	18
Business and team structures	18
Software projects contents	19
Application source code	19
Software structures	19
Version control systems	21
Binaries	22
Build systems	23
Single versus multi-module projects	23
Illusion of reusability	24
Technical dependencies	24
Organizational challenges	25
Reusability considerations	25
Project artifacts	26
One project per artifact	26
Build systems for Java EE	27
Apache Maven	28
Gradle	30
Structuring for modern frontend technologies	33
Enter JavaScript frameworks	33
Organizing modern frontends	34
Enterprise project code structure	35
Situation in enterprise projects	36
Horizontal versus vertical layering	36

Business-driven structure	37
Designing reasonable modules	38
Realizing package structures	40
Package contents	40
Horizontal package layering	41
Flat module package	42
Entity Control Boundary	43
Packages	44
Package access	45
Don't over-enforce architecture	46
Summary	47
Chapter 3: Implementing Modern Java Enterprise Applications	49
Use case boundaries	49
Core domain components of modern Java EE	50
EJB and CDI - differentiation and integration	51
CDI producers	53
Emitting domain events	53
Scopes	55
Patterns in Java EE	56
Design patterns revisited	56
Singleton	56
Abstract factory	59
Factory method	62
Object pool	63
Decorator	64
Facade	68
Proxy	70
Observer	70
Strategy	71
Further patterns	73
Domain-Driven Design	74
Services	74
Entities	75
Value objects	75
Aggregates	76
Repositories	76
Factories	77
Domain event	77
External and cross-cutting concerns in enterprise applications	78
Communication with external systems	78
How to choose communication technology	78
Synchronous HTTP communication	81
Representational State Transfer	82
Java API for RESTful web services	85

Mapping HTTP content types	88
Validating requests	95
Mapping errors	101
Accessing external systems	102
Stability when consuming HTTP	103
Accessing Hypermedia REST services	106
Asynchronous communication and messaging	108
Asynchronous HTTP communication	108
Message-oriented communication	109
Server-sent events	110
WebSocket	114
Connecting enterprise technology	116
Database systems	116
Integrating RDBMS systems	117
Mapping domain models	118
Integrating database systems	121
Transactions	124
Relational databases versus NoSQL	125
Cross-cutting concerns	126
Configuring applications	129
Caching	131
Flow of execution	133
Synchronous execution	134
Asynchronous execution	134
Asynchronous EJB methods	135
Managed Executor Service	135
Asynchronous CDI events	136
Scopes in asynchronicity	137
Timed execution	137
Asynchronous and reactive JAX-RS	139
Concepts and design principles of modern Java EE	143
Preserving maintainable code with high quality	144
Summary	146
Chapter 4: Lightweight Java EE	149
Lightweight enterprise technology	149
Why Java EE standards?	150
Convention over configuration	151
Dependency management of Java EE projects	152
Lightweight way of packaging applications	154
Java EE application servers	158
One application per application server	159
Summary	160
Chapter 5: Container and Cloud Environments with Java EE	163

Motivations and goals	163
Infrastructure as code	165
Stability and production readiness	166
Containers	167
Java EE in the container	169
Container orchestration frameworks	171
Realizing container orchestration	173
Java EE in orchestrated containers	179
Connecting external services	179
Configuring orchestrated applications	180
12-factor applications and Java EE	182
Have one codebase tracked in revision control, many deploys	182
Explicitly declare and isolate dependencies	183
Store config in the environment	184
Treat backing services as attached resources	184
Strictly separate build and run stages	185
Execute the app as one or more stateless processes	185
Export services via port binding	186
Scale out via the process model	186
Maximize robustness with fast startup and graceful shutdown	187
Keep development, staging, and production as similar as possible	188
Treat logs as event streams	189
Run admin/management tasks as one-off processes	190
Cloud, Cloud native, and their benefits	191
Cloud native	192
Summary	193
Chapter 6: Application Development Workflows	195
Motivation and goals of productive development workflows	196
Realizing development workflows	197
Version control everything	198
Building binaries	199
Java artifacts	199
Artifact versions	200
Building containers	201
Quality assurance	202
Deployment	204
Configuration	206
Credentials	208
Data migration	209
Adding database structures	210

Changing database structures	211
Removing database structures	212
Implementing migration	212
Testing	215
Build metadata	216
Going to production	218
Branching models	218
Technology	220
Pipeline-as-code	220
Workflows with Java EE	224
Continuous Delivery culture and team habits	225
Responsibility	225
Check in early and often	226
Immediately fixing issues	227
Visibility	227
Improve continuously	228
Summary	229
Chapter 7: Testing	231
The necessity of tests	231
Requirements of well-crafted tests	232
Predictability	233
Isolation	233
Reliability	234
Fast execution	234
Automation	234
Maintainability	235
What to test	235
Definition of test scopes	236
Unit tests	237
Component tests	237
Integration tests	237
System tests	238
Performance tests	238
Stress tests	239
Implementing tests	239
Unit tests	240
Implementation	240
Technology	244
Component tests	245
Motivation	245

Implementation	245
Delegating test components	248
Technology	250
Integration tests	251
Embedded containers	251
Embedded databases	253
Running integration tests	256
Code level integration tests versus system tests	257
Shortcomings of integration tests	257
Shortcomings of system tests	258
Conclusion	258
System tests	259
Managing test scenarios	260
Simulating external concerns	261
Designing system tests	262
Deploying and controlling external mocks	266
Performance tests	268
Motivation	269
Key performance indicators	269
Developing performance tests	270
Insights	273
Running tests locally	273
Maintaining test data and scenarios	276
Importance of maintainable tests	276
Signs of lack of test quality	277
Test code quality	277
Test technology support	280
Summary	282
Chapter 8: Microservices and System Architecture	285
Motivations behind distributed systems	286
Challenges of distribution	286
Communication overhead	287
Performance overhead	287
Organizational overhead	287
How to design systems landscapes	288
Context maps and bounded contexts	288
Separation of concerns	289
Teams	289
Project life cycles	290
How to design system interfaces	290
API considerations	291
Interface management	291

Change-resilient APIs	292
Breaking the business logic	292
Hypermedia REST and versioning	292
Documenting boundaries	293
Consistency versus scalability	295
Event sourcing, event-driven architectures, and CQRS	296
Shortcomings of CRUD-based systems	297
Scalability	297
Competing transactions	297
Reproducibility	298
Event sourcing	298
Benefits	299
Eventually consistent real world	300
Event-driven architectures	301
Eventual consistency in event-driven architectures	303
Enter CQRS	303
Principles	304
Design	304
Benefits	308
Shortcomings	309
Communication	310
Microservice architectures	310
Sharing data and technology in enterprises	311
Shared-nothing architectures	312
Interdependent systems	312
12-factor and cloud native applications	313
When to use and when not to use microservices	314
Implementing microservices with Java EE	314
Zero-dependency applications	315
Application servers	315
Implementing application boundaries	316
Implementing CQRS	316
System interfaces	317
Example scenario using Apache Kafka	317
Integrating Java EE	318
CDI events	319
Event handlers	320
State representation	321
Consuming Kafka messages	322
Producing Kafka messages	325
Application boundaries	326
Integrating further CQRS concepts	328
Java EE in the age of distribution	329

Discovering services	329
Communicating resiliently	330
Validating responses	330
Breaking timeouts and circuits	330
Bulkheads	332
Shaking hands and pushing back	333
More on being resilient	335
Summary	335
Chapter 9: Monitoring, Performance, and Logging	337
Business metrics	338
Collecting business metrics	339
Emitting metrics	340
Enter Prometheus	341
Realization with Java EE	343
Integrating the environment	344
Meeting performance requirements in distributed systems	345
Service level agreements	345
Achieving SLAs in distributed systems	345
Tackling performance issues	346
Theory of constraints	347
Identifying performance regression with jPDM	348
Subsystems	348
Actors	349
Application	350
JVM	350
Operating system and hardware	350
jPDM instances - production situations	350
Analyzing the jPDM instances	351
Dominating consumer - OS	353
Dominating consumer - none	354
Dominating consumer - JVM	354
Dominating consumer - application	355
Conclusion	355
Technical metrics	356
Types of technical metrics	356
High frequency monitoring versus sampling	357
Collecting technical metrics	357
Boundary metrics	358
Logging and tracing	359
Shortcomings of traditional logging	359
Performance	360
Log levels	360
Log format	361

Amounts of data	361
Obfuscation	362
The concerns of applications	362
Wrong choice of technology	362
Logging in a containerized world	364
Journaling	366
Tracing	366
Tracing in a modern world	369
Typical performance issues	371
Logging and memory consumption	371
Premature optimization	372
Relational databases	373
Communication	374
Threading and pooling	375
Performance testing	375
Summary	376
Chapter 10: Security	379
Lessons learned from the past	379
Security in a modern world	380
Security principles	381
Encrypt communication	381
Delegate security concerns	381
Treat user credentials properly	382
Avoid storing credentials in version control	382
Include tests	382
Possibilities and solutions	383
Encrypted communication	383
Protocol-based authentication	383
Decentralized security	384
Proxies	385
Integration in modern environments	385
Implementing security in Java EE applications	386
Transparent security	386
Servlets	386
Java principals and roles	386
JASPIC	387
Security API	388
Authentication mechanisms	388
Identity stores	388
Custom security	389
Accessing security information	391
Summary	392

Chapter 11: Conclusion	395
Motivations in enterprise development	395
Cloud and Continuous Delivery	396
Relevance of Java EE	396
API updates introduced in Java EE 8	397
CDI 2.0	397
JAX-RS 2.1	398
JSON-B 1.0	399
JSON-P 1.1	400
Bean Validation 2.0	400
JPA 2.2	401
Security 1.0	401
Servlet 4.0	402
JSF 2.3	403
JCP and participation	403
MicroProfile	404
Eclipse Enterprise for Java	404
Appendix: Links and further resources	405
Index	409