

Inhaltsverzeichnis

| | | |
|----------|------------------------------------|----------|
| 1 | Vorwort | 1 |
| 1.1 | Überblick über Microservices | 2 |
| 1.2 | Warum Microservices? | 3 |

Teil I Motivation und Grundlagen 7

| | | |
|----------|--|-----------|
| 2 | Einleitung | 9 |
| 2.1 | Überblick über das Buch | 9 |
| 2.2 | Für wen ist das Buch? | 9 |
| 2.3 | Übersicht über die Kapitel | 10 |
| 2.4 | Essays | 12 |
| 2.5 | Pfade durch das Buch | 12 |
| 2.6 | Danksagung | 13 |
| 2.7 | Links & Literatur | 13 |
| 3 | Microservice-Szenarien | 15 |
| 3.1 | Eine E-Commerce-Legacy-Anwendung modernisieren | 15 |
| 3.2 | Ein neues Signalsystem entwickeln | 23 |
| 3.3 | Fazit | 26 |

Teil II Microservices: Was, warum und warum vielleicht nicht? 29

| | | |
|----------|--|-----------|
| 4 | Was sind Microservices? | 31 |
| 4.1 | Größe eines Microservice | 31 |
| 4.2 | Das Gesetz von Conway | 39 |
| 4.3 | Domain-Driven Design und Bounded Context | 44 |

| | | |
|---|---|------------|
| 4.4 | Microservice: Mit UI? | 54 |
| 4.5 | Fazit | 56 |
| 4.6 | Links & Literatur | 58 |
| 5 | Gründe für Microservices | 59 |
| 5.1 | Technische Vorteile | 59 |
| 5.2 | Organisatorische Vorteile | 67 |
| 5.3 | Vorteile aus Geschäftssicht | 69 |
| 5.4 | Fazit | 71 |
| 5.5 | Links & Literatur | 72 |
| 6 | Herausforderungen bei Microservices | 73 |
| 6.1 | Technische Herausforderungen | 73 |
| 6.2 | Architektur | 77 |
| 6.3 | Infrastruktur und Betrieb | 80 |
| 6.4 | Fazit | 81 |
| 6.5 | Links & Literatur | 82 |
| 7 | Microservices und SOA | 83 |
| 7.1 | Was ist SOA? | 83 |
| 7.2 | Unterschiede zwischen SOA und Microservices | 89 |
| 7.3 | Fazit | 94 |
| 7.4 | Links & Literatur | 96 |
| Teil III Microservices umsetzen | | 97 |
| 8 | Architektur von Microservice-Systemen | 101 |
| 8.1 | Fachliche Architektur | 101 |
| 8.2 | Architekturmanagement | 106 |
| 8.3 | Techniken zum Anpassen der Architektur | 111 |
| 8.4 | Microservice-Systeme weiterentwickeln | 120 |
| 8.5 | Microservice und Legacy-Anwendung | 127 |
| 8.6 | Event-driven Architecture | 137 |
| 8.7 | Technische Architektur | 138 |
| 8.8 | Konfiguration und Koordination | 141 |

| | | |
|-----------|---|------------|
| 8.9 | Service Discovery | 143 |
| 8.10 | Load Balancing | 146 |
| 8.11 | Skalierbarkeit | 150 |
| 8.12 | Sicherheit | 153 |
| 8.13 | Dokumentation und Metadaten | 161 |
| 8.14 | Fazit | 162 |
| 8.15 | Links und Literatur | 165 |
| 9 | Integration und Kommunikation | 167 |
| 9.1 | Web und UI | 167 |
| 9.2 | REST | 179 |
| 9.3 | SOAP und RPC | 182 |
| 9.4 | Messaging | 183 |
| 9.5 | Datenreplikation | 187 |
| 9.6 | Schnittstellen: intern und extern | 190 |
| 9.7 | Fazit | 193 |
| 9.8 | Links & Literatur | 195 |
| 10 | Architektur eines Microservice | 197 |
| 10.1 | Fachliche Architektur | 197 |
| 10.2 | CQRS | 199 |
| 10.3 | Event Sourcing | 201 |
| 10.4 | Hexagonale Architekturen | 203 |
| 10.5 | Resilience und Stabilität | 207 |
| 10.6 | Technische Architektur | 212 |
| 10.7 | Fazit | 215 |
| 10.8 | Links & Literatur | 216 |
| 11 | Testen von Microservices und Microservice-Systemen | 217 |
| 11.1 | Warum testen? | 217 |
| 11.2 | Wie testen? | 219 |
| 11.3 | Risiken beim Deployment minimieren | 224 |
| 11.4 | Tests des Gesamtsystems | 225 |
| 11.5 | Legacy-Anwendungen mit Microservices testen | 228 |
| 11.6 | Tests einzelner Microservices | 231 |
| 11.7 | Consumer-Driven Contract Test | 233 |

| | | |
|----------------------------------|--|------------|
| 11.8 | Technische Standards testen | 237 |
| 11.9 | Fazit | 239 |
| 11.10 | Links & Literatur | 240 |
| 12 | Betrieb und Continuous Delivery von Microservices | 241 |
| 12.1 | Herausforderungen beim Betrieb von Microservices | 241 |
| 12.2 | Logging | 244 |
| 12.3 | Monitoring | 249 |
| 12.4 | Deployment | 256 |
| 12.5 | Steuerung | 262 |
| 12.6 | Infrastrukturen | 263 |
| 12.7 | Fazit | 269 |
| 12.8 | Link & Literatur | 271 |
| 13 | Organisatorische Auswirkungen der Architektur | 273 |
| 13.1 | Organisatorische Vorteile von Microservices | 273 |
| 13.2 | Alternativer Umgang mit dem Gesetz von Conway | 277 |
| 13.3 | Spielräume schaffen: Mikro- und Makro-Architektur | 281 |
| 13.4 | Technische Führung | 287 |
| 13.5 | DevOps | 288 |
| 13.6 | Schnittstelle zu den Fachbereichen | 293 |
| 13.7 | Wiederverwendbarer Code | 295 |
| 13.8 | Microservices ohne Organisationsänderung? | 299 |
| 13.9 | Fazit | 301 |
| 13.10 | Links & Literatur | 303 |
| Teil IV Technologien | | 305 |
| 14 | Ein Beispiel für eine Microservices-Architektur | 307 |
| 14.1 | Fachliche Architektur | 308 |
| 14.2 | Basistechnologien | 310 |
| 14.3 | Build | 314 |
| 14.4 | Deployment mit Docker | 316 |
| 14.5 | Vagrant | 317 |
| 14.6 | Docker Machine | 322 |
| 14.7 | Docker Compose | 324 |

| | | |
|-----------|--|------------|
| 14.8 | Service Discovery | 326 |
| 14.9 | Kommunikation | 329 |
| 14.10 | Resilience | 331 |
| 14.11 | Load Balancing | 335 |
| 14.12 | Integration anderer Technologien | 336 |
| 14.13 | Tests | 337 |
| 14.14 | Fazit | 342 |
| 14.15 | Links & Literatur | 343 |
| 15 | Technologien für Nanoservices | 345 |
| 15.1 | Warum Nanoservices? | 346 |
| 15.2 | Definition Nanoservice | 348 |
| 15.3 | Amazon Lambda | 349 |
| 15.4 | OSGi | 352 |
| 15.5 | Java EE | 355 |
| 15.6 | Vert.x | 358 |
| 15.7 | Erlang | 360 |
| 15.8 | Seneca | 363 |
| 15.9 | Fazit | 366 |
| 15.10 | Links und Literatur | 368 |
| 16 | Wie mit Microservices loslegen? | 369 |
| 16.1 | Warum Microservices? | 369 |
| 16.2 | Wege zu Microservices | 370 |
| 16.3 | Microservice: Hype oder Realität? | 371 |
| 16.4 | Fazit | 372 |
| | Index | 373 |