
Table of Contents

Preface.....	xiii
1. Microservices.....	1
What Are Microservices?	2
Small, and Focused on Doing One Thing Well	2
Autonomous	3
Key Benefits	4
Technology Heterogeneity	4
Resilience	5
Scaling	5
Ease of Deployment	6
Organizational Alignment	7
Composability	7
Optimizing for Replaceability	7
What About Service-Oriented Architecture?	8
Other Decompositional Techniques	9
Shared Libraries	9
Modules	10
No Silver Bullet	11
Summary	11
2. The Evolutionary Architect.....	13
Inaccurate Comparisons	13
An Evolutionary Vision for the Architect	15
Zoning	16
A Principled Approach	17
Strategic Goals	18
Principles	18

Practices	19
Combining Principles and Practices	19
A Real-World Example	19
The Required Standard	20
Monitoring	21
Interfaces	21
Architectural Safety	21
Governance Through Code	22
Exemplars	22
Tailored Service Template	22
Technical Debt	24
Exception Handling	24
Governance and Leading from the Center	25
Building a Team	26
Summary	27
3. How to Model Services.....	29
Introducing MusicCorp	29
What Makes a Good Service?	30
Loose Coupling	30
High Cohesion	30
The Bounded Context	31
Shared and Hidden Models	31
Modules and Services	33
Premature Decomposition	33
Business Capabilities	34
Turtles All the Way Down	34
Communication in Terms of Business Concepts	36
The Technical Boundary	36
Summary	37
4. Integration.....	39
Looking for the Ideal Integration Technology	39
Avoid Breaking Changes	39
Keep Your APIs Technology-Agnostic	39
Make Your Service Simple for Consumers	40
Hide Internal Implementation Detail	40
Interfacing with Customers	40
The Shared Database	41
Synchronous Versus Asynchronous	42
Orchestration Versus Choreography	43
Remote Procedure Calls	46

Technology Coupling	47
Local Calls Are Not Like Remote Calls	47
Brittleness	47
Is RPC Terrible?	49
REST	49
REST and HTTP	50
Hypermedia As the Engine of Application State	51
JSON, XML, or Something Else?	53
Beware Too Much Convenience	54
Downsides to REST Over HTTP	54
Implementing Asynchronous Event-Based Collaboration	55
Technology Choices	55
Complexities of Asynchronous Architectures	57
Services as State Machines	58
Reactive Extensions	58
DRY and the Perils of Code Reuse in a Microservice World	59
Client Libraries	59
Access by Reference	60
Versioning	62
Defer It for as Long as Possible	62
Catch Breaking Changes Early	63
Use Semantic Versioning	64
Coexist Different Endpoints	64
Use Multiple Concurrent Service Versions	66
User Interfaces	67
Toward Digital	67
Constraints	68
API Composition	68
UI Fragment Composition	69
Backends for Frontends	71
A Hybrid Approach	73
Integrating with Third-Party Software	73
Lack of Control	74
Customization	74
Integration Spaghetti	74
On Your Own Terms	75
The Strangler Pattern	77
Summary	78
5. Splitting the Monolith.....	79
It's All About Seams	79
Breaking Apart MusicCorp	80

The Reasons to Split the Monolith	81
Pace of Change	81
Team Structure	81
Security	81
Technology	82
Tangled Dependencies	82
The Database	82
Getting to Grips with the Problem	82
Example: Breaking Foreign Key Relationships	84
Example: Shared Static Data	85
Example: Shared Data	86
Example: Shared Tables	87
Refactoring Databases	88
Staging the Break	89
Transactional Boundaries	89
Try Again Later	91
Abort the Entire Operation	91
Distributed Transactions	92
So What to Do?	93
Reporting	93
The Reporting Database	93
Data Retrieval via Service Calls	95
Data Pumps	96
Alternative Destinations	98
Event Data Pump	98
Backup Data Pump	99
Toward Real Time	100
Cost of Change	100
Understanding Root Causes	101
Summary	102
6. Deployment.....	103
A Brief Introduction to Continuous Integration	103
Are You Really Doing It?	104
Mapping Continuous Integration to Microservices	105
Build Pipelines and Continuous Delivery	107
And the Inevitable Exceptions	108
Platform-Specific Artifacts	109
Operating System Artifacts	110
Custom Images	111
Images as Artifacts	113
Immutable Servers	113

Environments	114
Service Configuration	115
Service-to-Host Mapping	116
Multiple Services Per Host	116
Application Containers	118
Single Service Per Host	119
Platform as a Service	120
Automation	121
Two Case Studies on the Power of Automation	122
From Physical to Virtual	122
Traditional Virtualization	123
Vagrant	124
Linux Containers	124
Docker	126
A Deployment Interface	127
Environment Definition	128
Summary	130
7. Testing.....	131
Types of Tests	131
Test Scope	132
Unit Tests	134
Service Tests	134
End-to-End Tests	135
Trade-Offs	135
How Many?	136
Implementing Service Tests	136
Mocking or Stubbing	137
A Smarter Stub Service	138
Those Tricky End-to-End Tests	138
Downsides to End-to-End Testing	140
Flaky and Brittle Tests	140
Who Writes These Tests?	141
How Long?	141
The Great Pile-up	142
The Metaversion	143
Test Journeys, Not Stories	143
Consumer-Driven Tests to the Rescue	144
Pact	145
It's About Conversations	147
So Should You Use End-to-End Tests?	147
Testing After Production	148

Separating Deployment from Release	148
Canary Releasing	149
Mean Time to Repair Over Mean Time Between Failures?	150
Cross-Functional Testing	151
Performance Tests	152
Summary	153
8. Monitoring.....	155
Single Service, Single Server	156
Single Service, Multiple Servers	156
Multiple Services, Multiple Servers	157
Logs, Logs, and Yet More Logs...	158
Metric Tracking Across Multiple Services	159
Service Metrics	160
Synthetic Monitoring	161
Implementing Semantic Monitoring	162
Correlation IDs	162
The Cascade	164
Standardization	165
Consider the Audience	165
The Future	166
Summary	167
9. Security.....	169
Authentication and Authorization	169
Common Single Sign-On Implementations	170
Single Sign-On Gateway	171
Fine-Grained Authorization	172
Service-to-Service Authentication and Authorization	173
Allow Everything Inside the Perimeter	173
HTTP(S) Basic Authentication	174
Use SAML or OpenID Connect	174
Client Certificates	175
HMAC Over HTTP	176
API Keys	177
The Deputy Problem	178
Securing Data at Rest	180
Go with the Well Known	180
It's All About the Keys	181
Pick Your Targets	181
Decrypt on Demand	181
Encrypt Backups	181

Defense in Depth	182
Firewalls	182
Logging	182
Intrusion Detection (and Prevention) System	182
Network Segregation	183
Operating System	183
A Worked Example	184
Be Frugal	186
The Human Element	187
The Golden Rule	187
Baking Security In	187
External Verification	188
Summary	188
10. Conway's Law and System Design.....	191
Evidence	191
Loose and Tightly Coupled Organizations	192
Windows Vista	192
Netflix and Amazon	192
What Can We Do with This?	193
Adapting to Communication Pathways	193
Service Ownership	194
Drivers for Shared Services	195
Too Hard to Split	195
Feature Teams	195
Delivery Bottlenecks	196
Internal Open Source	196
Role of the Custodians	197
Maturity	197
Tooling	198
Bounded Contexts and Team Structures	198
The Orphaned Service?	198
Case Study: RealEstate.com.au	199
Conway's Law in Reverse	201
People	202
Summary	203
11. Microservices at Scale.....	205
Failure Is Everywhere	205
How Much Is Too Much?	206
Degraded Functionality	207
Architectural Safety Measures	208

The Antifragile Organization	210
Timeouts	211
Circuit Breakers	212
Bulkheads	214
Isolation	215
Idempotency	215
Scaling	216
Go Bigger	217
Splitting Workloads	217
Spreading Your Risk	217
Load Balancing	219
Worker-Based Systems	220
Starting Again	221
Scaling Databases	222
Availability of Service Versus Durability of Data	222
Scaling for Reads	222
Scaling for Writes	223
Shared Database Infrastructure	224
CQRS	224
Caching	225
Client-Side, Proxy, and Server-Side Caching	226
Caching in HTTP	226
Caching for Writes	228
Caching for Resilience	228
Hiding the Origin	228
Keep It Simple	229
Cache Poisoning: A Cautionary Tale	230
Autoscaling	230
CAP Theorem	232
Sacrificing Consistency	233
Sacrificing Availability	233
Sacrificing Partition Tolerance?	234
AP or CP?	235
It's Not All or Nothing	235
And the Real World	236
Service Discovery	236
DNS	237
Dynamic Service Registries	238
Zookeeper	238
Consul	239
Eureka	240
Rolling Your Own	241

Don't Forget the Humans!	241
Documenting Services	241
Swagger	242
HAL and the HAL Browser	242
The Self-Describing System	243
Summary	243
12. Bringing It All Together.....	245
Principles of Microservices	245
Model Around Business Concepts	246
Adopt a Culture of Automation	246
Hide Internal Implementation Details	247
Decentralize All the Things	247
Independently Deployable	248
Isolate Failure	248
Highly Observable	249
When Shouldn't You Use Microservices?	249
Parting Words	250
Index.....	251