

Contents

1	Introduction to Software Leadership	1
	Role of Judgment	1
	Goal of This Book	3
	Part I: Introduction	6
	Part II: Essential Background	7
	Part III: System Design	7
	Part IV: Putting Everything Together	8
2	Understanding Systems, Design, and Architecture	9
	What Is Software Architecture?	9
	How to Design a System	11
	Five Questions	12
	Question 1: When Is the Best Time to Market?	12
	Question 2: What Is the Skill Level of the Team?	13
	Question 3: What Is Our System's Performance Sensitivity?	14
	Question 4: When Can We Rewrite the System?	15
	Question 5: What Are the Hard Problems?	15
	Seven Principles: The Overarching Concepts	16
	Principle 1: Drive Everything from the User's Journey	16
	Principle 2: Use an Iterative Thin Slice Strategy	17
	Principle 3: On Each Iteration, Add the Most Value for the Least Effort to Support More Users	18
	Principle 4: Make Decisions and Absorb the Risks	20
	Principle 5: Design Deeply Things That Are Hard to Change but Implement Them Slowly	20
	Principle 6: Eliminate the Unknowns and Learn from the Evidence by Working on Hard Problems Early and in Parallel	22
	Principle 7: Understand the Trade-offs Between Cohesion and Flexibility in the Software Architecture	23

Designing for an Online Bookstore	24
Designing for the Cloud	27
Summary	29
Mental Models for Understanding and Explaining System Performance	31
A Computer System	32
Models for Performance	33
Model 1: Cost of Switching to the Kernel Mode from the User Mode	34
Model 2: Operations Hierarchy	34
Model 3: Context Switching Overhead	35
Model 4: Amdahl's Law	35
Model 5: Universal Scalability Law (USL)	36
Model 6: Latency and Utilization Trade-offs	37
Model 7: Designing for Throughput with the Maximal Useful Utilization (MUU) Model	37
Model 8: Adding Latency Limits	39
Optimization Techniques	41
CPU Optimization Techniques	42
I/O Optimization Techniques	43
Memory Optimization Techniques	44
Latency Optimization Techniques	45
Intuitive Feel for Performance	46
Leadership Considerations	46
Summary	47
Understanding User Experience (UX)	49
General UX Concepts for Architects	49
Principle 1: Understand the Users	50
Principle 2: Do as Little as Possible	50
Principle 3: Good Products Do Not Need a Manual: Its Use Is Self-Evident.	51
Principle 4: Think in Terms of Information Exchange	51

Principle 5: Make Simple Things Simple	52
Principle 6: Design UX Before Implementation	52
UX Design for Configurations	53
UX Design for APIs	54
UX Design for Extensions	56
Leadership Considerations	57
Summary	57
5 Macro Architecture: Introduction	59
History of Macro Architecture	60
Modern Architectures	62
Macro Architectural Building Blocks	63
Leadership Considerations	66
Summary	68
6 Macro Architecture: Coordination	69
Approach 1: Drive Flow from Client	69
Approach 2: Use Another Service	70
Approach 3: Use Centralized Middleware	71
Approach 4: Implement Choreography	71
Leadership Considerations	73
Summary	73
7 Macro Architecture: Preserving Consistency of State	75
Why Transactions?	75
Why Do We Need to Go Beyond Transactions?	76
Going Beyond Transactions	77
Approach 1: Redefining the Problem to Require Lesser Guarantees	78
Approach 2: Using Compensations	78
Best Practices	80
Leadership Considerations	81
Summary	83

8	Macro Architecture: Handling Security	85	10	Macro Architecture: Microservices Considerations	119
	User Management	86		Decision 1: Handling Shared Database(s)	120
	Interaction Security	88		Solution 1: One Microservice Updating the Database	121
	Authentication Techniques	89		Solution 2: Two Microservices Updating the Database	122
	Authorization Techniques	90		Decision 2: Securing Microservices	122
	Common Interaction Security Scenarios for an App	93		Decision 3: Coordinating Microservices	122
	Storage, GDPR, and Other Regulations	96		Decision 4: Avoiding Dependency Hell	122
	Security Strategy and Advice	98		Backward Compatibility	123
	Performance and Latency	99		Forward Compatibility	123
	Zero-Trust Approach	99		Dependency Graphs	124
	Take Caution When Running User-Provided Code	100		Loosely Coupled, Repository-Based Teams as an Alternative to Microservices	125
	Blockchain Topics	100		Leadership Considerations	126
	Other Topics	100		Summary	127
	Leadership Considerations	101	11	Server Architectures	129
	Summary	103		Writing a Service	129
9	Macro Architecture: Handling High Availability and Scale	105		Understanding Best Practices for Writing a Service	130
	Adding High Availability	105		Understanding Advanced Techniques	132
	Replication	105		Using Alternative I/O and Thread Models	132
	Fast Recovery	107		Understanding Coordination Overhead	138
	Understanding Scalability	109		Efficiently Saving Local State	139
	Scaling for a Modern Architecture: Base Solution	110		Choosing a Transport System	140
	Scaling: The Tools of Trade	111		Handling Latency	140
	Scale Tactic 1: Share Nothing	112		Separating Reads and Writes	141
	Scale Tactic 2: Distribution	112		Using Locks (and Signaling) in Applications	141
	Scale Tactic 3: Caching	112		Using Queues and Pools	142
	Scale Tactic 4: Async Processing	113		Handling Service Calls	143
	Building Scalable Systems	113		Using These Techniques in Practice	143
	Approach 1: Successive Bottleneck Elimination	114		CPU-Bound Applications (CPU >> Memory and No I/O)	144
	Approach 2: Shared-Nothing Design	115		Memory-Bound Applications (CPU + Bound Memory and No I/O)	144
	Leadership Considerations	117		Balanced Applications (CPU + Memory + I/O)	144
	Summary	118		I/O-Bound Applications (I/O + Memory > CPU)	145

Other Application Categorizations	145
Leadership Considerations	146
Summary	147
12 Building Stable Systems	149
Why Do Systems Fail, and What Can We Do About Them?	149
How to Handle Known Errors	151
Handling Unexpected Load	151
Handling Resource Failures	154
Handling Dependencies	157
Handling Human Changes	158
Common Bugs	159
Resource Leaks	159
Deadlocks and Slow Operations	160
How to Handle Unknown Errors	161
Observability	161
Bugs and Testing	161
Graceful Degradation	163
Leadership Considerations	163
Summary	164
13 Building and Evolving the Systems	165
Getting Your Hands Dirty	165
Get the Basics Right	165
Understand the Design Process	167
Make Decisions and Absorb Risks	169
Demand Excellence	170
Communicating the Design	172
Evolving the System: How to Learn from Your Users and Improve the System	172
Leadership Considerations	175
Summary	176
Index	179