

Table of contents provided by Syndetics

- **Preface** (p. vii)
- **1 Introduction** (p. 1)
- **A Brief History of Systems Development** (p. 1)
- **The Value of Patterns, Practices, and Components** (p. 4)
- **Standing on the Shoulders of Giants** (p. 4)
- **A Shared Language for Discussing Our Practice** (p. 5)
- **Summary** (p. 8)
- **An Example Sidecar: Adding HTTPS to a Legacy Service** (p. 11)
- **Hands On: Deploying the topz Container** (p. 14)
- **Building a Simple PaaS with Sidecars** (p. 15)
- **Parameterized Containers** (p. 17)
- **Define Each Container's API** (p. 17)
- **Summary** (p. 19)
- **A Brief History of Patterns in Software Development** (p. 2)
- **The Rise of Open Source Software** (p. 3)
- **Formalization of Algorithmic Programming** (p. 3)
- **Summary** (p. 6)
- **2 The Sidecar Pattern** (p. 11)
- **Modular Application Containers** (p. 14)
- **Patterns for Object-Oriented Programming** (p. 3)
- **Designing Sidecars for Modularity and Reusability** (p. 16)
- **Shared Components for Easy Reuse** (p. 5)
- **Part I Single-Node Patterns**
- **Documenting Your Containers** (p. 18)
- **Motivations** (p. 7)
- **Dynamic Configuration with Sidecars** (p. 12)
- **3 Ambassadors** (p. 21)
- **Using an Ambassador to Shard a Service** (p. 22)
- **Hands On: Implementing a Sharded Redis** (p. 23)
- **Using an Ambassador for Service Brokering** (p. 25)
- **Using an Ambassador to Do Experimentation or Request Splitting** (p. 26)
- **Hands On: Implementing 10% Experiments** (p. 27)
- **4 Adapters** (p. 31)
- **Monitoring** (p. 32)
- **Hands On: Using Prometheus for Monitoring** (p. 33)
- **Logging** (p. 34)
- **Hands On: Normalizing Different Logging Formats with Fluentd** (p. 35)
- **Adding a Health Monitor** (p. 36)
- **Hands On: Adding Rich Health Monitoring for MySQL** (p. 37)
- **Part II Serving Patterns**
- **Introduction to Microservices** (p. 41)
- **5 Replicated Load-Balanced Services** (p. 45)
- **Stateless Services** (p. 45)
- **Readiness Probes for Load Balancing** (p. 46)

- **Hands On: Creating a Replicated Service in Kubernetes** (p. 47)
- **Session Tracked Services** (p. 48)
- **Application-Layer Replicated Services** (p. 49)
- **Introducing a Caching Layer** (p. 49)
- **Deploying Your Cache** (p. 50)
- **Hands On: Deploying the Caching Layer** (p. 51)
- **Expanding the Caching Layer** (p. 53)
- **Rate Limiting and Denial-of-Service Defense** (p. 54)
- **SSL Termination** (p. 54)
- **Hands On: Deploying nginx and SSL Termination** (p. 55)
- **Summary** (p. 57)
- **6 Sharded Services** (p. 59)
- **Sharded Caching** (p. 59)
- **Why You Might Need a Sharded Cache** (p. 60)
- **The Role of the Cache in System Performance** (p. 61)
- **Replicated, Sharded Caches** (p. 62)
- **Hands On: Deploying an Ambassador and Memcache for a Sharded Cache** (p. 63)
- **An Examination of Sharding Functions** (p. 66)
- **Selecting a Key** (p. 67)
- **Consistent Hashing Functions** (p. 68)
- **Hands On: Building a Consistent HTTP Sharding Proxy** (p. 69)
- **Hot Sharding Systems** (p. 70)
- **7 Scatter/Gather** (p. 73)
- **Scatter/Gather with Root Distribution** (p. 74)
- **Hands On: Distributed Document Search** (p. 75)
- **Scatter/Gather with Leaf Sharding** (p. 76)
- **Scaling Scatter/Gather for Reliability and Scale** (p. 79)
- **8 Functions and Event-Driven Processing** (p. 81)
- **Determining When FaaS Makes Sense** (p. 82)
- **The Benefits of FaaS** (p. 82)
- **The Challenges of FaaS** (p. 82)
- **The Need for Background Processing** (p. 83)
- **The Need to Hold Data in Memory** (p. 83)
- **The Costs of Sustained Request-Based Processing** (p. 84)
- **Sharded, Replicated Serving** (p. 70)
- **Hands On: Sharded Document Search** (p. 77)
- **Choosing the Right Number of Leaves** (p. 78)
- **Patterns for FaaS** (p. 84)
- **The Decorator Pattern: Request or Response Transformation** (p. 85)
- **Hands On: Adding Request Defaulting Prior to Request Processing** (p. 86)
- **Handling Events** (p. 87)
- **Hands On: Implementing Two-Factor Authentication** (p. 88)
- **Event-Based Pipelines** (p. 89)
- **Hands On: Implementing a Pipeline for New-User Signup** (p. 89)
- **9 Ownership Election** (p. 93)
- **Determining If You Even Need Master Election** (p. 94)

- **The Basics of Master Election** (p. 95)
- **Hands On: Deploying etcd** (p. 97)
- **Implementing Locks** (p. 98)
- **Hands On: Implementing Locks in etcd** (p. 101)
- **Implementing Ownership** (p. 101)
- **Hands On: Implementing Leases in etcd** (p. 102)
- **Handling Concurrent Data Manipulation** (p. 103)
- **Part III Batch Computational Patterns**
- **10 Work Queue Systems** (p. 109)
- **A Generic Work Queue System** (p. 109)
- **The Source Container Interface** (p. 110)
- **The Worker Container Interface** (p. 112)
- **The Shared Work Queue Infrastructure** (p. 113)
- **Hands On: Implementing a Video Thumbnailer** (p. 115)
- **Dynamic Scaling of the Workers** (p. 117)
- **The Multi-Worker Pattern** (p. 118)
- **11 Event-Driven Batch Processing** (p. 121)
- **Patterns of Event-Driven Processing** (p. 122)
- **Copier** (p. 122)
- **Filter** (p. 123)
- **Splitter** (p. 124)
- **Sharder** (p. 125)
- **Merger** (p. 127)
- **Hands On: Building an Event-Driven Flow for New User Sign-Up** (p. 128)
- **Publisher/Subscriber Infrastructure** (p. 129)
- **Hands On: Deploying Kafka** (p. 130)
- **12 Coordinated Batch Processing** (p. 133)
- **Join (or Barrier Synchronization)** (p. 134)
- **Reduce** (p. 135)
- **Hands On: Count** (p. 136)
- **Sum** (p. 137)
- **Histogram** (p. 137)
- **Hands On: An Image Tagging and Processing Pipeline** (p. 138)
- **13 Conclusion: A New Beginning?** (p. 143)
- **Index** (p. 145)