Table of contents

Preface
Conventions Used in This Book
Using Code Examples
O'Reilly Online Learning
How to Contact Us
Acknowledgments
Acknowledgments from Mark Richards
Acknowledgments from Neal Ford
Acknowledgments from Pramod Sadalage
Acknowledgments from Zhamak Dehghani
1. What Happens When There Are No "Best Practices"?
Why "The Hard Parts"?
Giving Timeless Advice About Software Architecture
The Importance of Data in Architecture
Architectural Decision Records
Architecture Fitness Functions
Using Fitness Functions
Architecture Versus Design: Keeping Definitions Simple
Introducing the Sysops Squad Saga
Nonticketing Workflow
Ticketing Workflow
A Bad Scenario
Sysops Squad Architectural Components
Sysops Squad Data Model
I. Pulling Things Apart
2. Discerning Coupling in Software Architecture
Architecture (Quantum | Quanta)
Independently Deployable
High Functional Cohesion
High Static Coupling
Dynamic Quantum Coupling
Sysops Squad Saga: Understanding Quanta
3. Architectural Modularity
Modularity Drivers
Maintainability
Testability
Deployability
Scalability
Availability/Fault Tolerance
Sysops Squad Saga: Creating a Business Case
4. Architectural Decomposition
Is the Codebase Decomposable?
Afferent and Efferent Coupling
Abstractness and Instability
Distance from the Main Sequence
Component-Based Decomposition
Tactical Forking

Trade-Offs
Sysops Squad Saga: Choosing a Decomposition Approach
5. Component-Based Decomposition Patterns
Identify and Size Components Pattern
Pattern Description
Fitness Functions for Governance
Sysops Squad Saga: Sizing Components
Gather Common Domain Components Pattern
Pattern Description
Fitness Functions for Governance
Sysops Squad Saga: Gathering Common Components
Flatten Components Pattern
Pattern Description
Fitness Functions for Governance
Sysops Squad Saga: Flattening Components
Determine Component Dependencies Pattern
Pattern Description
Fitness Functions for Governance
Sysops Squad Saga: Identifying Component Dependencies
Create Component Domains Pattern
Pattern Description
Fitness Functions for Governance
Sysops Squad Saga: Creating Component Domains
Create Domain Services Pattern
Pattern Description
Fitness Functions for Governance
Sysops Squad Saga: Creating Domain Services
Summary
6. Pulling Apart Operational Data
Data Decomposition Drivers
Data Disintegrators
Data Integrators
Sysops Squad Saga: Justifying Database Decomposition
Decomposing Monolithic Data
Step 1: Analyze Database and Create Data Domains
Step 2: Assign Tables to Data Domains
Step 3: Separate Database Connections to Data Domains
Step 4: Move Schemas to Separate Database Servers
Step 5: Switch Over to Independent Database Servers
Selecting a Database Type
Relational Databases
Key-Value Databases
Document Databases
Column Family Databases
Graph Databases
NewSQL Databases
Cloud Native Databases
Time-Series Databases

Sysops Squad Saga: Polyglot Databases
7. Service Granularity
Granularity Disintegrators
Service Scope and Function
Code Volatility
Scalability and Throughput
Fault Tolerance
Security
Extensibility
Granularity Integrators
Database Transactions
Workflow and Choreography
Shared Code
Data Relationships
Finding the Right Balance
Sysops Squad Saga: Ticket Assignment Granularity
Sysops Squad Saga: Customer Registration Granularity
II. Putting Things Back Together
8. Reuse Patterns
Code Replication
When to Use
Shared Library
Dependency Management and Change Control
Versioning Strategies
When To Use
Shared Service
Change Risk
Performance
Scalability
Fault Tolerance
When to Use
Sidecars and Service Mesh
When to Use
Sysops Squad Saga: Common Infrastructure Logic
Code Reuse: When Does It Add Value?
Reuse via Platforms
Sysops Squad Saga: Shared Domain Functionality
9. Data Ownership and Distributed Transactions
Assigning Data Ownership
Single Ownership Scenario
Common Ownership Scenario
Joint Ownership Scenario
Table Split Technique
Data Domain Technique
Delegate Technique
Service Consolidation Technique
Data Ownership Summary
Distributed Transactions

Eventual Consistency Patterns
Background Synchronization Pattern
Orchestrated Request-Based Pattern
Event-Based Pattern
Sysops Squad Saga: Data Ownership for Ticket Processing
10. Distributed Data Access
Interservice Communication Pattern
Column Schema Replication Pattern
Replicated Caching Pattern
Data Domain Pattern
Sysops Squad Saga: Data Access for Ticket Assignment
11. Managing Distributed Workflows
Orchestration Communication Style
Choreography Communication Style
Workflow State Management
Trade-Offs Between Orchestration and Choreography
State Owner and Coupling
Sysops Squad Saga: Managing Workflows
12. Transactional Sagas
Transactional Saga Patterns
Epic Saga(sao) Pattern
Phone Tag Saga(sac) Pattern
Fairy Tale Saga(seo) Pattern
Time Travel Saga(sec) Pattern
Fantasy Fiction Saga(aao) Pattern
Horror Story(aac) Pattern
Parallel Saga(aeo) Pattern
Anthology Saga(aec) Pattern
State Management and Eventual Consistency
Saga State Machines
Techniques for Managing Sagas
Sysops Squad Saga: Atomic Transactions and Compensating Updates
13. Contracts
Strict Versus Loose Contracts
Trade-Offs Between Strict and Loose Contracts
Contracts in Microservices
Stamp Coupling
Over-Coupling via Stamp Coupling
Bandwidth
Stamp Coupling for Workflow Management
Sysops Squad Saga: Managing Ticketing Contracts
14. Managing Analytical Data
Previous Approaches
The Data Warehouse
The Data Lake
The Data Mesh
Definition of Data Mesh
Data Product Quantum

Data Mesh, Coupling, and Architecture Quantum
When to Use Data Mesh
Sysops Squad Saga: Data Mesh
15. Build Your Own Trade-Off Analysis
Finding Entangled Dimensions
Coupling
Analyze Coupling Points
Assess Trade-Offs
Trade-Off Techniques
Qualitative Versus Quantative Analysis
MECE Lists
The “Out-of-Context” Trap
Model Relevant Domain Cases
Prefer Bottom Line over Overwhelming Evidence
Avoiding Snake Oil and Evangelism
Sysops Squad Saga: Epilogue
A. Concept and Term References
B. Architecture Decision Record References
C. Trade-Off References
Index