

Table of contents

1. [Preface](#)
 1. [What This Book Isn't](#)
 2. [What This Book Is About](#)
 3. [Who Should Read This Book](#)
 4. [Prerequisites](#)
 5. [What You'll Learn and How It Will Improve Your Abilities](#)
 6. [Navigating This Book](#)
 7. [Conventions Used in This Book](#)
 8. [How to Contact Us](#)
 9. [Acknowledgments](#)
2. [I. Foundation and Building Blocks](#)
3. [1. Data Engineering Described](#)
 1. [What Is Data Engineering?](#)
 1. [Data Engineering Defined](#)
 2. [The Data Engineering Lifecycle](#)
 3. [Evolution of the Data Engineer](#)
 4. [Data Engineering and Data Science](#)
 2. [Data Engineering Skills and Activities](#)
 1. [Data Maturity and the Data Engineer](#)
 2. [The Background and Skills of a Data Engineer](#)
 3. [Business Responsibilities](#)
 4. [Technical Responsibilities](#)
 5. [The Continuum of Data Engineering Roles, from A to B](#)
 3. [Data Engineers Inside an Organization](#)
 1. [Internal-Facing Versus External-Facing Data Engineers](#)
 2. [Data Engineers and Other Technical Roles](#)
 3. [Data Engineers and Business Leadership](#)
 4. [Conclusion](#)
 5. [Additional Resources](#)
4. [2. The Data Engineering Lifecycle](#)
 1. [What Is the Data Engineering Lifecycle?](#)
 1. [The Data Lifecycle Versus the Data Engineering Lifecycle](#)
 2. [Generation: Source Systems](#)
 3. [Storage](#)
 4. [Ingestion](#)
 5. [Transformation](#)
 6. [Serving Data](#)
 2. [Major Undercurrents Across the Data Engineering Lifecycle](#)
 1. [Security](#)
 2. [Data Management](#)
 3. [DataOps](#)
 4. [Data Architecture](#)
 5. [Orchestration](#)
 6. [Software Engineering](#)

3. [Conclusion](#)
4. [Additional Resources](#)
5. [3. Designing Good Data Architecture](#)
 1. [What Is Data Architecture?](#)
 1. [Enterprise Architecture Defined](#)
 2. [Data Architecture Defined](#)
 3. [“Good” Data Architecture](#)
 2. [Principles of Good Data Architecture](#)
 1. [Principle 1: Choose Common Components Wisely](#)
 2. [Principle 2: Plan for Failure](#)
 3. [Principle 3: Architect for Scalability](#)
 4. [Principle 4: Architecture Is Leadership](#)
 5. [Principle 5: Always Be Architecting](#)
 6. [Principle 6: Build Loosely Coupled Systems](#)
 7. [Principle 7: Make Reversible Decisions](#)
 8. [Principle 8: Prioritize Security](#)
 9. [Principle 9: Embrace FinOps](#)
 3. [Major Architecture Concepts](#)
 1. [Domains and Services](#)
 2. [Distributed Systems, Scalability, and Designing for Failure](#)
 3. [Tight Versus Loose Coupling: Tiers, Monoliths, and Microservices](#)
 4. [User Access: Single Versus Multitenant](#)
 5. [Event-Driven Architecture](#)
 6. [Brownfield Versus Greenfield Projects](#)
 4. [Examples and Types of Data Architecture](#)
 1. [Data Warehouse](#)
 2. [Data Lake](#)
 3. [Convergence, Next-Generation Data Lakes, and the Data Platform](#)
 4. [Modern Data Stack](#)
 5. [Lambda Architecture](#)
 6. [Kappa Architecture](#)
 7. [The Dataflow Model and Unified Batch and Streaming](#)
 8. [Architecture for IoT](#)
 9. [Data Mesh](#)
 10. [Other Data Architecture Examples](#)
 5. [Who’s Involved with Designing a Data Architecture?](#)
 6. [Conclusion](#)
 7. [Additional Resources](#)
6. [4. Choosing Technologies Across the Data Engineering Lifecycle](#)
 1. [Team Size and Capabilities](#)
 2. [Speed to Market](#)
 3. [Interoperability](#)
 4. [Cost Optimization and Business Value](#)
 1. [Total Cost of Ownership](#)
 2. [Total Opportunity Cost of Ownership](#)
 3. [FinOps](#)

5. [Today Versus the Future: Immutable Versus Transitory Technologies](#)
 1. [Our Advice](#)
6. [Location](#)
 1. [On Premises](#)
 2. [Cloud](#)
 3. [Hybrid Cloud](#)
 4. [Multicloud](#)
 5. [Decentralized: Blockchain and the Edge](#)
 6. [Our Advice](#)
 7. [Cloud Repatriation Arguments](#)
7. [Build Versus Buy](#)
 1. [Open Source Software](#)
 2. [Proprietary Walled Gardens](#)
 3. [Our Advice](#)
8. [Monolith Versus Modular](#)
 1. [Monolith](#)
 2. [Modularity](#)
 3. [The Distributed Monolith Pattern](#)
 4. [Our Advice](#)
9. [Serverless Versus Servers](#)
 1. [Serverless](#)
 2. [Containers](#)
 3. [How to Evaluate Server Versus Serverless](#)
 4. [Our Advice](#)
10. [Optimization, Performance, and the Benchmark Wars](#)
 1. [Big Data...for the 1990s](#)
 2. [Nonsensical Cost Comparisons](#)
 3. [Asymmetric Optimization](#)
 4. [Caveat Emptor](#)
11. [Undercurrents and Their Impacts on Choosing Technologies](#)
 1. [Data Management](#)
 2. [DataOps](#)
 3. [Data Architecture](#)
 4. [Orchestration Example: Airflow](#)
 5. [Software Engineering](#)
12. [Conclusion](#)
13. [Additional Resources](#)
7. [II. The Data Engineering Lifecycle in Depth](#)
8. [5. Data Generation in Source Systems](#)
 1. [Sources of Data: How Is Data Created?](#)
 2. [Source Systems: Main Ideas](#)
 1. [Files and Unstructured Data](#)
 2. [APIs](#)
 3. [Application Databases \(OLTP Systems\)](#)
 4. [Online Analytical Processing System](#)
 5. [Change Data Capture](#)

6. [Logs](#)
7. [Database Logs](#)
8. [CRUD](#)
9. [Insert-Only](#)
10. [Messages and Streams](#)
11. [Types of Time](#)
3. [Source System Practical Details](#)
 1. [Databases](#)
 2. [APIs](#)
 3. [Data Sharing](#)
 4. [Third-Party Data Sources](#)
 5. [Message Queues and Event-Streaming Platforms](#)
4. [Whom You'll Work With](#)
5. [Undercurrents and Their Impact on Source Systems](#)
 1. [Security](#)
 2. [Data Management](#)
 3. [DataOps](#)
 4. [Data Architecture](#)
 5. [Orchestration](#)
 6. [Software Engineering](#)
6. [Conclusion](#)
7. [Additional Resources](#)
9. [6. Storage](#)
 1. [Raw Ingredients of Data Storage](#)
 1. [Magnetic Disk Drive](#)
 2. [Solid-State Drive](#)
 3. [Random Access Memory](#)
 4. [Networking and CPU](#)
 5. [Serialization](#)
 6. [Compression](#)
 7. [Caching](#)
 2. [Data Storage Systems](#)
 1. [Single Machine Versus Distributed Storage](#)
 2. [Eventual Versus Strong Consistency](#)
 3. [File Storage](#)
 4. [Block Storage](#)
 5. [Object Storage](#)
 6. [Cache and Memory-Based Storage Systems](#)
 7. [The Hadoop Distributed File System](#)
 8. [Streaming Storage](#)
 9. [Indexes, Partitioning, and Clustering](#)
 3. [Data Engineering Storage Abstractions](#)
 1. [The Data Warehouse](#)
 2. [The Data Lake](#)
 3. [The Data Lakehouse](#)
 4. [Data Platforms](#)

5. [Stream-to-Batch Storage Architecture](#)
 4. [Big Ideas and Trends in Storage](#)
 1. [Data Catalog](#)
 2. [Data Sharing](#)
 3. [Schema](#)
 4. [Separation of Compute from Storage](#)
 5. [Data Storage Lifecycle and Data Retention](#)
 6. [Single-Tenant Versus Multitenant Storage](#)
 5. [Whom You'll Work With](#)
 6. [Undercurrents](#)
 1. [Security](#)
 2. [Data Management](#)
 3. [DataOps](#)
 4. [Data Architecture](#)
 5. [Orchestration](#)
 6. [Software Engineering](#)
 7. [Conclusion](#)
 8. [Additional Resources](#)
10. [7. Ingestion](#)
 1. [What Is Data Ingestion?](#)
 2. [Key Engineering Considerations for the Ingestion Phase](#)
 1. [Bounded Versus Unbounded Data](#)
 2. [Frequency](#)
 3. [Synchronous Versus Asynchronous Ingestion](#)
 4. [Serialization and Deserialization](#)
 5. [Throughput and Scalability](#)
 6. [Reliability and Durability](#)
 7. [Payload](#)
 8. [Push Versus Pull Versus Poll Patterns](#)
 3. [Batch Ingestion Considerations](#)
 1. [Snapshot or Differential Extraction](#)
 2. [File-Based Export and Ingestion](#)
 3. [ETL Versus ELT](#)
 4. [Inserts, Updates, and Batch Size](#)
 5. [Data Migration](#)
 4. [Message and Stream Ingestion Considerations](#)
 1. [Schema Evolution](#)
 2. [Late-Arriving Data](#)
 3. [Ordering and Multiple Delivery](#)
 4. [Replay](#)
 5. [Time to Live](#)
 6. [Message Size](#)
 7. [Error Handling and Dead-Letter Queues](#)
 8. [Consumer Pull and Push](#)
 9. [Location](#)
 5. [Ways to Ingest Data](#)

1. [Direct Database Connection](#)
2. [Change Data Capture](#)
3. [APIs](#)
4. [Message Queues and Event-Streaming Platforms](#)
5. [Managed Data Connectors](#)
6. [Moving Data with Object Storage](#)
7. [EDI](#)
8. [Databases and File Export](#)
9. [Practical Issues with Common File Formats](#)
10. [Shell](#)
11. [SSH](#)
12. [SFTP and SCP](#)
13. [Webhooks](#)
14. [Web Interface](#)
15. [Web Scraping](#)
16. [Transfer Appliances for Data Migration](#)
17. [Data Sharing](#)
6. [Whom You'll Work With](#)
 1. [Upstream Stakeholders](#)
 2. [Downstream Stakeholders](#)
7. [Undercurrents](#)
 1. [Security](#)
 2. [Data Management](#)
 3. [DataOps](#)
 4. [Orchestration](#)
 5. [Software Engineering](#)
8. [Conclusion](#)
9. [Additional Resources](#)
11. [8. Queries, Modeling, and Transformation](#)
 1. [Queries](#)
 1. [What Is a Query?](#)
 2. [The Life of a Query](#)
 3. [The Query Optimizer](#)
 4. [Improving Query Performance](#)
 5. [Queries on Streaming Data](#)
 2. [Data Modeling](#)
 1. [What Is a Data Model?](#)
 2. [Conceptual, Logical, and Physical Data Models](#)
 3. [Normalization](#)
 4. [Techniques for Modeling Batch Analytical Data](#)
 5. [Modeling Streaming Data](#)
 3. [Transformations](#)
 1. [Batch Transformations](#)
 2. [Materialized Views, Federation, and Query Virtualization](#)
 3. [Streaming Transformations and Processing](#)
 4. [Whom You'll Work With](#)

1. [Upstream Stakeholders](#)
 2. [Downstream Stakeholders](#)
 5. [Undercurrents](#)
 1. [Security](#)
 2. [Data Management](#)
 3. [DataOps](#)
 4. [Data Architecture](#)
 5. [Orchestration](#)
 6. [Software Engineering](#)
 6. [Conclusion](#)
 7. [Additional Resources](#)
 12. [9. Serving Data for Analytics, Machine Learning, and Reverse ETL](#)
 1. [General Considerations for Serving Data](#)
 1. [Trust](#)
 2. [What's the Use Case, and Who's the User?](#)
 3. [Data Products](#)
 4. [Self-Service or Not?](#)
 5. [Data Definitions and Logic](#)
 6. [Data Mesh](#)
 2. [Analytics](#)
 1. [Business Analytics](#)
 2. [Operational Analytics](#)
 3. [Embedded Analytics](#)
 3. [Machine Learning](#)
 4. [What a Data Engineer Should Know About ML](#)
 5. [Ways to Serve Data for Analytics and ML](#)
 1. [File Exchange](#)
 2. [Databases](#)
 3. [Streaming Systems](#)
 4. [Query Federation](#)
 5. [Data Sharing](#)
 6. [Semantic and Metrics Layers](#)
 7. [Serving Data in Notebooks](#)
 6. [Reverse ETL](#)
 7. [Whom You'll Work With](#)
 8. [Undercurrents](#)
 1. [Security](#)
 2. [Data Management](#)
 3. [DataOps](#)
 4. [Data Architecture](#)
 5. [Orchestration](#)
 6. [Software Engineering](#)
 9. [Conclusion](#)
 10. [Additional Resources](#)
 13. [III. Security, Privacy, and the Future of Data Engineering](#)
 14. [10. Security and Privacy](#)

1. [People](#)
 1. [The Power of Negative Thinking](#)
 2. [Always Be Paranoid](#)
2. [Processes](#)
 1. [Security Theater Versus Security Habit](#)
 2. [Active Security](#)
 3. [The Principle of Least Privilege](#)
 4. [Shared Responsibility in the Cloud](#)
 5. [Always Back Up Your Data](#)
 6. [An Example Security Policy](#)
3. [Technology](#)
 1. [Patch and Update Systems](#)
 2. [Encryption](#)
 3. [Logging, Monitoring, and Alerting](#)
 4. [Network Access](#)
 5. [Security for Low-Level Data Engineering](#)
4. [Conclusion](#)
5. [Additional Resources](#)
15. [11. The Future of Data Engineering](#)
 1. [The Data Engineering Lifecycle Isn't Going Away](#)
 2. [The Decline of Complexity and the Rise of Easy-to-Use Data Tools](#)
 3. [The Cloud-Scale Data OS and Improved Interoperability](#)
 4. ["Enterprisey" Data Engineering](#)
 5. [Titles and Responsibilities Will Morph...](#)
 6. [Moving Beyond the Modern Data Stack, Toward the Live Data Stack](#)
 1. [The Live Data Stack](#)
 2. [Streaming Pipelines and Real-Time Analytical Databases](#)
 3. [The Fusion of Data with Applications](#)
 4. [The Tight Feedback Between Applications and ML](#)
 5. [Dark Matter Data and the Rise of...Spreadsheets?!](#)
 7. [Conclusion](#)
16. [A. Serialization and Compression Technical Details](#)
 1. [Serialization Formats](#)
 1. [Row-Based Serialization](#)
 2. [Columnar Serialization](#)
 3. [Hybrid Serialization](#)
 2. [Database Storage Engines](#)
 3. [Compression: gzip, bzip2, Snappy, Etc.](#)
17. [B. Cloud Networking](#)
 1. [Cloud Network Topology](#)
 1. [Data Egress Charges](#)
 2. [Availability Zones](#)
 3. [Regions](#)
 4. [GCP-Specific Networking and Multiregional Redundancy](#)
 5. [Direct Network Connections to the Clouds](#)
 2. [CDNs](#)

3. [The Future of Data Egress Fees](#)
18. [Index](#)