**Index**

**A**

* ACID properties, databases, [Relational Databases](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013392840), [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012620152)
* ActiveMQ, [Apache ActiveMQ](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011633080), [Summary of Message Broker Technologies](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011570232)
* Advanced Message Queuing Protocol (see AMQP)
* aggregates, [Designing the Application](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399014945288)
* aggregation operations
  + Course Correction pattern, [Course Correction Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#ix_aggreg_ops_course-corr)-[Update results with new information](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010845000)
  + Data Sharding pattern, [Considerations](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013018104)
  + log aggregation systems, [Observability and Monitoring](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011481912)
  + Windowed Aggregation pattern, [Windowed Aggregation Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#ix_aggreg_ops_windowed)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399011201096), [Summary of Streaming Data Processing Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399011015640)
  + without window, [Considerations](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399011220184)
* Amazon API Gateway, [Technologies for Implementing API Management Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch07.html#idm46399010249032)
* Amazon DynamoDB, [NoSQL Databases](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013335816), [Amazon DynamoDB](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012454200), [Summary of Technologies](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012379352)
* Amazon Kinesis, [Amazon Kinesis](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010532072), [Summary of Stream-Processing Technologies](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010489016)
* Amazon Simple Storage Service (S3), [Filesystem Storage](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013286024), [Amazon S3](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012437000), [Summary of Technologies](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012367624)
* Amazon SNS, [Amazon SNS](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011606216), [Summary of Message Broker Technologies](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011545592)
* Amazon SQS, [Amazon SQS](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011614952), [Summary of Message Broker Technologies](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011551000)
* AMQP (Advanced Message Queuing Protocol), [How it’s used in practice](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014578168), [AMQP](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014262600), [Apache ActiveMQ](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011630376)
* AOF (Append Only File), [Redis](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012460760)
* Apache ActiveMQ, [Apache ActiveMQ](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011637944), [Summary of Message Broker Technologies](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011568744)
* Apache Avro, [How it’s used in practice](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014389896)
* Apache Camel, [How it’s used in practice](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013544904)
* Apache Cassandra, [NoSQL Databases](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013357896), [NoSQL Databases](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013300984), [Apache Cassandra](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012497320), [Summary of Technologies](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012401096)
* Apache CouchBase, [NoSQL Databases](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013340312)
* Apache CouchDB, [NoSQL Databases](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013344472)
* Apache Flink, [Considerations](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010768344), [Apache Flink](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010535144), [Summary of Stream-Processing Technologies](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010492168)
* Apache HBase, [NoSQL Databases](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013354568), [Apache HBase](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012479832), [Summary of Technologies](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012396312)
* Apache HDFS, [Apache HDFS](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012448232), [Summary of Technologies](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012374936)
* Apache Kafka, [Kafka](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014254600), [Summary of Message Broker Technologies](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011559736)
  + abstractions for communication DevOps, [DevOps](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014181368)
  + ksqlDB, [ksqlDB](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010544136), [Summary of Stream-Processing Technologies](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010501464)
  + Schema Registry, [Schema definitions with Kafka Schema Registry](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014379880)
  + in sidecar bridge, [Sidecar bridge](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013893000)
* Apache Solr, [Filesystem Storage](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013293624)
* Apache Spark, [Apache Spark](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010553832), [Summary of Stream-Processing Technologies](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010497416)
* Apache Thrift, [gRPC](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014292024)
* API, [API Management Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch07.html#idm46399010403656)
* API consumption patterns, [Exposing capabilities as managed APIs](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399014872520), [API Management and Consumption Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399014762824), [API Consumption Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch07.html#ix_api_consumption)-[Summary of API Consumption Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch07.html#idm46399010152808)
* API control plane admin, [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch07.html#idm46399010381080)
* API creator, [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch07.html#idm46399010378792)
* API developer, [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch07.html#idm46399010377560)
* API developer portal, [Exposing capabilities as managed APIs](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399014869592), [Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399014006840), [API Management Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch07.html#idm46399010390952), [API developer portal](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch07.html#idm46399010343304)
* API gateway layer, [Exposing capabilities as managed APIs](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399014882072)
  + API management role, [API Management Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch07.html#idm46399010400888), [API gateway](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch07.html#idm46399010354968)
  + frontends relationship to, [Frontends Consuming Services Through API Gateway Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch07.html#idm46399010203176)
  + Request-Response in, [Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014686296)
  + Service Connectivity pattern, [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#ix_api-gatway-service-conn)-[How it’s used in practice](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399014127192)
* API Gateway pattern, [API Gateway Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch07.html#ix_api_gateway_pattern)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch07.html#idm46399010317912), [Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch07.html#ix_api_gateway_pattern2)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch07.html#idm46399010267144), [Summary of API Management Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch07.html#idm46399010233080)
* API management plane, [Exposing capabilities as managed APIs](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399014883848), [API management plane](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch07.html#idm46399010345144)
  + API layer in SOA-based design, [Designed as a Collection of Microservices](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399015393880)
  + building external APIs, [Building External APIs](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch08.html#ix_api_man_plane_external)-[Building External APIs](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch08.html#idm46399010069240)
  + composite services, [Considerations](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013653272)
  + frontends access through, [Frontends Consuming Services Through API Gateway Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch07.html#idm46399010198184)
  + management patterns, [API Management and Consumption Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399014776072), [API Management Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch07.html#ix_api_man_plane_man)-[Summary of API Management Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch07.html#idm46399010238616)
  + patterns in practice, [Building the High-Level Architecture](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch08.html#idm46399010094984)
  + reference architecture, [Reference Architecture for Cloud Native Apps](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399014748840)
  + Request-Response in, [Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014631048)
  + RPC pattern in, [Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014632472)
  + Service Connectivity pattern, [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399014145768)
* API Microgateway pattern, [API Microgateway Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch07.html#ix_api_microgateway)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch07.html#idm46399010287288)
* API product, [API Management Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch07.html#idm46399010394840)
* API proxy, [API Management Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch07.html#idm46399010401928)
* API publisher, [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch07.html#idm46399010374408)
* Append Only File (see AOF)
* application developer, [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch07.html#idm46399010372072)
* application metrics, data stores, [Observability and Monitoring](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012315416)
* AsyncAPI, [How it’s used in practice](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014392472)
* asynchronous communication, [Communication Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399014821032), [Communication Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014724840), [Asynchronous Messaging Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#ix_async_ch2)-[Summary of Asynchronous Messaging Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014473128)
  + (see also EDA)
  + DevOps, [DevOps](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014189432)
  + Materialized View pattern, [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012899976)
  + Multiple-Receiver pattern, [Multiple-Receiver Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#ix_async-com_multi-receiver)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014529384), [Summary of Asynchronous Messaging Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014484648), [NATS](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014247608)
  + observability and monitoring, [Observability and Monitoring](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014192696)
  + Request-Reply pattern, [Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014683544), [Asynchronous Request-Reply Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#ix_async-com_req-reply)-[Summary of Asynchronous Messaging Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014475960), [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011781848)
  + security, [Security](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014209352)
  + service definition patterns, [Service Definition in Asynchronous Communication](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#ix_async-com_serv-def)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014360296)
  + Single-Receiver pattern, [Single-Receiver Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#ix_async_single-receiver)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014560584), [Summary of Asynchronous Messaging Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014478648), [NATS](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014244280)
  + technologies of, [Technologies to Implement Asynchronous Messaging Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#ix_async-com_tech)-[NATS](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014239640)
  + testing, [Testing](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014228136)
* Asynchronous Request-Reply pattern, [Asynchronous Request-Reply Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#ix_req-reply_pattern_ch2)-[Summary of Asynchronous Messaging Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014474488), [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011787080)
* at-least-once delivery, message broker guarantee, [Event-Driven Architecture](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399012239208)
* at-most-once delivery, message broker guarantee, [Event-Driven Architecture](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399012244440), [Deliver events with best effort](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399012115128)
* atomicity, database, [Relational Databases](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013391880), [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012616952)
* automating app development, release, and deployment, [Automate the Development Life Cycle](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#ix_auto_dev_deploy)-[Automate the Development Life Cycle](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399014989528), [Automating the Development, Release, and Deployment](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399014864584), [Implementing Dynamic Management in a Cloud Environment](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch08.html#idm46399010021704)
* autoscaling, [DevOps](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011475880)
* availability, database, [NoSQL Databases](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013326952)
* Avro, [How it’s used in practice](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014396328), [Event Schema](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399012203864)
* AWS, [Amazon DynamoDB](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012451624), [Amazon S3](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012430648)
  + (see also specific Amazon technologies)
* Azure, [How it’s used in practice](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013557224)
* Azure API Management, [Technologies for Implementing API Management Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch07.html#idm46399010247912)
* Azure Cosmos DB, [NoSQL Databases](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013334536), [Azure Cosmos DB](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012432904), [Summary of Technologies](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012362056)
* Azure Event Grid, [Azure Event Grid](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011602968), [Summary of Message Broker Technologies](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011541000)
* Azure Logic Apps, [Technologies for Implementing Service Composition Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013505288)
* Azure Service Bus Queues, [Azure Service Bus Queues](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011593944), [Summary of Message Broker Technologies](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011535384)
* Azure Storage services, [Filesystem Storage](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013283048)
* Azure Stream Analytics, [Azure Stream Analytics](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010540792), [Summary of Stream-Processing Technologies](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010484024)

**B**

* Backend for Frontends pattern, [Backend for Frontends Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch07.html#ix_backend_frontends)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch07.html#idm46399010159416)
* backing service, [Service-led interactions](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399014894968), [Implementing Dynamic Management in a Cloud Environment](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch08.html#idm46399010002504)
* backward compatibility of event schema, [DevOps](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011466856)
* batch-processing systems, [Data Architecture](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013441128)
* best effort message delivery, [Make sure all events are delivered to all subscribers](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399012106824)
* BFFs (backends for frontend applications), [Backend for Frontends Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch07.html#ix_backend_frontends_bff)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch07.html#idm46399010161208)
* bindings, RabbitMQ, [RabbitMQ](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011622808)
* Boomi, [Technologies for Implementing Service Composition Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013507000)
* bottom-up API management, [How it’s used in practice](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch07.html#idm46399010351816)
* bounded contexts, [Designing the Application](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399014953416), [Security](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010458120)
* broadcast events, [Broadcast events](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399012120952)
* Buffered Event Ordering pattern, [Considerations](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399011108344), [Buffered Event Ordering Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#ix_buffer_event_order)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010877256), [Considerations](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010773112)
* Building Microservices (O’Reilly), [Designed as a Collection of Microservices](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399015361128)
* business domain analysis, [Designed as a Collection of Microservices](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399015372824), [Designing the Application](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399014949544)

**C**

* cache aside, [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012810440)
* cache hit or miss, [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012815496)
* Caching pattern, [Data Composition Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#ix_caching_pattern1)-[Data Composition Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013214984), [Caching Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#ix_caching_pattern2)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012722200), [Summary of Performance Optimization Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012650472), [Redis](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012464840)
* Cadence, [Technologies for Implementing Service Composition Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013498392)
* callback-based event delivery, [Request Callback Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#ix_callback_event_delivery)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011941464), [Summary of Event-Delivery Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011893448)
* Camel, [How it’s used in practice](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013554232)
* Camunda, [How it’s used in practice](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013546200), [Technologies for Implementing Service Composition Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013499512)
* CAP theorem, [NoSQL Databases](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013329624)
* Cassandra, [NoSQL Databases](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013355928), [NoSQL Databases](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013301992), [Apache Cassandra](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012494424), [Summary of Technologies](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012400024)
* causation IDs, logging and tracking events, [Observability and Monitoring](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011481576), [Observability and Monitoring](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010434920)
* CD (continuous deployment), [Automate the Development Life Cycle](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399015002760)
* CDNs (content delivery networks), [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012702008)
* centralized data management, [Centralized Data Management](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013268840)
* CI (continuous integration), [Automate the Development Life Cycle](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399015010088)
* circuit breaker, in Resilient Connectivity pattern, [Circuit breaker](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#ix_circuit_breaker)-[Circuit breaker](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013947800)
* circuit reset time-out, [Circuit breaker](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013949544)
* client libraries, [Technologies for Implementing Service Connectivity Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013740040)
* client or server stubs, [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014665864)
* Client-Side Mashup pattern, [Client-Side Mashup Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#ix_client-side_mashup)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013119784), [Summary of Data Composition Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013089624)
* client-side service discovery, [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399014046936)
* cloud native architecture, [Introduction to Cloud Native](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#ix_cloud_native_ch1)-[What Is Cloud Native?](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399015408072)
  + automating development life cycle, [Automate the Development Life Cycle](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#ix_cloudnative_auto_dev_deploy)-[Automate the Development Life Cycle](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399014987992)
  + containerization, [Use Containerization and Container Orchestration](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#ix_cloudnative_container)-[Virtual machines](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399015021464)
  + design patterns for building apps, [Design Patterns for Building Cloud Native Apps](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#ix_cloudnative_design)-[API Management and Consumption Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399014759192)
  + dynamic management, [Dynamic Management](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399014981192)
  + methodology for building apps, [Methodology for Building Cloud Native Apps](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#ix_cloudnative_methods)-[Observability and Monitoring](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399014835704)
  + as microservices, [Designed as a Collection of Microservices](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#ix_cloudnative_microserv)-[Designed as a Collection of Microservices](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399015348152)
  + patterns in practice, [Cloud Native Patterns in Practice](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch08.html#ix_cloudnative_practice)-[Summary](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch08.html#idm46399009988728)
  + reference architecture, [Reference Architecture for Cloud Native Apps](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#ix_cloudnative_ref_archit)-[Reference Architecture for Cloud Native Apps](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399014737848)
* Cloudera, [Apache HDFS](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012446600)
* CloudEvents, [How it’s used in practice](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014383208), [CloudEvents](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399012215064)
* CNCF (Cloud Native Computing Foundation), [What Is Cloud Native?](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399015413528), [CloudEvents](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399012213784)
* column store, NoSQL, [NoSQL Databases](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013360824), [NoSQL Databases](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013315592)
* Command and Query Responsibility Segregation pattern (see CQRS pattern)
* commit log, Event Sourcing pattern, [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011870600), [Considerations](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011859208)
* communication patterns, [Communication Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399014822824), [Communication Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#ix_communication_ch2)-[Summary](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014172632)
  + (see also asynchronous communication; synchronous communication)
  + DevOps, [DevOps](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014191032)
  + observability and monitoring, [Observability and Monitoring](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014201304)
  + security, [Security](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014223000)
  + service definition patterns, [Service Definition Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#ix_communication_serv-def)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014368328)
  + testing, [Testing](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014234712)
* compensating operation, Saga pattern, [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013573208)
* Composite Data Services pattern, [Composite Data Services Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#ix_comp_data_serv)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013151480), [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012907880)
* compositions (composite services), [Interservice communication and compositions](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399014878760), [Compositions](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399014800392)
  + (see also data composition patterns; service composition patterns)
* configuration data type, [Types and Forms of Data](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013430744)
* configuration management tools, [Automate the Development Life Cycle](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399014995816)
* connectivity patterns, [Connectivity, Compositions, and APIs](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#ix_connectivity_ch1)-[Exposing capabilities as managed APIs](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399014868088), [Connectivity](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399014803368), [Connectivity Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#ix_connectivity_ch3)-[Summary](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013468744)
  + Resilient Connectivity pattern, [Resilient Connectivity Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#ix_connectivity_resilient)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013926664), [Summary of Connectivity Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013709336), [Achieve high availability by relaxing the data store dependency](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012773128)
  + Service Abstraction pattern, [Service Abstraction Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#ix_connectivity_serv-abstract)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399014069480), [Technologies for Implementing Service Connectivity Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013755944), [Summary of Connectivity Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013719784)
  + Service Connectivity pattern, [Service Connectivity Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#ix_connectivity_service-conn)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399014124696), [Technologies for Implementing Service Connectivity Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#ix_connect_serv_connect)-[Technologies for Implementing Service Connectivity Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013734008)
  + Service Mesh pattern, [Service Mesh Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#ix_connectivity_serv-mesh)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013809352), [Technologies for Implementing Service Connectivity Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013747992), [Summary of Connectivity Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013699352)
  + Service Registry and Discovery pattern, [Service Registry and Discovery Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#ix_connectivity_serv-reg-discov)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399014021512), [Summary of Connectivity Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013714344)
  + Sidecar pattern, [Sidecar Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#ix_connectivity_sidecar_pattern)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013868616), [Summary of Connectivity Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013701384)
  + Sidecarless Service Mesh pattern, [Sidecarless Service Mesh Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#ix_connectivity_sidecarless_serv-mesh)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013769944), [Summary of Connectivity Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013691208)
  + technologies of, [Technologies for Implementing Service Connectivity Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#ix_connectivity_tech)-[Technologies for Implementing Service Connectivity Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013731176)
* consistency, database, [Relational Databases](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013387688), [NoSQL Databases](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013323992), [Considerations](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012973800), [Considerations](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012888728), [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012614936)
* container instance, [What are containers?](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399015084008)
* container registry, [What are containers?](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399015085928)
* containerization, [Use Containerization and Container Orchestration](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#ix_container_ch1)-[Virtual machines](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399015015464), [Containerization](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399014915992)
  + (see also Kubernetes)
  + container orchestration, [Why container orchestration?](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#ix_container_orchest)-[Virtual machines](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399015020680), [Sidecar bridge](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013898680), [Orchestration Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#ix_container_c-orchest)-[Summary of Orchestration Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011672424), [How it’s used in practice](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch07.html#idm46399010304920)
  + serverless functions, [Serverless functions](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399015025928)
  + service abstraction, [Service-led interactions](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399014892376)
* content delivery networks (see CDNs)
* continuous delivery, [Automate the Development Life Cycle](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399014998552)
* Continuous Delivery (Humble and Farley), [Automating the Development, Release, and Deployment](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399014861160)
* continuous deployment (see CD)
* continuous integration (see CI)
* control plane, [Exposing capabilities as managed APIs](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399014885320)
  + (see also API management plane)
  + API control plane admin, [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch07.html#idm46399010382424)
  + container orchestration, [Kubernetes](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399015049496), [Considerations](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013881928)
  + dynamic management, [Control Plane for Dynamic Management](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399014849032)
  + for hybrid and multicloud deployments, [Implementing Dynamic Management in a Cloud Environment](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch08.html#idm46399009993912)
  + and Service Mesh pattern, [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013847976)
  + Sidecarless Service Mesh pattern, [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#ix_control_plane_sidecarless)-[Sidecarless gRPC services in Google Traffic Director](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013779112)
* correcting previous decisions, Course Correction pattern, [Correct previous decisions](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010842936)
* Cosmos DB, [NoSQL Databases](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013333528), [Azure Cosmos DB](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012431624), [Summary of Technologies](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012363416)
* CouchBase, [NoSQL Databases](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013341912)
* CouchDB, [NoSQL Databases](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013342952)
* Course Correction pattern, [Course Correction Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#ix_course_corr_pattern)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010823416), [Summary of Scaling and Performance Optimization Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010724888)
* CPU/bandwidth trade-off, Data Locality pattern, [Reduce bandwidth usage when retrieving data](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012842376)
* CQRS pattern, [Command and Query Responsibility Segregation Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#ix_cqrs_pattern)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012953992)

**D**

* Dapr, and sidecar bridge, [Sidecar bridge](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013886648), [Technologies for Implementing Service Connectivity Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013742184)
* data composition patterns, [Data Composition Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#ix_data_comp_ch4)-[Summary of Data Composition Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013108184)
  + Client-Side Mashup pattern, [Client-Side Mashup Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#ix_data_comp_client-side_mashup)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013118488), [Summary of Data Composition Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013088200)
  + Composite Data Services pattern, [Composite Data Services Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#ix_data_comp_comp_data_serv)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013143272), [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012904248)
  + Data Service pattern, [Data Service Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#ix_data_comp_data_serv)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013173688), [Summary of Data Composition Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013097848), [Considerations](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012692248)
* data denormalization, [Performance Optimization Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012919848)
* Data Locality pattern, [Data Locality Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#ix_data_locality)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012834184), [Summary of Performance Optimization Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012657048)
* data management, [Data Management Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399014788472), [Data Management Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#ix_data_man_ch4)-[Summary](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012270664)
  + data architecture, [Data Architecture](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#ix_data_man_architect)-[Data Architecture](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013439576)
  + data composition patterns, [Data Composition Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#ix_data_man_data_comp)-[Summary of Data Composition Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013106744)
  + data management techniques, [Data Management](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#ix_data_man_techniques)-[Data Management Summary](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013236008)
  + data scaling patterns, [Data Scaling Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#ix_data_man_scaling)-[Summary of Data Scaling Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012947208)
  + data stores, [Data Architecture](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013449224), [Data Stores](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#ix_data_man_stores)-[Data Store Summary](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013272664)
  + DevOps, [DevOps](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012291944)
  + observability and monitoring, [Observability and Monitoring](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#ix_data_man_observ)-[Observability and Monitoring](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012296456)
  + performance optimization patterns, [Performance Optimization Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#ix_data_man_perf_opt)-[Summary of Performance Optimization Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012674200)
  + performing, [Performing Data Management](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch08.html#ix_data_man_perform)-[Performing Data Management](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch08.html#idm46399010047672)
  + reliability patterns, [Reliability Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#ix_data_man_reliability)-[Summary of Transaction Reliability Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012563384)
  + security, [Security: Vault Key Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#ix_data_man_security)-[Summary of the Vault Key Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012521128), [Security](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012335976)
  + technologies of, [Technologies for Implementing Data Management Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#ix_data_man_tech_of)-[Summary of Technologies](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012412376)
  + testing, [Testing](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#ix_data_man_test)-[Testing](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012344152)
  + types and forms of data, [Types and Forms of Data](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013434008)
* data plane, Service Mesh pattern, [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013849608)
* data processing patterns, streaming, [Streaming Data Processing Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#ix_data_proc_stream)-[Summary of Streaming Data Processing Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399011038888)
  + Filters and Thresholds pattern, [Filters and Thresholds Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#ix_data_proc_stream_filter-thresh)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399011348872), [Summary of Streaming Data Processing Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399011021064)
  + Machine Learner pattern, [Machine Learner Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#ix_data_proc_stream_machlearn)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399011045240)
  + Stream Join pattern, [Stream Join Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#ix_data_proc_stream_join)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399011164504), [Summary of Streaming Data Processing Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399011009592)
  + Temporal Event Ordering pattern, [Temporal Event Ordering Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#ix_data_proc_stream_temp_order)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399011103576), [Summary of Streaming Data Processing Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399011002056), [Synchronize events generated from nonsynchronized sources](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010797176)
  + Transformation pattern, [Transformation Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#ix_data_process_stream_trans)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399011389816), [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399011254056), [Summary of Streaming Data Processing Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399011031384)
  + Windowed Aggregation pattern, [Windowed Aggregation Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#ix_data_proc_stream_windowed)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399011199800), [Summary of Streaming Data Processing Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399011014248)
* data scaling patterns, [Data Scaling Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#ix_data_scaling_ch4)-[Summary of Data Scaling Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012950376)
  + CQRS pattern, [Command and Query Responsibility Segregation Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#ix_data_scaling_cqrs)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012955112)
  + Data Sharding pattern, [Data Sharding Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#ix_data_scaling_sharding)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013012200), [Considerations](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012972504), [Summary of Data Scaling Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012940072)
* Data Service pattern, [Data Service Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#ix_data_serv_pattern)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013173928), [Summary of Data Composition Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013099368), [Considerations](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012693992)
* Data Sharding pattern, [Data Sharding Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#ix_data_sharding)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013003032), [Considerations](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012979336), [Summary of Data Scaling Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012938312)
* data sources, [Data Architecture](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013459256)
* data store logs, [Observability and Monitoring](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012297432)
* data stores, [Data Stores](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#ix_data_stores_ch4)-[Data Store Summary](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013274808)
  + application metrics, [Observability and Monitoring](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012322392)
  + client access, [How it’s used in practice](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012534056)
  + Data Service pattern, [Expose abstract legacy/proprietary data stores](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013192376)
  + dedicated, [Data Management Summary](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013243256), [Performing Data Management](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch08.html#idm46399010046680)
  + reducing lock contention, [Considerations](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012969816), [Considerations](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012886680), [Reduce data store contention](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012785080)
  + system metrics, [Observability and Monitoring](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012298776)
* data, characteristics of cloud native, [Data Architecture](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013439192)
* data-ingestion systems, [Data Architecture](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013453416)
* data-retrieval efficiency, [Improve data-retrieval efficiency](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012895352), [Reduce latency when retrieving data](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012845064), [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012814024), [Improve time to retrieve data](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012794216)
* DDD (domain-driven design), [Designed as a Collection of Microservices](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399015364376), [Designing the Application](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399014952136)
* deadlines, in Resilient Connectivity pattern, [Deadlines](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013969064)
* decentralized data management, [Decentralized Data Management](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013260648)
* decoupled configurations, [Decoupled configurations](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399014927960), [Build large-scale decoupled systems](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011733992), [Google Cloud Pub/Sub](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011584136)
* dedicated client libraries, [Technologies for Implementing Service Connectivity Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013746504)
* dedicated data stores, [Data Management Summary](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013241928), [Performing Data Management](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch08.html#idm46399010047288)
* Deployment abstraction, Kubernetes, [Kubernetes](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399015036344)
* deploys, [Independent codebase](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399014937960)
* designing the application, [Designing the Application](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399014957080)
* developer portal, [Exposing capabilities as managed APIs](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399014871240), [Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399014017480), [API Management Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch07.html#idm46399010391752), [API developer portal](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch07.html#idm46399010353496)
* developing the application, [Developing the Application](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#ix_dev_app)-[Containerization](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399014904872)
* DevOps
  + communication patterns, [DevOps](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014195272)
  + data management, [DevOps](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012290568)
  + event-driven architecture, [DevOps](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#ix_devops_eda)-[DevOps](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011458776)
  + in Service Mesh Sidecar pattern, [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch07.html#idm46399010280152)
  + stream-processing systems, [DevOps](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010443304)
* Direct Frontend-to-Microservices Communication pattern, [Direct Frontend-to-Microservices Communication Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch07.html#ix_frontend_microserv_comm)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch07.html#idm46399010204648), [Summary of API Consumption Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch07.html#idm46399010146536)
* distributed transactions, Saga pattern, [Saga Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#ix_distr_transaction_saga)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013536632)
* distributed versus monolithic microservices applications, [Designed as a Collection of Microservices](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#ix_distr_vs_mono_apps)-[Designed as a Collection of Microservices](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399015382584), [Data Management Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399014785992)
* Docker, [What are containers?](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399015079608)
  + (see also containerization)
* Docker Compose, [Testing](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014224760)
* document store, NoSQL, [NoSQL Databases](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013349400), [NoSQL Databases](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013320392)
* domain models, building with Event Sourcing, [Build different domain models](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011856824), [Considerations](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011860472)
* domain-driven design (see DDD)
* Driven Design paradigm, [Designed as a Collection of Microservices](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399015368200)
* dumb pipes, microservices, [Designed as a Collection of Microservices](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399015353048)
* durability, database, [Relational Databases](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013385320), [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012611256), [Considerations](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011855448)
* durable subscription, [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399012131288)
* durable topic, in Multiple-Receiver pattern, [How it’s used in practice](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014521976)
* dynamic management, [Dynamic Management](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399014984824), [Running in a Dynamic Environment](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#ix_dynamic_man_ch1)-[Observability and Monitoring](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399014835176), [Implementing Dynamic Management in a Cloud Environment](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch08.html#ix_dynamic_man_ch8)-[Implementing Dynamic Management in a Cloud Environment](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch08.html#idm46399009991816)
* dynamic runtime environment, [Implementing Dynamic Management in a Cloud Environment](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch08.html#idm46399010006392)
* DynamoDB, [NoSQL Databases](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013331864), [Amazon DynamoDB](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012455480), [Summary of Technologies](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012380360)

**E**

* EDA (event-driven architecture), [Event-Driven Architecture Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399014781528), [Event-Driven Architecture Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#ix_EDA_ch5)-[Summary](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011450392)
  + (see also stream-processing systems)
  + CloudEvents, [CloudEvents](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399012218904)
  + DevOps, [DevOps](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011490920)
  + event schema, [Event Schema](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399012208024), [DevOps](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011463736)
  + event-delivery patterns, [Event-Delivery Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#ix_eda_event-delivery)-[Summary of Event-Delivery Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011933944)
  + exactly once processing, [Exactly Once Processing](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399012236888)
  + message broker categories, [Message Broker Categories](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399012233512)
  + Multiple-Receiver pattern in, [Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014533384)
  + observability and monitoring, [Observability and Monitoring](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011493816)
  + orchestration patterns, [Orchestration Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#ix_eda_orchest_patterns)-[Summary of Orchestration Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011675368)
  + patterns in practice, [Using Event-Driven Architecture](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch08.html#ix_eda_practice)-[Using Event-Driven Architecture](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch08.html#idm46399010024968)
  + reference architecture, [Reference Architecture for Cloud Native Apps](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399014744616)
  + security, [Security](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011504232)
  + service definitions, [How it’s used in practice](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014386408)
  + state management patterns, [State Management Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#ix_eda_state_man)-[Summary of State Management Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011820856)
  + technologies of, [Technologies for Event-Driven Architecture](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#ix_eda_tech_of)-[Summary of Message Broker Technologies](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011581016)
  + testing, [Testing](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#ix_eda_test)-[Testing](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011515352)
  + Transaction pattern in, [Combine operations across multiple systems](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012598408)
* Ehcache, [NoSQL Databases](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013371096)
* EIPs (enterprise integration patterns), [Compositions](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399014797688)
* Elasticsearch, [Filesystem Storage](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013292344)
* encryption
  + data, [Security](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012328072)
  + event, [Security](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011508664), [DevOps](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010426888)
* end-of-sequence message, [Considerations](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010936376)
* enterprise integration patterns (see EIPs)
* Envoy proxy, [Sidecar proxy](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013900376)
* Envoy xDS API, [Sidecarless gRPC services in Google Traffic Director](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013789624), [Considerations](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013775768), [Technologies for Implementing Service Connectivity Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013736232)
* ESB (enterprise service bus), [Designed as a Collection of Microservices](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#ix_esb_ch1)-[Designed as a Collection of Microservices](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399015371992), [Compositions](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399014792856), [Service Mesh Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013859080), [Technologies for Event-Driven Architecture](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011643560)
* Esper, [Esper](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010560936), [Summary of Stream-Processing Technologies](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010510584)
* event broker, [Asynchronous Messaging Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014599304)
* event bus, Multiple-Receiver pattern, [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014543432)
* Event Grid, [Azure Event Grid](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011594744), [Summary of Message Broker Technologies](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011539608)
* event schema, [Event Schema](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399012209176), [DevOps](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011468008)
* Event Sourcing pattern, [Considerations](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012972952), [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012898568), [Considerations](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012754760), [Event Sourcing Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#ix_event_sourcing_pattern)-[Summary of State Management Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011836920)
* event-delivery patterns, [Event-Delivery Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#ix_event-delivery_patterns)-[Summary of Event-Delivery Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011934920), [Publisher-Subscriber Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399012137736)
  + (see also Publisher-Subscriber pattern)
  + Fire and Forget pattern, [Fire and Forget Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#ix_event-delivery_fire-forget)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399012056104), [Summary of Event-Delivery Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011911848)
  + Polling Pattern, [Polling Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#ix_event-delivery_polling)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399012000040), [Summary of Event-Delivery Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011901288)
  + Producer-Consumer pattern, [Producer-Consumer Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#ix_event-delivery_prod-consum)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399012147832), [Summary of Event-Delivery Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011925352)
  + Request Callback pattern, [Request Callback Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#ix_event_delivery_req_callback)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011940184), [Summary of Event-Delivery Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011892408)
  + Store and Forward pattern, [Store and Forward Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#ix_event-delivery_store-forward)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399012033976), [Summary of Event-Delivery Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011905128)
* event-driven architecture (see EDA)
* eventing, [Performing Data Management](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch08.html#idm46399010041224)
* events, [Event-Driven Architecture Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399012263576)
* eviction policy, cached data, [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012805016)
* exactly once processing of events, [Exactly Once Processing](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399012235320)
* exchanges, RabbitMQ, [RabbitMQ](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011624920)
* explicit dependencies, [Explicit dependencies](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399014929800)
* external APIs, building, [Building External APIs](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch08.html#ix_external_api_build)-[Building External APIs](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch08.html#idm46399010071304)

**F**

* facade, microservices, [Designed as a Collection of Microservices](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399015346424)
* fail-fast, in Resilient Connectivity pattern, [Fail-fast](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013939416)
* failover
  + Service Abstraction pattern, [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399014094520)
  + Two-Node Failover pattern, [Two-Node Failover Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#ix_failover_two-node)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010595768)
* Farley, David
  + Continuous Delivery, [Automating the Development, Release, and Deployment](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399014859912)
* FIFO (first in, first out) order, [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012801096), [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399012183160), [Amazon SQS](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011612712), [Azure Service Bus Queues](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011589992)
* filesystem storage, data stores, [Filesystem Storage](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013295864)
* filtering of events, [Filter events by category](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399011365480)
* Filters and Thresholds pattern, [Filters and Thresholds Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#ix_filter-thresh_pattern)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399011347432), [Summary of Streaming Data Processing Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399011022152)
* Fire and Forget pattern, [Fire and Forget Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#ix_fire-forget_pattern)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399012056584), [Summary of Event-Delivery Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011914232)
* Flink, [Considerations](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010763352), [Apache Flink](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010533864), [Summary of Stream-Processing Technologies](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010490856)
* Fowler, Martin
  + Software Delivery Guide, [DevOps](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014177048)
* Frontends Consuming Services through API Gateway pattern, [Frontends Consuming Services Through API Gateway Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch07.html#ix_frontend_consume)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch07.html#idm46399010176104), [Summary of API Consumption Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch07.html#idm46399010142488)
* functional data sharding, [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013054696)

**G**

* geographical distribution of data, Data Sharding, [Geographically distribute data](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013022472)
* Go kit, [Technologies for Implementing Service Composition Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013520664)
* Google Apigee, [Technologies for Implementing API Management Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch07.html#idm46399010246824)
* Google Cloud Pub/Sub, [Google Cloud Pub/Sub](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011586536), [Summary of Message Broker Technologies](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011529800)
* Google Cloud Spanner, [Google Cloud Spanner](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012416792), [Summary of Technologies](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012357464)
* Google Cloud Storage, [Filesystem Storage](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013282200)
* Google Dataflow, [Google Dataflow](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010523160), [Summary of Stream-Processing Technologies](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010478792)
* Google Traffic Director, [Sidecarless gRPC services in Google Traffic Director](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013788024)
* graph store, NoSQL, [NoSQL Databases](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013338152), [NoSQL Databases](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013321320)
* GraphQL, [How it’s used in practice](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014431256), [GraphQL](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#ix_GraphQL)-[GraphQL](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014326952), [Summary of Synchronous Messaging Technologies](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014283016), [WebSockets](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011975656), [API Management Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch07.html#idm46399010399640)
* gRPC, [How it’s used in practice](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014649096), [How it’s used in practice](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014432520), [gRPC](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#ix_grpc_ch2)-[gRPC](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014296792), [Sidecarless gRPC services in Google Traffic Director](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013784616)

**H**

* hash-based data sharding, [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013043608)
* HBase, [NoSQL Databases](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013353608), [Apache HBase](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012478552), [Summary of Technologies](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012394024)
* HDFS (Hadoop Distributed File System), [Filesystem Storage](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013291096), [Apache HBase](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012483480), [Apache HDFS](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012449480), [Summary of Technologies](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012373848)
* high-level architecture, building, [Building the High-Level Architecture](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch08.html#ix_high-level_architect)-[Building the High-Level Architecture](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch08.html#idm46399010089736)
* horizontal data sharding, [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013069512), [Scale beyond a single node](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013042744)
* HTTP services, [How it’s used in practice](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014698200), [RESTful Services](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014344856), [Webhooks](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011965624)
* HTTP2 protocol, [How it’s used in practice](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014650088), [gRPC](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014307704), [gRPC](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014293032)
* HTTPS, [Amazon SQS](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011610408), [Google Cloud Pub/Sub](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011585112)
* Humble, Jez
  + Continuous Delivery, [Automating the Development, Release, and Deployment](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399014858552)
* hybrid data management, [Hybrid Data Management](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013252216)
* hypervisors, [What are containers?](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399015090328)

**I**

* IaC (infrastructure as code), [Automate the Development Life Cycle](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399014991512)
* IDL (interface definition language), [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014668376)
* independent codebase, developing the application, [Independent codebase](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399014939080)
* independent testing, [Independent testing](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399014920696)
* indexing data for performance optimization, [Performance Optimization Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012919272), [MongoDB](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012471720)
* indexing system, filesystem storage, [Filesystem Storage](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013294888)
* Indrasiri, Kasun
  + Microservices for the Enterprise, [Security](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014217800)
* infrastructure as code (see IaC)
* input data type, [Types and Forms of Data](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013426872)
* integration platform-as-a-service (see iPaaS)
* integration services (see compositions)
* integration tests, microservice, [Independent testing](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399014917064)
* interface definition language (see IDL)
* iPaaS (integration platform-as-a-service), [Technologies for Implementing Service Composition Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013509544)
* isolation, database, [Relational Databases](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013386920), [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012612728)
* Istio service mesh implementation, [How it’s used in practice](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013825400), [Technologies for Implementing Service Connectivity Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013750440)

**J**

* joining events (see Stream Join pattern)
* JSON (JavaScript Object Notation), [Kubernetes](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399015033720), [How it’s used in practice](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014387528), [RESTful Services](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014340536), [Types and Forms of Data](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013411512), [NoSQL Databases](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013346984), [Event Schema](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399012206536)

**K**

* Kafka (see Apache Kafka)
* KEDA (Kubernetes Event Driven Autoscaler), [DevOps](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011475032)
* key-value store, NoSQL, [NoSQL Databases](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013372408), [NoSQL Databases](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013316664)
* Kinesis, [Amazon Kinesis](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010530888), [Summary of Stream-Processing Technologies](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010485912)
* Knative, [Technologies for Implementing Service Connectivity Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013751288)
* Kong, [Technologies for Implementing API Management Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch07.html#idm46399010256568)
* ksqlDB, [ksqlDB](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010546728), [Summary of Stream-Processing Technologies](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010502984)
* kube-scheduler, [Kubernetes](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399015046168)
* Kubernetes, [Kubernetes](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#ix_kubernetes_ch1)-[Kubernetes](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399015030936)
  + automating testing with, [Testing](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014221144)
  + DevOps for asynchronous communication, [DevOps](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014188664)
  + implementing Service Connectivity patterns, [Technologies for Implementing Service Connectivity Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013758552)
  + reconciliation loops, [Automate the Development Life Cycle](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399014990712)
  + Service abstraction in, [Kubernetes](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399015037976), [Service-led interactions](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399014897688), [Kubernetes services](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#ix_kubernetes_serv-abstract)-[Considerations](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399014067048)
  + service discovery in, [Service discovery in Kubernetes](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399014025416)
  + Sidecar pattern with, [How it’s used in practice](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013913464)
  + as universal runtime abstraction, [Running in a Dynamic Environment](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399014850120)
* Kubernetes Event Driven Autoscaler (see KEDA)

**L**

* legacy data stores, abstract, [Expose abstract legacy/proprietary data stores](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013185576)
* length batch window, [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399011286360)
* length sliding window, [Aggregate events over length](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399011239528)
* Linkerd service mesh implementation, [How it’s used in practice](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013831624)
* load balancing
  + in Data Sharding pattern, [Considerations](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013019256)
  + Service Abstraction pattern, [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399014088920)
  + service registry, [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399014042328)
* local cache, [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012806728), [Considerations](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012760216)
* log aggregation systems, [Observability and Monitoring](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011491832)
* log processing, Producer-Consumer pattern, [Handle sudden bursts of events](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399012172568)
* log-based message brokers, [Message Broker Categories](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399012231112), [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010697336)
* Logic Apps, [Technologies for Implementing Service Composition Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013506440)
* long polling technique, [Considerations](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399012007416)
* lookup-based data sharding, [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013048776)
* LRU (least recently used) execution policy, [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012802936)

**M**

* Machine Learner pattern, [Machine Learner Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#ix_machlearn)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399011048584)
* managed API, [Exposing capabilities as managed APIs](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399014886744)
  + (see also API management plane)
* Materialized View pattern, [Materialized View Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#ix_material_view)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012866408), [Summary of Performance Optimization Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012664072)
* mediator microservices, [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#ix_mediator_microserv)-[Considerations](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011754664)
* Mediator pattern, [Mediator Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#ix_mediator_pattern)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011752616), [Summary of Orchestration Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011665496)
* Memcached, [NoSQL Databases](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013369976)
* message brokers, [Communication Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014730904), [Asynchronous Messaging Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014597064)
  + (see also EDA)
  + categories, [Message Broker Categories](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399012222968), [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010697720)
  + DevOps, [DevOps](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#ix_message_broker_devops)-[DevOps](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011460856)
  + event-delivery guarantees, [Event-Driven Architecture](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399012251880), [How it’s used in practice](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399012125320)
  + in Service Choreography pattern, [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#ix_message_broker_choreog)-[How it’s used in practice](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013617448)
  + in Single-Receiver pattern, [How it’s used in practice](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014576584)
  + technologies of, [Technologies for Event-Driven Architecture](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#ix_message_broker_tech_of)-[Summary of Message Broker Technologies](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011579736)
* message transformation, [Message transformation](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399011413928)
* messages, [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014706536)
* metadata repository, [Service Definition in Asynchronous Communication](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#ix_metadata_rep)-[How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014394200)
* Micronaut, [Technologies for Implementing Service Composition Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013522152)
* microservices
  + cloud native architecture as, [Designed as a Collection of Microservices](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#ix_microserv_cloudnative)-[Designed as a Collection of Microservices](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399015345304)
  + coupling between microservices, reducing, [Considerations](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013184424)
  + mediator, [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#ix_microserv_mediator)-[Considerations](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011757768)
  + monolithic versus distributed applications, [Designed as a Collection of Microservices](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#ix_microserv_distr_vs_mono)-[Designed as a Collection of Microservices](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399015384440)
  + reactive, [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013634104)
  + service-led interactions, [Service-led interactions](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399014901944)
  + testing of, [Independent testing](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399014922088)
* Microservices for the Enterprise (Indrasiri and Siriwardena), [Security](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014213288)
* MongoDB, [NoSQL Databases](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013345480), [MongoDB](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012482152), [Summary of Technologies](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012391528)
* MongoDB Cloud, [MongoDB](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012480680)
* monitoring (see observability and monitoring)
* monolithic versus distributed microservices applications, [Designed as a Collection of Microservices](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#ix_mono_vs_distr_apps)-[Designed as a Collection of Microservices](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399015377384), [Data Management Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399014784728)
* MRU (most recently used) execution policy, [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012797592)
* MuleSoft, [Technologies for Implementing Service Composition Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013514904), [Technologies for Implementing API Management Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch07.html#idm46399010254696)
* MuleSoft Anypoint Platform, [Technologies for Implementing Service Composition Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013504168), [Technologies for Implementing API Management Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch07.html#idm46399010250664)
* Multiple-Receiver pattern, [Multiple-Receiver Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#ix_multi-receiver_ch2)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014519560), [Summary of Asynchronous Messaging Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014483720), [NATS](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014248776)

**N**

* NATS, [NATS](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014251624), [Summary of Message Broker Technologies](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011555304)
* Neo4j, [NoSQL Databases](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013337048)
* Netflix Conductor, [Technologies for Implementing Service Composition Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013500536)
* Newman, Sam
  + Building Microservices (O’Reilly), [Designed as a Collection of Microservices](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399015359816)
* nondeterministic finite-state machines, [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399011140760)
* nonoccurrence of event, Temporal Event Ordering pattern, [Detect nonoccurrence of event](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399011125688)
* NoSQL databases, [NoSQL Databases](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#ix_nosql_databases)-[NoSQL Databases](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013299656), [Apache Cassandra](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#ix_NoSQL_databases2)-[Amazon DynamoDB](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012443384), [Azure Cosmos DB](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012426184), [Summary of Technologies](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012402136)
* Nygard, Michael
  + Release It! (Pragmatic Bookshelf), [How it’s used in practice](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013992168)

**O**

* object-relational mapping (see ORM)
* observability and monitoring, [Observability and Monitoring](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399014844248)
  + communication patterns, [Observability and Monitoring](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014202840)
  + data management, [Observability and Monitoring](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#ix_observ_data_man)-[Observability and Monitoring](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012295112)
  + event-driven architecture, [Observability and Monitoring](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011501928)
  + patterns in practice, [Implementing Dynamic Management in a Cloud Environment](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch08.html#idm46399009998168)
  + and Saga pattern, [Considerations](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013540040)
  + stream-processing systems, [Observability and Monitoring](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010451560)
* online machine learning models, [Online machine learning models](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399011087192), [Continuous learning with data](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399011063704), [Considerations](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399011054216)
* online retail system, [Designed as a Collection of Microservices](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#ix_online_retail_ch1)-[Designed as a Collection of Microservices](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399015343064), [Building an Online Retail System](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch08.html#ix_online_retail_ch8)-[Customer and Partner Management](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch08.html#idm46399010104808)
* OpenAPI service definition specification, [Service definitions for RESTful services with OpenAPI and Consul](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014423704)
* orchestration patterns, [Orchestration Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#ix_orchest_patterns_ch5)-[Summary of Orchestration Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011674600)
  + API microgateways, [How it’s used in practice](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch07.html#idm46399010301752)
  + Mediator pattern, [Mediator Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#ix_orchest_mediator)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011750504), [Summary of Orchestration Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011663896)
  + Pipe and Filter pattern, [Considerations](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011764120), [Pipe and Filter Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#ix_orchest_pipe-filter)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011715656), [Summary of Orchestration Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011657832)
  + Priority Queue pattern, [Priority Queue Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#ix_orchest_priority-queue)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011678664), [RabbitMQ](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011623816)
* ORM (object-relational mapping), [Considerations](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012966056)

**P**

* partition tolerance, database, [NoSQL Databases](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013323336)
* partitioning of stream processing, Sequential Convoy pattern, [Scale stream-processing applications](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010959400)
* performance optimization patterns
  + data management, [Performance Optimization Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#ix_perf_opt_data_man)-[Summary of Performance Optimization Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012674728)
    - Caching pattern, [Caching Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#ix_perf_opt_caching)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012720664), [Summary of Performance Optimization Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012652344)
    - Data Locality pattern, [Data Locality Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#ix_perf_opt_data_locality)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012837352), [Summary of Performance Optimization Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012659000)
    - Materialized View pattern, [Materialized View Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#ix_per_opt_material_view)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012865336), [Summary of Performance Optimization Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012666152)
    - Static Content Hosting pattern, [Static Content Hosting Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#ix_perf_opt_static_host)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012678888), [Summary of Performance Optimization Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012645400)
  + stream-processing systems, [Scaling and Performance Optimization Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#ix_perf_opt_stream-proc)-[Summary of Scaling and Performance Optimization Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010749640)
    - Buffered Event Ordering pattern, [Buffered Event Ordering Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#ix_perf_opt_stream_proc_buffered)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010875976), [Considerations](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010776360)
    - Course Correction pattern, [Course Correction Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#ix_perf_opt_stream-proc_course_corr)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010822024), [Summary of Scaling and Performance Optimization Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010726600)
    - Sequential Convoy pattern, [Sequential Convoy Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#ix_perf_opt_seq_convoy)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010920120)
    - Watermark pattern, [Watermark Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#ix_perf_opt_stream-proc_watermark)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010755960)
* Periodic Snapshot State Persistence pattern, [Periodic Snapshot State Persistence Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#ix_periodic_snapshot)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010633352), [Summary of Reliability Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010574760), [Observability and Monitoring](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010449240)
* persisted state
  + Periodic Snapshot State Persistence pattern, [Considerations](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010684456)
  + Replay pattern, [Replay events when system state is not persisted](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010691064)
* persistence store, selecting for stream processing, [DevOps](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010440392)
* Pipe and Filter pattern, [Considerations](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011763720), [Pipe and Filter Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#ix_pipe-filter_pattern)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011714408), [Summary of Orchestration Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011659880)
* pod, Kubernetes, [Kubernetes](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399015043960)
* point-to-point asynchronous messaging, [Single-Receiver Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#ix_point-point_async)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014561224)
* Polling pattern, [Polling Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#ix_polling_pattern)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011992712), [Summary of Event-Delivery Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011899160)
* prebuilt machine learning models, [Prebuilt machine learning models](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399011090536), [Predict based on prebuilt machine learning models](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399011074440), [Considerations](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399011058872)
* prediction (see Machine Learner pattern)
* prefetching of data, Caching pattern, [Prefetch data to improve data-retrieval time](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012779784)
* Priority Queue pattern, [Priority Queue Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#ix_priority-queue_pattern)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011679624), [RabbitMQ](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011618536)
* privacy issue, data management, [Security](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012329192)
* Producer-Consumer pattern, [Producer-Consumer Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#ix_prod_consum_pattern)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399012148936), [Summary of Event-Delivery Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011923704)
* proprietary data stores, abstract, [Expose abstract legacy/proprietary data stores](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013194712)
* Protobuf, [Event Schema](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399012202104)
* Protocol Buffers, [How it’s used in practice](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014647208), [How it’s used in practice](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014388664), [gRPC](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014309864)
* protocol switching, [Protocol switching](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399011404216)
* Publisher-Consumer pattern, [Apache ActiveMQ](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011630792)
* Publisher-Subscriber pattern, [Multiple-Receiver Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014548968), [Considerations](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012743080), [Publisher-Subscriber Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#ix_pub-sub_pattern)-[Related pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399012084904), [Summary of Event-Delivery Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011918264), [Amazon SNS](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011598408)

**Q**

* Quarkus, [Technologies for Implementing Service Composition Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013523336)
* query data set, CQRS pattern, [Use different domain models for command and query](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012976200)
* query-based interaction, [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014703432)
* queues as buffers in asynchronous event communication, [Process each event by a single application](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399012165928), [Considerations](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399012151960)

**R**

* RabbitMQ, [DevOps](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014186216), [RabbitMQ](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011626072), [Summary of Message Broker Technologies](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011564680)
* range-based data sharding, [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013045864)
* RDB (Redis Database Backup), [Redis](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012460360)
* RDBMSs (relational database management systems), [Relational Database Management Systems](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012502536), [Summary of Technologies](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012407208)
* reactive microservices, [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013631944)
* read committed transaction isolation, [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012600536)
* read uncommitted transaction isolation, [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012603752)
* read-through cache operation, [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012817352)
* real-time systems, [Data Architecture](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013445784)
* reconciliation loops, [Automate the Development Life Cycle](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399014989224)
* Red Hat 3scale API Management, [Technologies for Implementing API Management Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch07.html#idm46399010258760)
* Red Hat Fuse, [Technologies for Implementing Service Composition Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013515528)
* Redis, [NoSQL Databases](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013363272), [Redis](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012469656), [Summary of Technologies](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012385096)
* reference architecture, [Reference Architecture for Cloud Native Apps](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#ix_ref_archit_ch1)-[Reference Architecture for Cloud Native Apps](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399014745400)
* registry repository API, [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399014050408), [Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399014015336)
* regrouping of events, [Considerations](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010938568)
* relational databases, [Relational Databases](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013397128), [Relational Database Management Systems](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012501528), [Google Cloud Spanner](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012424312), [Summary of Technologies](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012408648)
* relaxing dependency with Caching pattern, [Achieve high availability by relaxing the data store dependency](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012774168)
* Release It! (Pragmatic Bookshelf), [How it’s used in practice](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013997064)
* reliability patterns, [Reliability Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#ix_reliability_patterns_ch4)-[Summary of Transaction Reliability Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012567544), [Reliability Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#ix_reliability_patterns_ch6)-[Summary of Reliability Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010589128)
  + Periodic Snapshot State Persistence pattern, [Periodic Snapshot State Persistence Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#ix_reliability_periodic_snapshot)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010638104), [Summary of Reliability Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010573624), [Observability and Monitoring](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010446648)
  + Replay pattern, [Replay Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#ix_reliability_patterns_replay)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010676056), [Summary of Reliability Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010581752)
  + Stream Join pattern, [Considerations](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399011167624)
  + Transaction pattern, [Transaction Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#ix_reliability_patterns_trans)-[Summary of Transaction Reliability Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012569720)
  + Two-Node Failover pattern, [Two-Node Failover Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#ix_reliability_two-node)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010594408)
  + Windowed Aggregation pattern, [How it’s used in practice](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399011248792)
* reordering of events from same sources, [Considerations](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010883976)
* repeatable reads transaction isolation, [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012606200)
* Replay pattern, [Replay Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#ix_replay_pattern_ch6)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010674744), [Summary of Reliability Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010579976)
* replaying of events with Event Sourcing, [Replay events](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011861816)
* ReplicaSet abstraction, Kubernetes, [Kubernetes](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399015039000)
* Request Callback pattern, [Request Callback Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#ix_req_callback_pattern)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011941048), [Summary of Event-Delivery Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011897288)
* request queue, Asynchronous Request-Reply pattern, [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014508088)
* Request-Reply pattern, [Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014684648), [Asynchronous Request-Reply Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#ix_req-reply_pattern2)-[Summary of Asynchronous Messaging Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014477208), [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011783496)
* Request-Response pattern, [Request-Response Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#ix_req-resp_pattern)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014680104), [Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014624008), [How it’s used in practice](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014503336), [RESTful Services](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#ix_req-resp_pattern_tech)-[gRPC](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014290616)
* resiliency, with Material View pattern, [Considerations](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012875096)
* Resilient Connectivity pattern, [Resilient Connectivity Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#ix_resilient_connectivity_pattern)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013925416), [Summary of Connectivity Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013710632), [Achieve high availability by relaxing the data store dependency](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012772360)
* resources, REST, [RESTful Services](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014347688)
* response queue, Asynchronous Request-Reply pattern, [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014497880)
* REST API, [Control Plane for Dynamic Management](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399014848488)
* REST/Open API, [API Management Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch07.html#idm46399010396680)
* RESTful services, [How it’s used in practice](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014697096), [How it’s used in practice](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014654808), [How it’s used in practice](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014433064), [RESTful Services](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014344104), [Summary of Synchronous Messaging Technologies](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014285128)
* retry logic, in Resilient Connectivity pattern, [Retry](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013977720)
* RIAs (right internet applications), [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013129816)
* RPC (Remote Procedure Calls) API, [Control Plane for Dynamic Management](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399014847640), [How it’s used in practice](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014655656), [How it’s used in practice](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014434088), [gRPC](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#ix_RPC_pattern2)-[gRPC](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014294808), [Sidecarless gRPC services in Google Traffic Director](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013784136)
* RPC (Remote Procedure Calls) pattern, [Remote Procedure Calls Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#ix_RPC_pattern_ch2)-[Summary of Synchronous Messaging Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014618984)
* running in dynamic environment, automated deployment, [Running in a Dynamic Environment](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399014853240)

**S**

* S3, Amazon, [Filesystem Storage](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013284904), [Amazon S3](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012438200), [Summary of Technologies](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012368856)
* Saga Execution Coordinator (see SEC)
* Saga log, [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013563608)
* Saga pattern, [Saga Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#ix_saga_pattern)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013532120), [Summary of Service Composition Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013477368), [Considerations](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012579736)
* scalability
  + aggregation operations, [Considerations](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399011224472)
  + autoscaling, [DevOps](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011471272)
  + data scaling patterns, [Data Scaling Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#ix_scalability_data_scaling)-[Summary of Data Scaling Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012949032)
  + NoSQL’s capability, [NoSQL Databases](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013375000)
  + relational database limitation in, [Relational Databases](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013380104)
  + with Service Abstraction pattern, [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399014096152)
  + stream-processing systems, [Scaling and Performance Optimization Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#ix_scalability_stream-proc)-[Summary of Scaling and Performance Optimization Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010747192)
    - Buffered Event Ordering pattern, [Buffered Event Ordering Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#ix_scalability_stream_proc_buffer)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010872296), [Considerations](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010774792)
    - Course Correction pattern, [Course Correction Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#ix_scalability_stream-proc_course_corr)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010827064)
    - Sequential Convoy pattern, [Sequential Convoy Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010977768), [Summary of Scaling and Performance Optimization Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010736840)
    - Watermark pattern, [Watermark Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#ix_scalability_stream-proc_watermark)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010754680)
  + Transformation pattern consideration, [Considerations](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399011399160)
* scatter and gather, Stream Join pattern, [Scatter and gather](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399011179016)
* schema and schema registry, asynchronous communication, [Service Definition in Asynchronous Communication](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#ix_schema_schema_registry)-[Considerations](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014375512)
* schema on read principle, databases, [NoSQL Databases](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013378104)
* schema on write principle, databases, [Relational Databases](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013394200)
* schemas, event, [Event Schema](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399012210680), [DevOps](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011462520)
* SEC (Saga Execution Coordinator), [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013565240)
* security
  + accessing nonsensitive data in secure systems, [Provide access to nonsensitive data hosted in secure systems](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012885272)
  + Caching pattern, [Considerations](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012738744)
  + communication patterns, [Security](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014225064)
  + data management, [Security: Vault Key Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#ix_security_data_man)-[Summary of the Vault Key Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012519880), [Security](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012334568)
  + event-driven architecture, [Security](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011511720)
  + patterns in practice, [Implementing Dynamic Management in a Cloud Environment](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch08.html#idm46399009999960)
  + stream-processing systems, [Security](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010463288)
  + Vault Key pattern, [Considerations](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012693288), [Security: Vault Key Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#ix_security_vault_key)-[Summary of the Vault Key Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012529192)
* segregation of data, Data Sharding pattern, [Segregate data to improve data-retrieval time](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013040872)
* segregation of duties, Pipe and Filter pattern, [Provide segregation of duties](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011727736)
* selective delivery of events, [Selectively deliver events to subscribers](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399012112600)
* semi-structured data, [Types and Forms of Data](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013412792)
* sequence numbers, Buffered Event Ordering pattern, [Considerations](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010940488), [Considerations](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010882888)
* sequence of event occurrence, Temporal Event Ordering pattern, [Detect sequence of event occurrence](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399011131320)
* Sequential Convoy Pattern, [Sequential Convoy Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#ix_seq_convoy)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010926456)
  + Temporal Event Ordering pattern, [Considerations](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399011116600)
* Sequential Convoy pattern, [Summary of Scaling and Performance Optimization Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010738008)
* serializable transaction isolation, [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012608504)
* server or client stubs, [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014659336)
* Server-Side Mashup pattern, [Composite Data Services Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#ix_server-side_mashup)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013153512), [Summary of Data Composition Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013095816), [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012908872)
* server-side service discovery, [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399014041272)
* serverless functions, containerization, [Serverless functions](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399015027432)
* Service Abstraction pattern, [Service Abstraction Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#ix_serv-abstract_pattern)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399014060216), [Summary of Connectivity Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013718008)
* Service abstraction, Kubernetes, [Kubernetes](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399015040152), [Service-led interactions](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399014893848), [Kubernetes services](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#ix_serv-abstract_Kube)-[Considerations](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399014072552)
* Service Bus Queues, [Azure Service Bus Queues](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011600328), [Summary of Message Broker Technologies](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011534120)
* service chaining, [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013682312)
* Service Choreography pattern, [Service Choreography Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#ix_serv_choreog_ch3)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013603832), [Summary of Service Composition Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013481944)
* service composition patterns, [Service Composition Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#ix_serv_comp_ch3)-[Summary of Service Composition Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013493016)
  + Request-Response in, [Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014622856)
  + RPC pattern in, [Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014635160)
  + Saga pattern, [Saga Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#ix_serv_comp_saga_pattern)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013530872), [Summary of Service Composition Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013474568), [Considerations](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012581752)
  + Service Choreography pattern, [Service Choreography Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#ix_serv_comp_choreog)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013611512), [Summary of Service Composition Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013479496)
  + Service Orchestration pattern, [Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014693976), [Service Orchestration Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#ix_serv_comp_orchest)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013647272), [Summary of Service Composition Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013484600)
  + technologies of, [Technologies for Implementing Service Composition Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013526632)
* Service Connectivity pattern, [Service Connectivity Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#ix__service-connectivity)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399014110136), [Technologies for Implementing Service Connectivity Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#ix_serv_connect_ch3)-[Technologies for Implementing Service Connectivity Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013733192)
* service definitions, [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014665272), [Service Definition Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#ix_serv-def_patterns)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014359544)
* service discovery, [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014446584), [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399014092712), [Service Registry and Discovery Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399014059000), [Summary of Connectivity Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013711976)
* Service Mesh pattern, [Service Mesh Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#ix_serv-mesh_pattern)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013807768), [Technologies for Implementing Service Connectivity Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013749480), [Summary of Connectivity Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013695368)
* Service Mesh Sidecar pattern, [Service Mesh Sidecar as an API Gateway Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch07.html#ix_service_mesh_sidecar2)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch07.html#idm46399010266728)
* Service Orchestration pattern, [Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014691608), [Service Orchestration Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#ix_serv_orchest_ch3)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013645928), [Summary of Service Composition Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013486088)
* service registry, [Service Definitions in Synchronous Communication](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#ix_serv_registry)-[Service definitions with AsyncAPI](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014372104)
* Service Registry and Discovery pattern, [Service Registry and Discovery Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#ix_serv-reg-discover_pattern)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399014016328), [Summary of Connectivity Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013713176)
* service-led interactions, [Service-led interactions](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399014900392)
* service-oriented architecture (see SOA)
* services, connecting, patterns in practice, [Connecting Services](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch08.html#ix_connect_services)-[Connecting Services](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch08.html#idm46399010058088)
* sharding events, [Considerations](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399011225048)
* shared cache, [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012807768)
* Siddhi, [Siddhi](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010550728), [Summary of Stream-Processing Technologies](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010506616)
* sidecar bridge, [Sidecar bridge](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013887608)
* sidecar injection, [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013841176)
* Sidecar pattern, [Sidecar Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#ix_sidecar_pattern)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013870424), [Summary of Connectivity Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013702920)
* sidecar proxies, [Sidecar proxy](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013909192), [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013852424), [Service Mesh Sidecar as an API Gateway Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch07.html#ix_sidecar_proxies)-[Considerations](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch07.html#idm46399010264744)
* Sidecarless Service Mesh pattern, [Sidecarless Service Mesh Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#ix_sidecarless_serv-mesh)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013774616), [Summary of Connectivity Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013693096)
* Single-Receiver pattern, [Single-Receiver Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#ix_single-receiver_ch2)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014563608), [Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014486008), [AMQP](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014260024), [NATS](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014245352)
* Siriwardena, Prabath
  + Microservices for the Enterprise, [Security](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014212104)
* smart endpoints, microservices, [Designed as a Collection of Microservices](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399015354424)
* snapshots, importance in DevOps for stream processing, [DevOps](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010441272)
* SOA (service-oriented architecture), [Designed as a Collection of Microservices](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#ix_SOA_ch1)-[Designed as a Collection of Microservices](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399015363528), [Compositions](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399014794168)
* Software Delivery Guide (Fowler), [DevOps](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014177976)
* Solr, [Filesystem Storage](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013289576)
* sorting events, [Sort and distribute events](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011775576)
* Spark, [Apache Spark](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010552680), [Summary of Stream-Processing Technologies](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010495928)
* splitting events into subevents, [Split events into multiple subevents](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011771704)
* Spring Boot, [Technologies for Implementing Service Composition Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013524456)
* SQL (Structured Query Language) databases, [Relational Databases](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013396056), [Relational Database Management Systems](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012500008), [Amazon S3](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012429720)
* standard (store-backed) message brokers, [Message Broker Categories](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399012225720)
* state data type, [Types and Forms of Data](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013424792)
* state handling, testing stream-processing patterns for, [Testing](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010470504)
* state management patterns, [State Management Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#ix_state_man_patterns)-[Summary of State Management Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011822792)
* state, recreating with Event Sourcing, [Re-create application state](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011866344)
* stateful applications, [Types and Forms of Data](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013418904)
  + (see also data management)
* stateless applications, [Types and Forms of Data](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013421768)
  + (see also compositions; connectivity patterns)
* Static Content Hosting pattern, [Static Content Hosting Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#ix_static_content_host)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012680024), [Summary of Performance Optimization Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012642696)
* static content loading, improving with Caching Pattern, [Improve static content loading](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012788296), [Considerations](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012756744)
* Store and Forward pattern, [Store and Forward Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#ix_store-forward_pattern)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399012020552), [Summary of Event-Delivery Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011907800)
* Stream Join pattern, [Stream Join Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#ix_stream_join_pattern)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399011151960), [Summary of Streaming Data Processing Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399011007928)
* stream versus event, [Stream-Processing Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399014767880)
* stream-processing systems, [Stream-Processing Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399014769240), [Stream-Processing Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#ix_stream-process_ch6)-[Summary](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010413784)
  + in data architecture, [Data Architecture](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013444344)
  + data processing patterns, [Streaming Data Processing Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#ix_stream_proc_data_proc)-[Summary of Streaming Data Processing Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399011040152)
    - Filters and Thresholds pattern, [Filters and Thresholds Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#ix_stream_proc_data_filter-thresh)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399011345192), [Summary of Streaming Data Processing Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399011030072)
    - Machine Learner pattern, [Machine Learner Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#ix_stream-proc_data_proc_mach)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399011046760)
    - Stream Join pattern, [Stream Join Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#ix_stream-proc_data_proc_join)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399011165416), [Summary of Streaming Data Processing Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399011006888)
    - Temporal Event Ordering pattern, [Temporal Event Ordering Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#ix_stream_proc_data_proc_temp_order)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399011099416), [Summary of Streaming Data Processing Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399011000536), [Synchronize events generated from nonsynchronized sources](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010786488)
    - Transformation pattern, [Transformation Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#ix_stream-proc_data_proc_trans)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399011396088), [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399011250664), [Summary of Streaming Data Processing Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399011028456)
    - Windowed Aggregation pattern, [Windowed Aggregation Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#ix_stream-proc_data_proc_wind)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399011204280)
  + DevOps, [DevOps](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010438504)
  + observability and monitoring, [Observability and Monitoring](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010453016)
  + patterns in practice, [Using Stream Processing](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch08.html#idm46399010027064)
  + reference architecture, [Reference Architecture for Cloud Native Apps](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399014742664)
  + reliability patterns, [Reliability Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#ix_stream_proc_reliability)-[Summary of Reliability Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010590376), [Observability and Monitoring](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010448216)
  + scaling and performance optimization patterns, [Scaling and Performance Optimization Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#ix_stream-process_scaling-perf)-[Summary of Scaling and Performance Optimization Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010750216)
    - Buffered Event Ordering pattern, [Buffered Event Ordering Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#ix_stream_proc_perf_opt_buffer)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010871112), [Considerations](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010772296)
    - Course Correction pattern, [Course Correction Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#ix_stream-proc_scaling_course_corr)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010825096), [Summary of Scaling and Performance Optimization Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010728392)
    - Sequential Convoy pattern, [Sequential Convoy Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#ix_stream_proc_scaling_seq_convoy)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010927320), [Summary of Scaling and Performance Optimization Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010740184)
    - Watermark pattern, [Watermark Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#ix_stream_proc_scalability_watermark)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010757512)
  + security, [Security](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010464776)
  + technologies of, [Technologies](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#ix_stream-proc_tech_of)-[Summary of Stream-Processing Technologies](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010520232)
  + testing, [Testing](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010474216)
* structured data, [Types and Forms of Data](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013415624)
* Structured Query Language (see SQL)
* support for filtering and thresholds, [Considerations](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399011359064)
* synchronizing events, Watermark pattern, [Synchronize events generated from event sources that are time synchronized](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#ix_sync_events_watermark)-[Synchronize events generated from nonsynchronized sources](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010784184)
* synchronous communication, [Communication Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399014820584), [Communication Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014722616), [Synchronous Messaging Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#ix_sync_com_ch2)-[Summary of Synchronous Messaging Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014616088)
  + DevOps, [DevOps](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014183416)
  + observability and monitoring, [Observability and Monitoring](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014193272)
  + Request-Response pattern, [Request-Response Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#ix_sync-com_req-resp)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014678856), [Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014625480), [How it’s used in practice](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014502328), [RESTful Services](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#ix_sync_com_req-resp_tech)-[gRPC](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014289320)
  + RPC pattern, [Remote Procedure Calls Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#ix_sync_com_RPC)-[Summary of Synchronous Messaging Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014617960)
  + security, [Security](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014221912)
  + Service Connectivity pattern, [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399014149032)
  + service definition patterns, [Service Definition Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#ix_sync-com_serv-def)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014424280)
  + technologies of, [Technologies to Implement Synchronous Messaging Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#ix_sync-com_tech_implem)-[gRPC](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014298824)
  + testing, [Testing](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014232392)
* system metrics, data stores, [Observability and Monitoring](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012303016)

**T**

* task execution order, ensuring with mediator microservices, [Ensure task execution order](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011770008)
* TCP (Transmission Control Protocol), [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014646360), [WebSocket](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014323256)
* Temporal Event Ordering pattern, [Temporal Event Ordering Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#ix_temp_event_order)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399011102440), [Summary of Streaming Data Processing Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010998792)
* temporal queries, running with Event Sourcing, [Run temporal queries](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011848600)
* testing, [Independent testing](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399014924840)
  + communication patterns, [Testing](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014236264)
  + data management, [Testing](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#ix_testing_data_man)-[Testing](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012339656)
  + event-driven architecture, [Testing](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#ix_testing_eda)-[Testing](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011514216)
  + stream-processing systems, [Testing](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010472872)
* threshold count, circuit breaker, [Circuit breaker](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013952056)
* threshold for alerting, applying to events, [Scenario: Apply a threshold for alerting](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399011361752)
* Thrift, [gRPC](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014300536)
* time sliding window, [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399011329592), [Aggregate events over time](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399011244168)
* time-bounded patterns, testing in stream processing, [Testing](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010469032)
* time-outs, [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014701880)
  + cache, [Reduce data store contention](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012782344), [Considerations](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012749288)
  + circuit reset, [Circuit breaker](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013945432)
  + in Resilient Connectivity pattern, [Time-out](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013986760)
* timestamps, Buffered Event Ordering pattern, [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#ix_timestamps_buffered)-[Considerations](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010879752)
* TLS (Transport Layer Security), [Security](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014216968)
* top-down API management, [How it’s used in practice](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch07.html#idm46399010331752)
* topic
  + in Multiple-Receiver pattern, [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014541736)
  + in Publisher-Subscriber pattern, [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399012133944)
* transaction manager, Saga pattern, [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013584056)
* Transaction pattern, [Transaction Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#ix_transaction_pattern)-[Summary of Transaction Reliability Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012575960)
* Transformation pattern, [Transformation Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#ix_transform_pattern)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399011387592), [Summary of Streaming Data Processing Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399011029608)
  + Windowed Aggregation pattern, [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399011251528)
* Transmission Control Protocol (see TCP)
* Transport Layer Security (see TLS)
* Twelve-Factor App, [Methodology for Building Cloud Native Apps](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399014969656)
* Two-Node Failover pattern, [Two-Node Failover Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#ix_two-node_failover)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010612600)
* two-phase commit (2PC) protocol, [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013580536)

**U**

* unit tests, microservice, [Independent testing](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399014918728)
* unstructured data, [Types and Forms of Data](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013406408)
* updating caches for optimized performance, [Considerations](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#ix_update_cache)-[Considerations](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012735064)
* updating results with new information, Course Correction pattern, [Update results with new information](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#ix_updating_course_corr)-[Correct previous decisions](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010839832)

**V**

* Vault Key pattern, [Considerations](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012690200), [Security: Vault Key Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#ix_vault_key)-[Summary of the Vault Key Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012529976)
* vertical data sharding, [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013061048)
* virtual machines, [Virtual machines](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399015017512)
* VMs (virtual machines) versus container runtime engine, [What are containers?](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399015089080), [Virtual machines](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399015023704)

**W**

* watermark event, [Watermark Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010810520)
* Watermark pattern, [Watermark Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#ix_watermark_pattern)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010759416), [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399010648360)
* webhooks, [Webhooks](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#ix_webhooks)-[Considerations](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011961704)
* WebSocket protocol, [WebSocket](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014317176), [Summary of Synchronous Messaging Technologies](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014284360), [WebSockets](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#ix_websocket)-[Considerations](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011959448)
* WebSub, [Webhooks](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399011963992)
* Windowed Aggregation pattern, [Windowed Aggregation Pattern](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#ix_windowed_aggregat)-[Related patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399011203608), [Summary of Streaming Data Processing Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch06.html#idm46399011017448)
* write-through cache operation, [How it works](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012812072)
* WSO2 API Manager, [Technologies for Implementing API Management Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch07.html#idm46399010257688)
* WSO2 Micro Integrator, [Technologies for Implementing Service Composition Patterns](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch03.html#idm46399013513320)

**X**

* XA transactions, [Combine operations across multiple systems](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#ix_xa_trans)-[Considerations](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399012566296)
* XML (Extensible Markup Language), [RESTful Services](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch02.html#idm46399014338072), [Types and Forms of Data](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013409976), [NoSQL Databases](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013347896), [Event Schema](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch05.html#idm46399012205176)

**Y**

* YAML, [Kubernetes](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch01.html#idm46399015033112), [Types and Forms of Data](https://learning.oreilly.com/library/view/design-patterns-for/9781492090700/ch04.html#idm46399013408664)