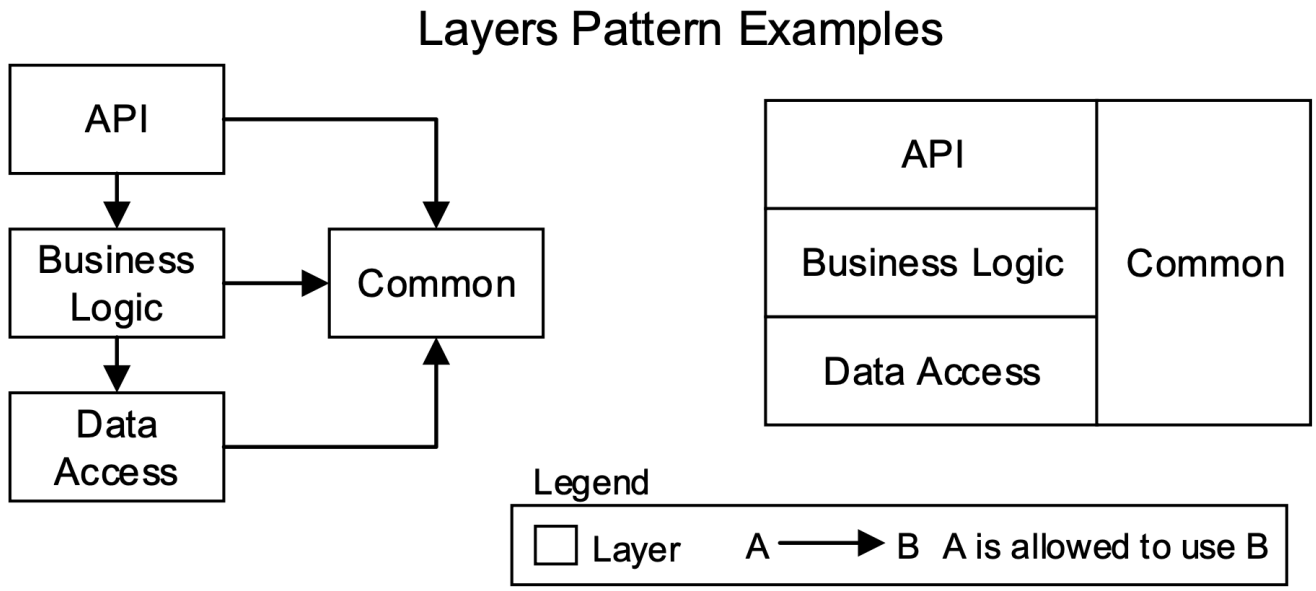
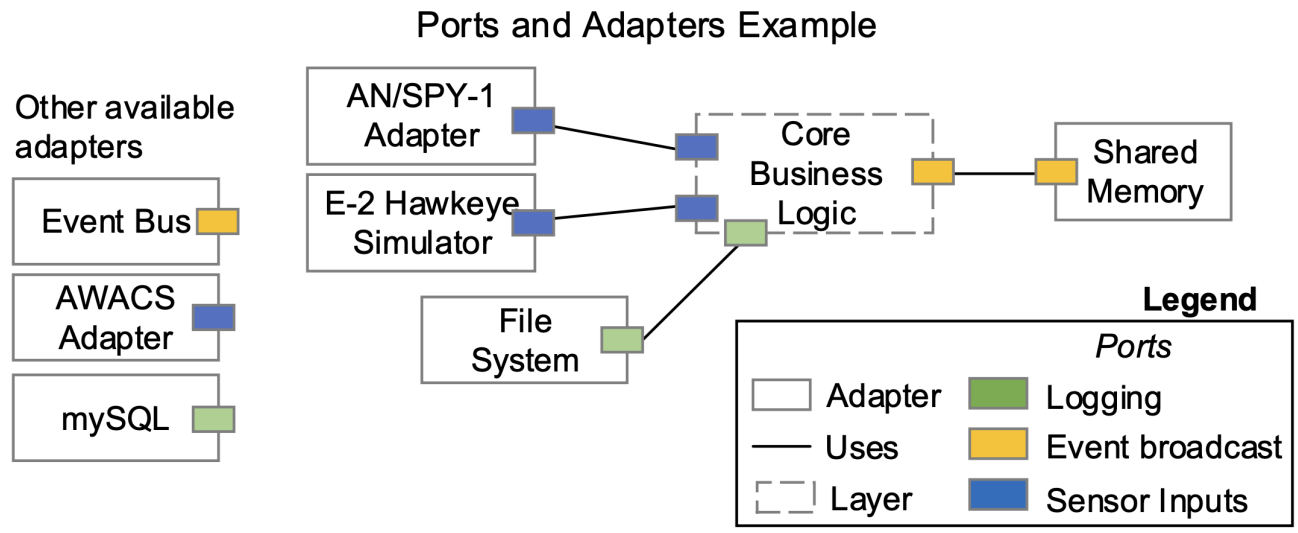
Chapter 7 - Create a Foundation with Patterns

architecture pattern - a reusable solution to a specific problem

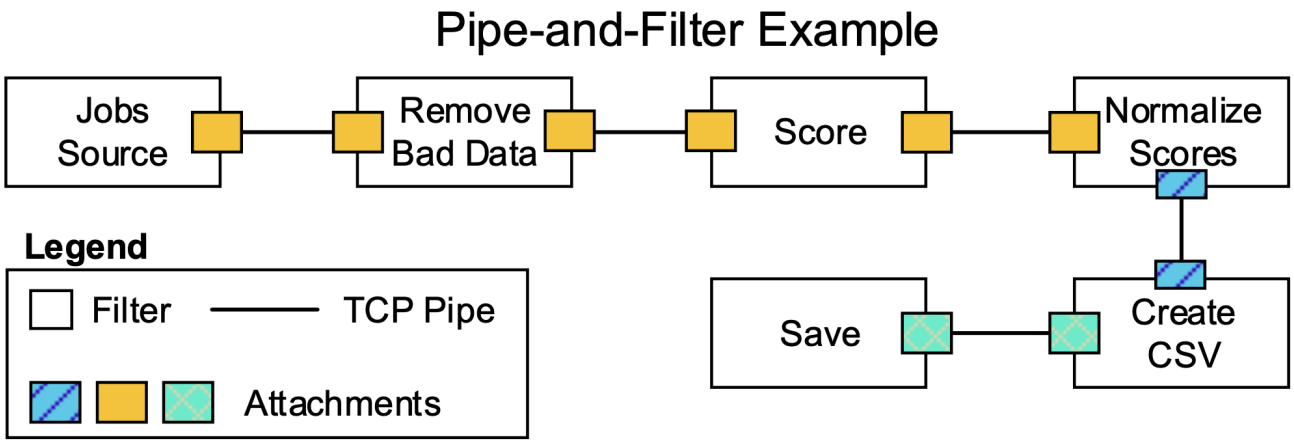
Layers Pattern - Partitioning code into distinct and independent layers organized around a specific set of related concerns



Ports and Adapters Pattern - isolates core business logic so it can be used in a variety of contexts and tested in isolation from components that provide data and events

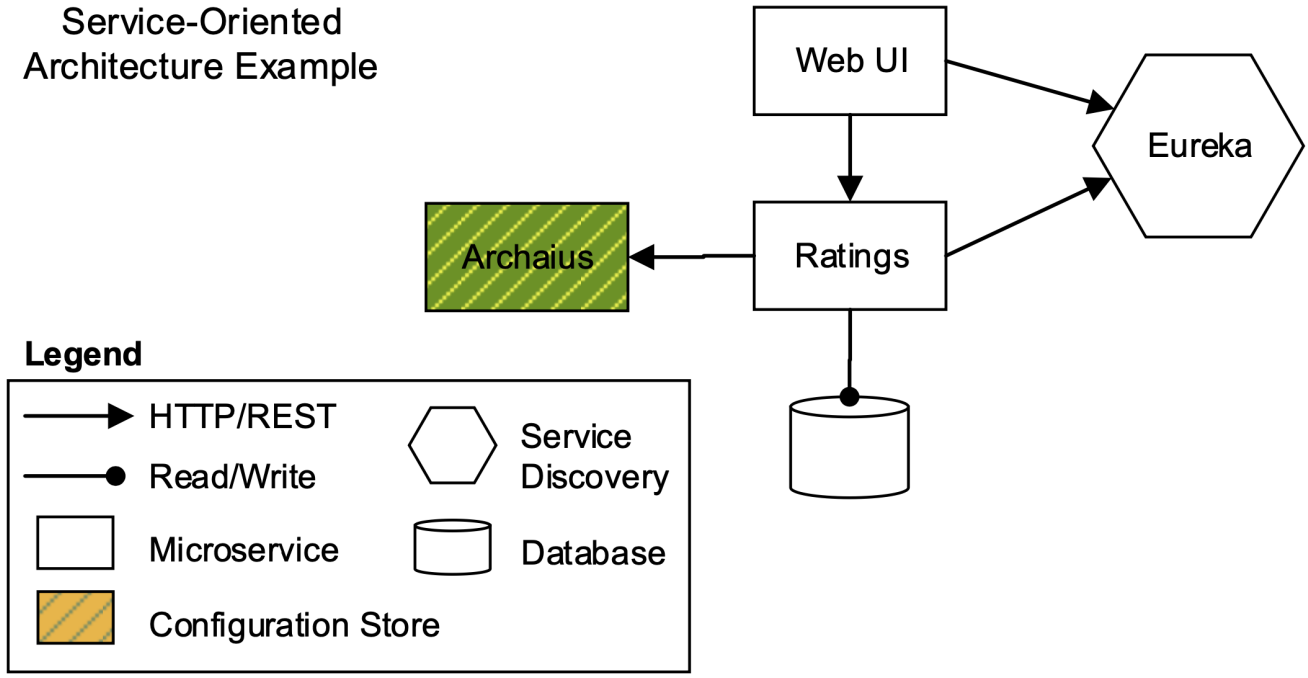


Pipe-and-Filter Pattern - each component called a filter is responsible for a single transformation or data operation. Data is streamed from one filter to the next as quickly as possible, and data operations occur in parallel. Loosely coupled filters can be reused and combined in different ways to create new pipelines

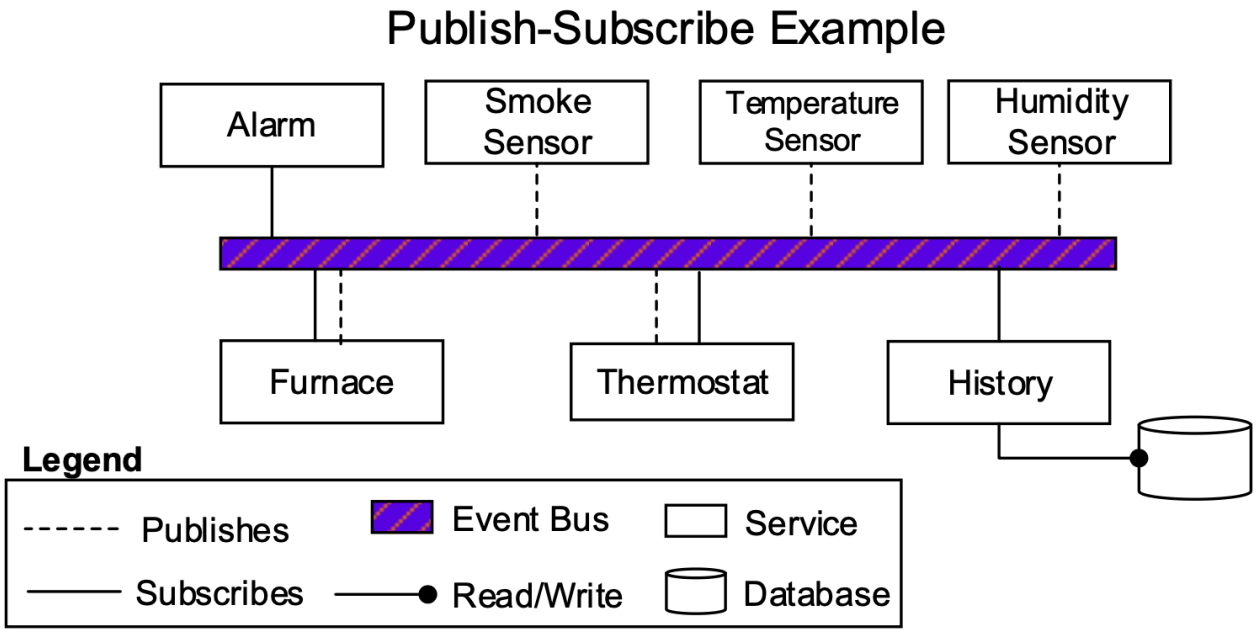


batch sequential pattern - similar to pipe-and-filter. Stages of a batch-sequential system operate in turn, one at a time instead of in parallel like in a pipe-and-filter system

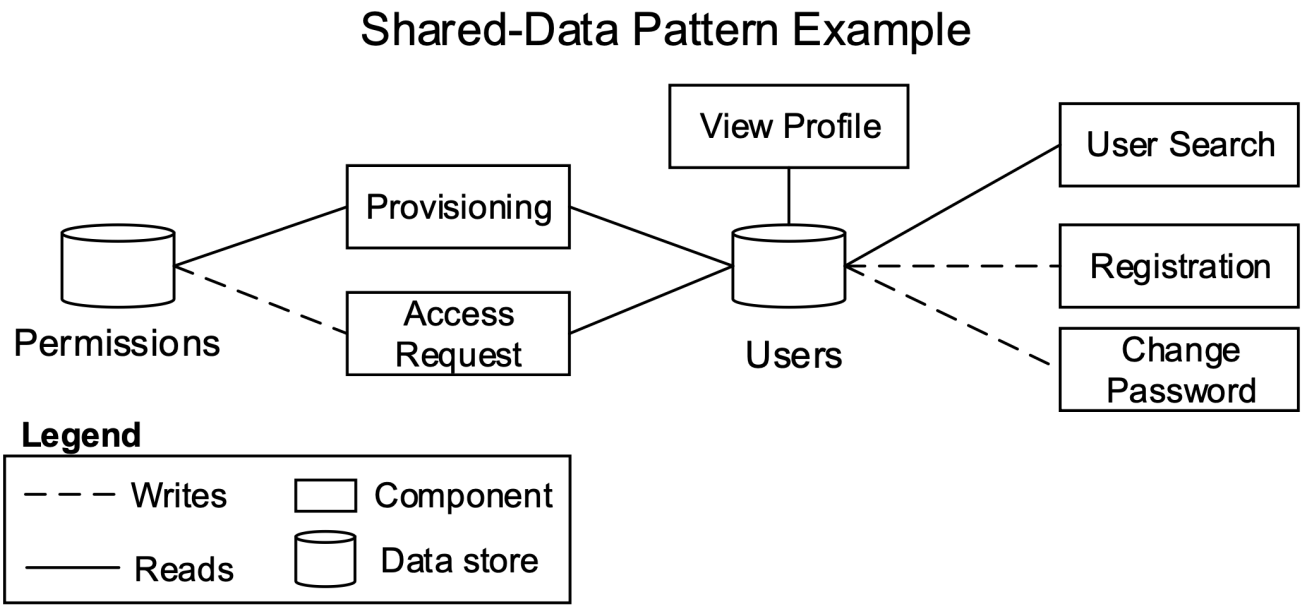
Service-Oriented Architecture Pattern - independent components are implemented as services, which provide specific functionality



Publish-Subscribe Pattern - producers and consumers exist independently and unaware of one another. Consumers subscribe to events published by producers



Shared-Data Pattern - multiple components access data through a common data store. No single component is entirely responsible for the data or data store

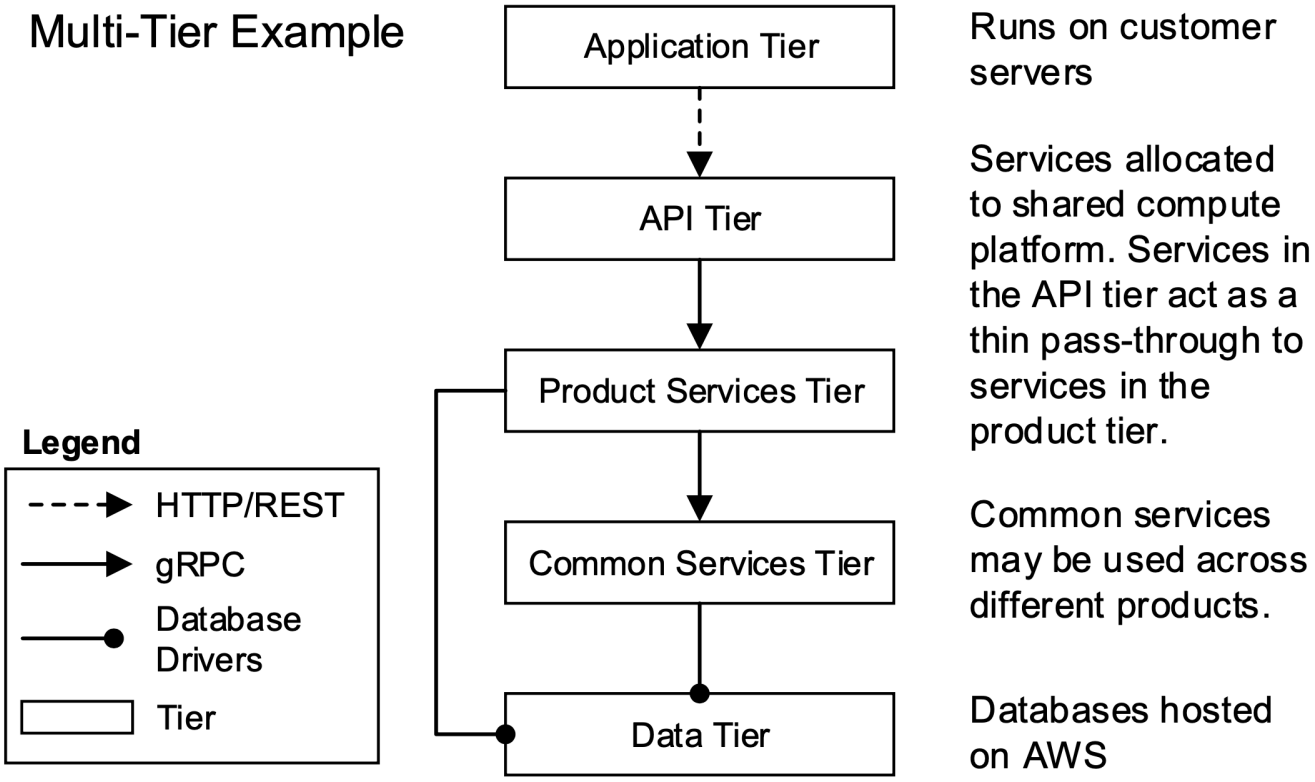


Open Source Contribution Pattern - teams are given responsibility for developing specific architectural components but are not expected to be the only contributing developers. For this pattern to be successful, teams must know they are responsible for specific components and have a firm understanding of where their components fit within the larger context

Big Ball of Mud Pattern - no defined elements or relations

Architectural mismatch - occurs when the assumptions made about how a component will be used conflicts with that component's current use

Multi-Tier Pattern - runtime structures are organized into logical groups. These logical groups may be allocated to specific physical components, such as a server or cloud platform. similar to the layers pattern



Center of Competence Pattern (CoC) - a team of experts is charged with defining patterns, establishing best practices, developing support tools, and providing education for a subset of the architecture

CoC Team	Responsibility Area
Job Scheduling Use Case	Develop a framework for the job scheduling use case and create tools so teams can instantiate the framework on clusters themselves.
Performance	Consult with teams about load and performance testing, provide tools for testing and data collection, collect and organize data sets and other testing assets.
Database Technologies	Consult with teams to select supported database technologies appropriate to use case, maintain tools for provisioning databases, create or distribute training materials.
Core Platform	Maintain common container management system, provide supported Docker base images, create tools for day-to-day tasks such as log aggregation and health checks.