COM S 362
Object-Oriented Analysis & Design

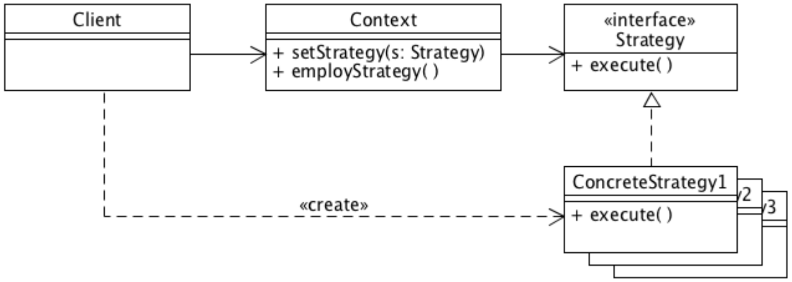
Layered and Pipeline
Architectures

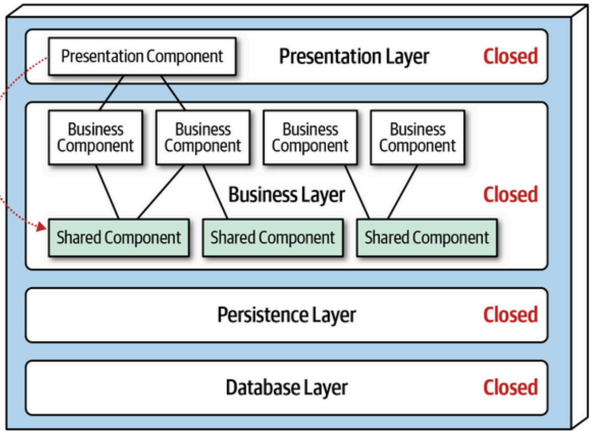
Reading

Mark Richards and Neal Ford. Fundamentals of Software Architecture: An Engineering Approach, First Edition, 2020.

- Chapter 9: Foundations
- Chapter 10: Layered Architecture
- Chapter 11: Pipeline Architecture

Design Patterns vs Architectural Patterns

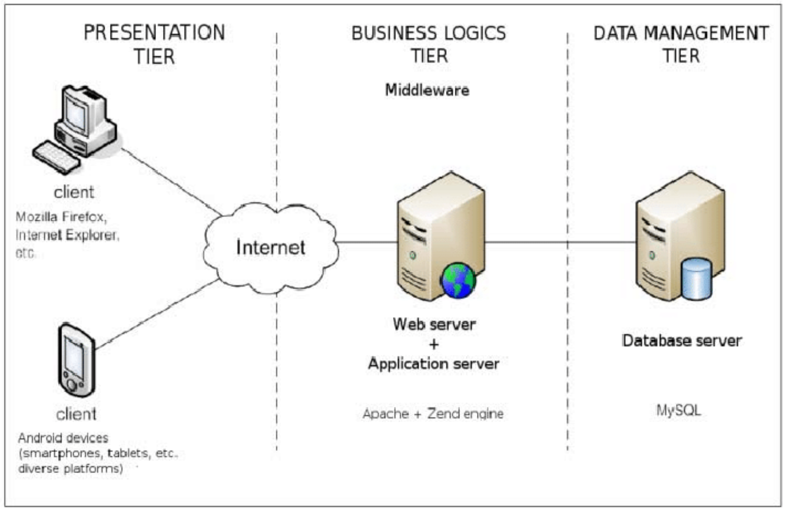




- Design patterns provide templates for solving common design problems that describe relationships between classes and objects
- Architectural patterns provide ways for organizing relationships between components
- Both provide a named relationships between software entities, that is valuable for communicating, documenting and reasoning about software
- Architectural patterns are not strictly object-oriented

History of Architecture

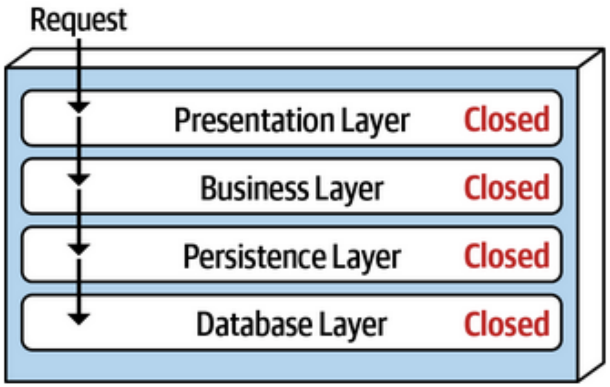
- Big Ball of Mud – an anti-pattern, the lack of a clear architecture
- Unitary Architecture – describes a non-distributed (e.g., single process) architecture
- Client/Server (aka two-tier or frontend/backend) – simplest organization of a distributed system, clients send messages making requests, servers listen and reply with response
- Browser + Web Server – common use of client/server, browser is (was?) thin compared to web server
- Three-Tier Architecture – early division of Internet services into client, application tier and data tier


PRESENTATION TIER
BUSINESS LOGICS TIER
DATA MANAGEMENT TIER
Middleware
client
Mozilla Firefox, Internet Explorer, etc.
Internet
client
Android devices (smartphones, tablets, etc. diverse platforms)
Web server + Application server
Apache + Zend engine
Database server
MySQL

Architectures

- Monolithic
  - Layered architecture
  - Pipeline architecture
  - Microkernel architecture
- Distributed
  - Service-based architecture
  - Event-driven architecture
  - Space-based architecture
  - Service-oriented architecture
  - Microservices architecture

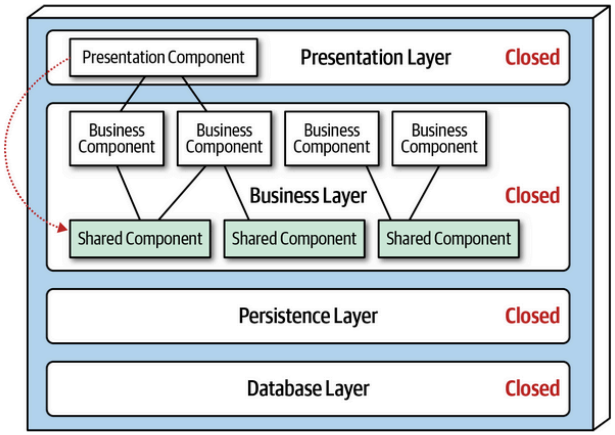
Layered Architecture

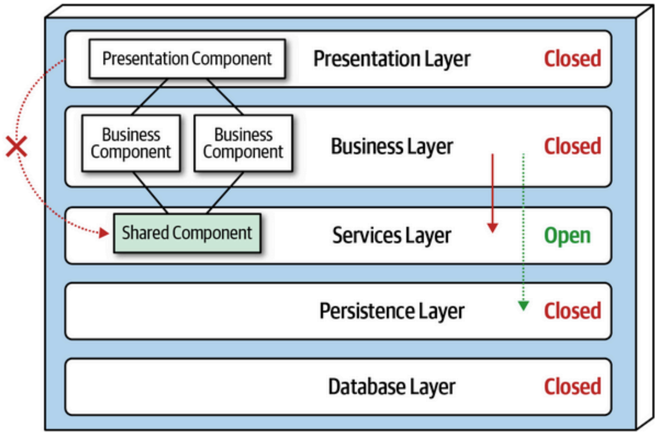


- **Layered** or **n-tiered** – components organized into layers, requests passed down through the layers and responses return back up through the layers
- Simple and widely used architecture
- Common layers: presentation, business, persistence, database
- Follows design principle of Separation of Concerns
- Seen in GRASP controller pattern

Layers

- Layers can be closed or open
  - **Closed** – all requests must pass through the layer
  - **Open** – the layer can be bypassed
- **Layers of isolation** – changes in one layer generally don't affect other layers, low coupling
- Necessary for each layer to know the interface (API) for the layer directly blow it
- Open layer means layers above need to know more interfaces, higher coupling

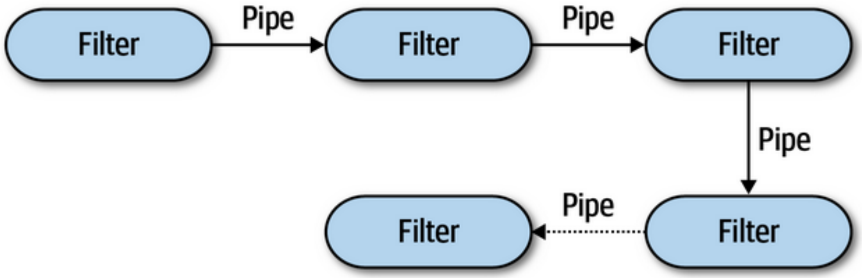




Architectures

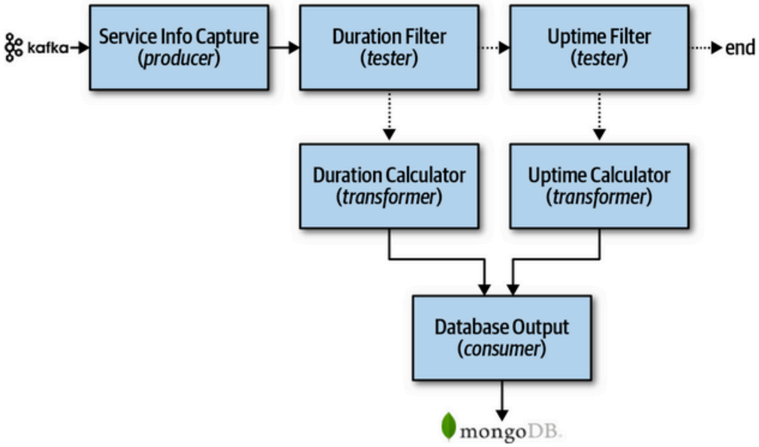
- Monolithic
  - Layered architecture
  - Pipeline architecture
  - Microkernel architecture
- Distributed
  - Service-based architecture
  - Event-driven architecture
  - Space-based architecture
  - Service-oriented architecture
  - Microservices architecture

Pipeline Architecture



- Pipeline – data is passed through a series of data processing steps or filters
- Common architecture for processing data, offers easy configuration of data processing steps and can be distributed
- Major design decision behind the UNIX operating system
  - Simple utilities can be piped together to solve problems
  - In shell: ls | grep | cut

Types of Filters



- Producer – starting point or data source
- Transformer – accepts input, performs a transformation, then forwards transformed data to an output pipe
- Tester – decides whether to forward data based on criteria
- Consumer – end point for data