

ASSIGNMENT 3

Section:	BSCS 3B
Name:	Zain Israr
Arid No.:	21-arid-685

Question:

Here are some architectural styles given :

- a. Pipes and Filters
- b. 2-Tiered (Client/Server)
- c. N-Tiered
- d. Layered
- e. Blackboard
- f. Model View Controller (MVC)
- g. Object Oriented Style
- h. Heterogeneous Styles..

Discuss three situations/examples in which these architectural styles can be adopted? (Note: give 3 examples for

each architectural style separately).

Answer:

Here are three examples for each of the architectural styles you listed:

a. Pipes and Filters:

- A data processing pipeline that takes in raw data as input, applies a series of transformations to it, and produces cleaned and normalized data as output
- A system for converting audio files from one format to another, where each step in the conversion process is represented as a separate filter
- A system for performing image recognition, where each stage in the recognition process is represented as a separate filter

b. 2-Tiered (Client/Server):

- A web application where the client is a web browser and the server is a web server that serves up HTML, CSS, and JavaScript files
- A desktop application that communicates with a remote database server to store and retrieve data
- A system that consists of a set of sensors and actuators connected to a central controller, where the controller acts as the server and the sensors and actuators act as clients

c. N-Tiered:

- A multi-tier e-commerce application, where the front-end tier consists of web servers serving up the web pages, the middle tier consists of application servers running the business logic, and the back-end tier consists of database servers storing the data
- A cloud-based analytics platform, where the front-end tier consists of user interfaces for interacting with the platform, the middle tier consists of data processing and analysis services, and the back-end tier consists of data storage and management services
- A large-scale distributed system for processing big data, where the front-end tier consists of a load balancer and a set of task dispatchers, the middle tier consists of a set of worker nodes that perform the actual data processing, and the back-end tier consists of a set of data storage and management nodes

d. Layered:

- A multi-layer software system for a computer operating system, where the bottom layer consists of the hardware-specific drivers, the middle layer consists of the kernel and other core system components, and the top layer consists of the user-facing applications and libraries
- A multi-layer network protocol stack, where the bottom layer consists of the physical layer, the middle layer consists of the data link and network layers, and the top layer consists of the transport and application layers
- A multi-layer software system for a mobile app, where the bottom layer consists of the platform-specific code, the middle layer consists of the business logic and data access layer, and the top layer consists of the user interface code

e. Blackboard:

- A system for performing natural language processing tasks, where the blackboard acts as a shared memory space for storing and accessing intermediate results and knowledge during the processing of a text
- A system for scheduling and coordinating tasks in a distributed environment, where the blackboard acts as a central repository for storing and accessing information about the tasks and their dependencies

- A system for performing complex decision-making tasks, where the blackboard acts as a central repository for storing and accessing information and rules used by the decision-making process

f. Model View Controller (MVC):

- A web application for managing a database of customer information, where the model represents the data and business logic, the view is the user interface for displaying and editing the data, and the controller is responsible for handling user input and updating the model and view as necessary
- A desktop application for managing a project, where the model represents the project data and business logic, the view is the user interface for displaying and editing the data, and the controller is responsible for handling user input and updating the model and view as necessary
- A mobile app for managing a to-do list, where the model represents the to-do list data and business logic, the view is the user interface for displaying and interacting with the to-do list, and the controller is responsible for handling user input and updating the model and view as necessary

g. Object Oriented Style:

- A software system for managing a library's collection of books, where each book is represented as an object and the software provides methods for checking out and returning books, searching for books by various criteria, and so on
- A software system for managing a fleet of vehicles, where each vehicle is represented as an object and the software provides methods for tracking the location, maintenance status, and other attributes of each vehicle
- A software system for managing a company's financial accounts, where each account is represented as an object and the software provides methods for performing transactions, generating reports, and so on

h. Heterogeneous Styles:

- A software system that combines a microservices architecture with an event-driven architecture, where each microservice is responsible for a specific set of tasks and communicates with other microservices through asynchronous events
- A software system that combines an object-oriented architecture with a functional programming style, where the object-oriented portion of the system handles the state management and the functional portion handles the data processing
- A software system that combines a client/server architecture with a peer-to-peer architecture, where the client/server portion handles the communication with external systems and the peer-to-peer portion handles the communication between internal components.

