

## DEPARTMENT OF CSE , CSIT & AI&DS

**COURSE NAME : ADAPTIVE SOFTWARE ENGINEERING**  
**COURSE CODE : 23CI2001**

**Topic:**

## ARCHITECTURAL PATTERNS

**Session - 18**



Simple

## AIM OF THE SESSION



To familiarize students with the concept of **ARCHITECTURAL PATTERNS**

---

## INSTRUCTIONAL OBJECTIVES



This Session is designed to:

1. Describe and Demonstrate **ARCHITECTURAL PATTERNS**

## LEARNING OUTCOMES



At the end of this session, you would be able to:

1. Know and understand **ARCHITECTURAL PATTERNS**.

## AGENDA

---

- ❖ **Introduction**
- ❖ **Different Architectural Patterns**
  - Layered Pattern
  - Client-Server Pattern
  - Event-Driven Pattern
  - Microkernel Pattern
  - Micro services Pattern

## INTRODUCTION

---

- Software architecture is the blueprint of building software. It shows the overall structure of the software, the collection of components in it, and how they interact with one another while hiding the implementation
- This helps the software development team to clearly communicate how the software is going to be built as per the requirements of customers.
- Architecture patterns determine the destiny of the software about to be built. And there is no one-stop solution to build any kind of software.

# DIFFERENT ARCHITECTURAL PATTERNS

---

Among the multiple software architecture patterns, we are going to see a few of the most important and commonly used patterns.

### **Different Software Architecture Patterns :**

1. Layered Pattern
2. Client-Server Pattern
3. Event-Driven Pattern
4. Microkernel Pattern
5. Microservices Pattern

## 1. LAYERED PATTERN

---

- components(code) in this pattern are separated into layers of subtasks and they are arranged one above another.
- Since each layer is independent, one can modify the code inside a layer without affecting others
- It is the most commonly used pattern for designing the majority of software. This layer is also known as 'N-tier architecture'.
- Basically, this pattern has 4 layers.
  - Presentation layer (The user interface layer where we see and enter data into an application.)
  - Business layer (this layer is responsible for executing business logic as per the request.)
  - Application layer (this layer acts as a medium for communication between the 'presentation layer' and 'data layer'.
  - Data layer (this layer has a database for managing data.)

**NOTE:** Ideal For E-commerce Web Applications Development Like Amazon

## 2. CLIENT-SERVER PATTERN

---

- The client-server pattern has two major entities. They are a server and multiple clients.
- Here the server has resources(data, files or services) and a client requests the server for a particular resource. Then the server processes the request and responds back accordingly.
- Examples of software developed in this pattern:
  - Email.
  - WWW.
  - File sharing apps.
  - Banking, etc...

**NOTE:** This Pattern Is Suitable For Developing The Kind Of Software Listed In The Examples.

### 3. EVENT-DRIVEN PATTERN

---

- Event-Driven Architecture is an agile approach in which services (operations) of the software are triggered by events.
- Well, what does an event mean?

When a user takes action in the application built using the EDA approach, a state change happens and a reaction is generated that is called an event.

- **Eg:** A new user fills the signup form and clicks the signup button on Facebook and then a FB account is created for him, which is an event.

**NOTE:** ideal for Building websites with JavaScript and e-commerce websites in general



### 4. MICROKERNEL PATTERN

---

- Microkernel pattern has two major components. They are a core system and plug-in modules.
  - The core system handles the fundamental and minimal operations of the application.
  - The plug-in modules handle the extended functionalities (like extra features) and customized processing.

**NOTE:** ideal for Product-based applications and scheduling applications.  
Such as Instagram reels, YouTube Shorts.  
So this pattern is mostly preferred for app development.

## 5. MICRO SERVICES PATTERN

---

- The collection of small services that are combined to form the actual application is the concept of micro services pattern. Instead of building a bigger application, small programs are built for every service (function) of an application independently. And those small programs are bundled together to be a full-fledged application.
- So adding new features and modifying existing micro services without affecting other micro services are no longer a challenge when an application is built in a micro services pattern.
- Modules in the application of micro services patterns are loosely coupled. So they are easily understandable, modifiable and scalable.

**NOTE:** this pattern is most suitable for websites and web apps having small components.

# CONCLUSION

---

- Architecture patterns determine the destiny of the software about to be built. And there is no one-stop solution to build any kind of software. So choose what suits you the most!

## SELF-ASSESSMENT QUESTIONS

1. Define Architectural Patterns?
2. Name the different Architectural Patterns?
3. Explain in detail about different Architectural Patterns?

## Reference Books:

---

### TEXTBOOKS:

1. Roger S.Pressman, “Software Engineering – A Practitioner’s Approach” 7th Edition, Mc Graw Hill,(2014).
2. Ian Sommerville, “Software Engineering”, Tenth Edition, Pearson Education, (2015).
3. Agile Software Development Ecosystems, Jim Highsmith, Addison Wesley; ISBN: 0201760436; 1st edition

### Reference Book

Agile Modelling: Effective Practices for Extreme Programming and the Unified Process Scott Amber John Wiley & Sons; ISBN: 0471202827; 1st edition.

### WEB REFERENCES/MOOCs:

<https://www.digite.com/kanban/what-is-kanban/>  
<http://www.scaledagileframework.com>  
<https://www.guru99.com/test-driven-development.html>  
<https://junit.org/junit5/>

**THANK YOU**



**Team – Adaptive Software Engineering**