



CSE 3421

Software Architecture

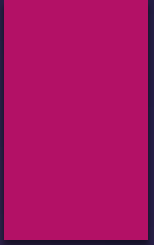
SPRING 2022

MD. RAFI-UR-RASHID

LECTURER, DEPT. OF CSE, UIU



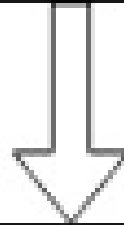
Why This Lesson?



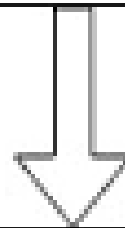
It is essential to learn how a software is
modularized and distributed into
manageable pieces

N-tier (Layered) Architecture

Presentation Layer



Business Logic
Layer



Data Access Layer

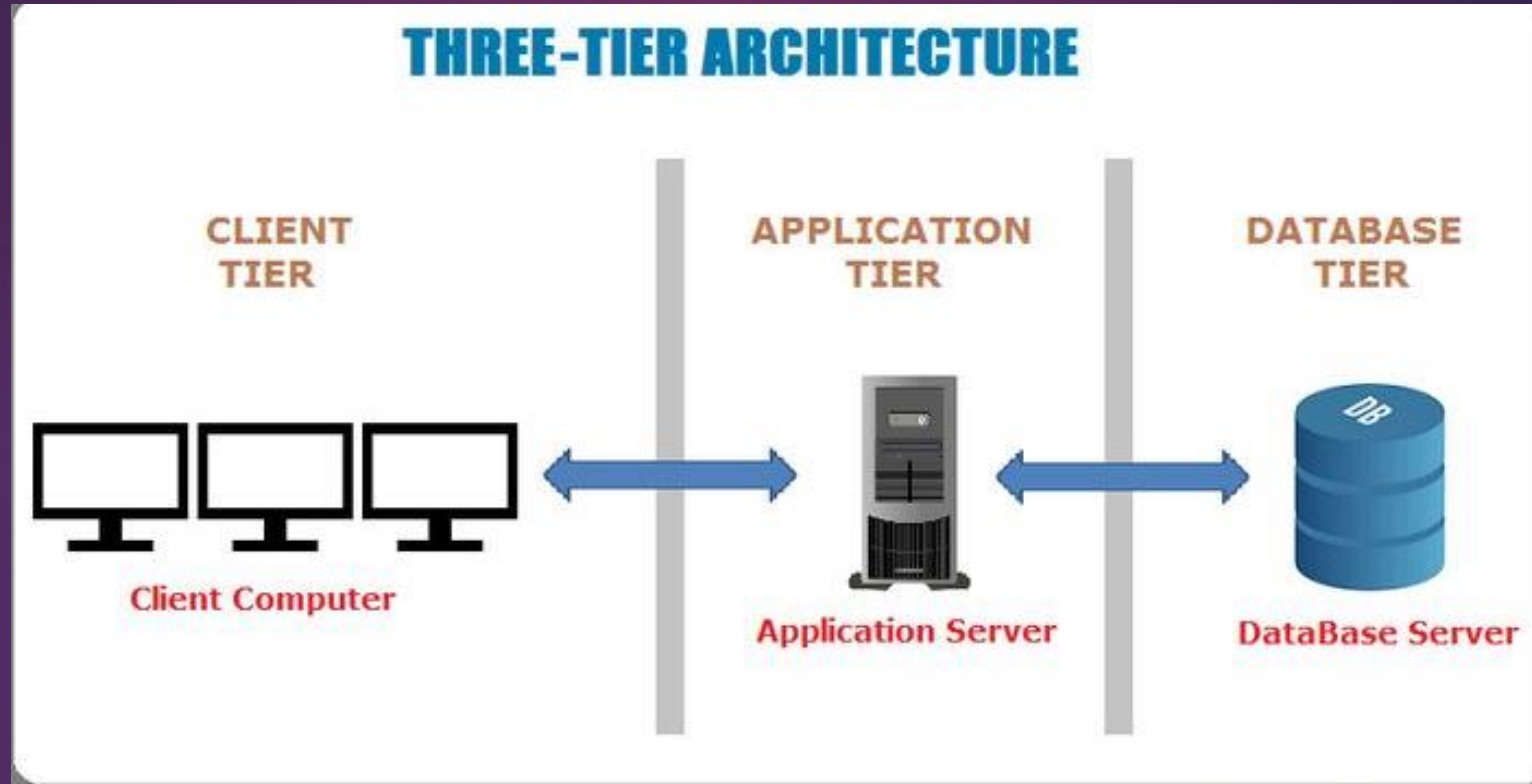
What is N-tier architecture?

*An approach where the project/program is distributed among **N** separate computers in a distributed network.*

Types of N-tier architecture

- ▶ 3-tier
- ▶ 2-tier
- ▶ 1-tier

3-tier Architecture



Presentation tier

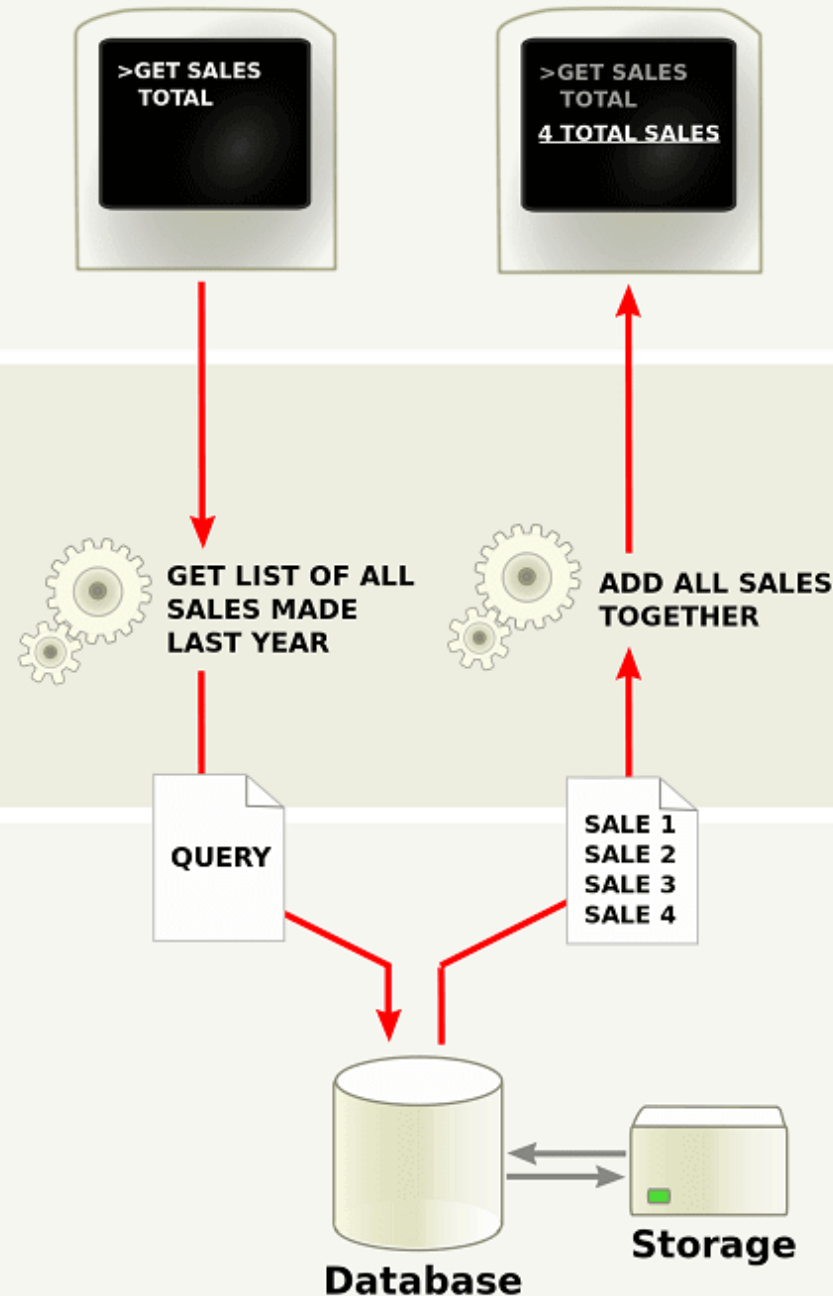
The top-most level of the application is the user interface. The main function of the interface is to translate tasks and results to something the user can understand.

Logic tier

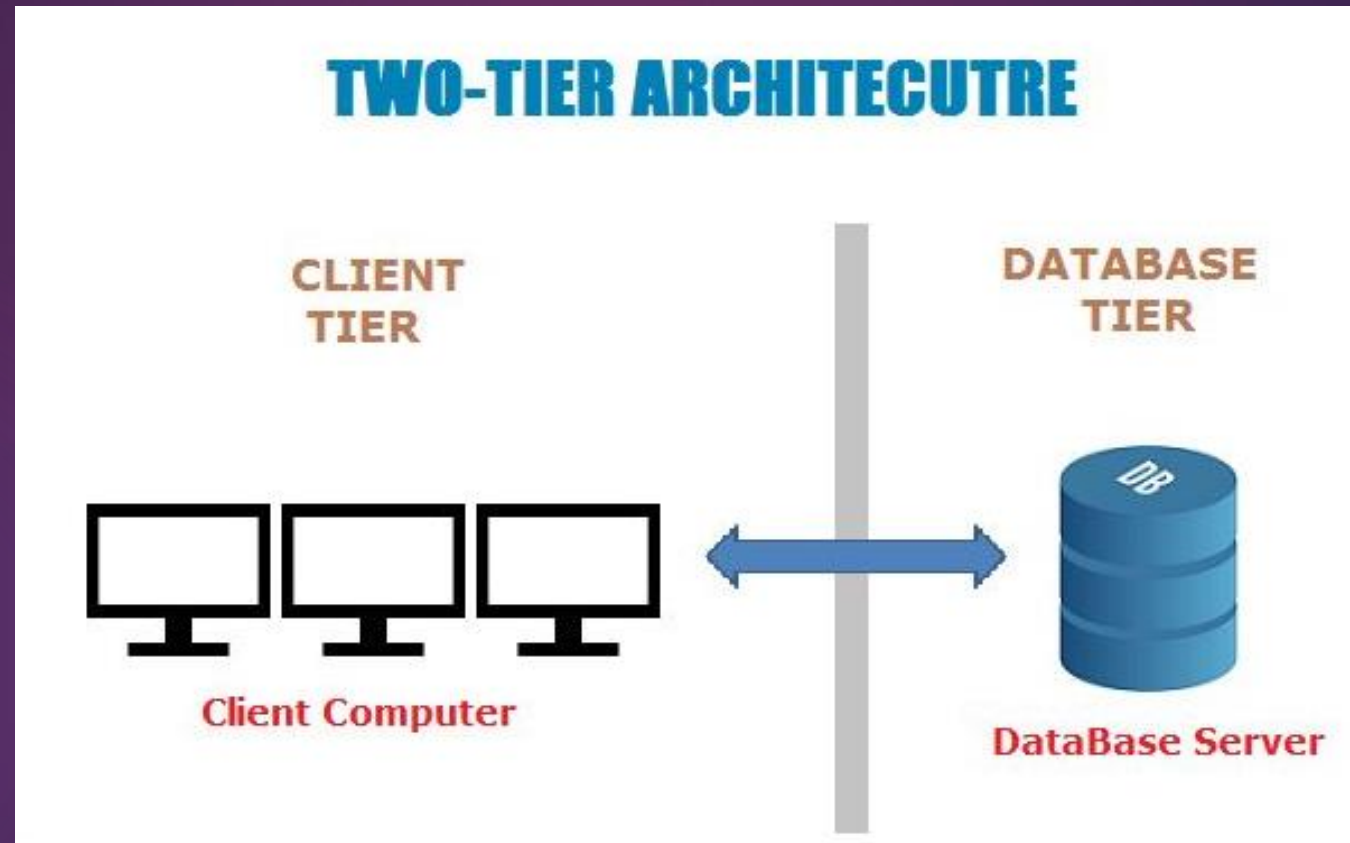
This layer coordinates the application, processes commands, makes logical decisions and evaluations, and performs calculations. It also moves and processes data between the two surrounding layers.

Data tier

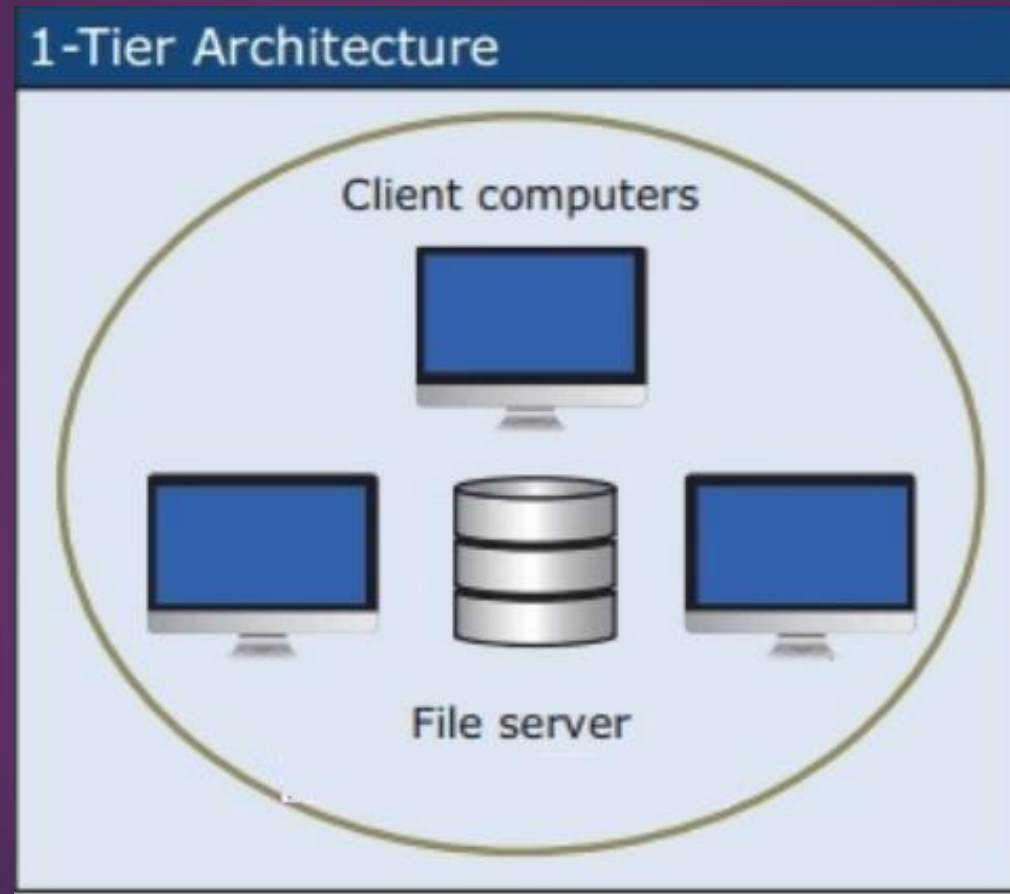
Here information is stored and retrieved from a database or file system. The information is then passed back to the logic tier for processing, and then eventually back to the user.



2-tier Architecture



1-tier Architecture



Impacts of multi-tiering

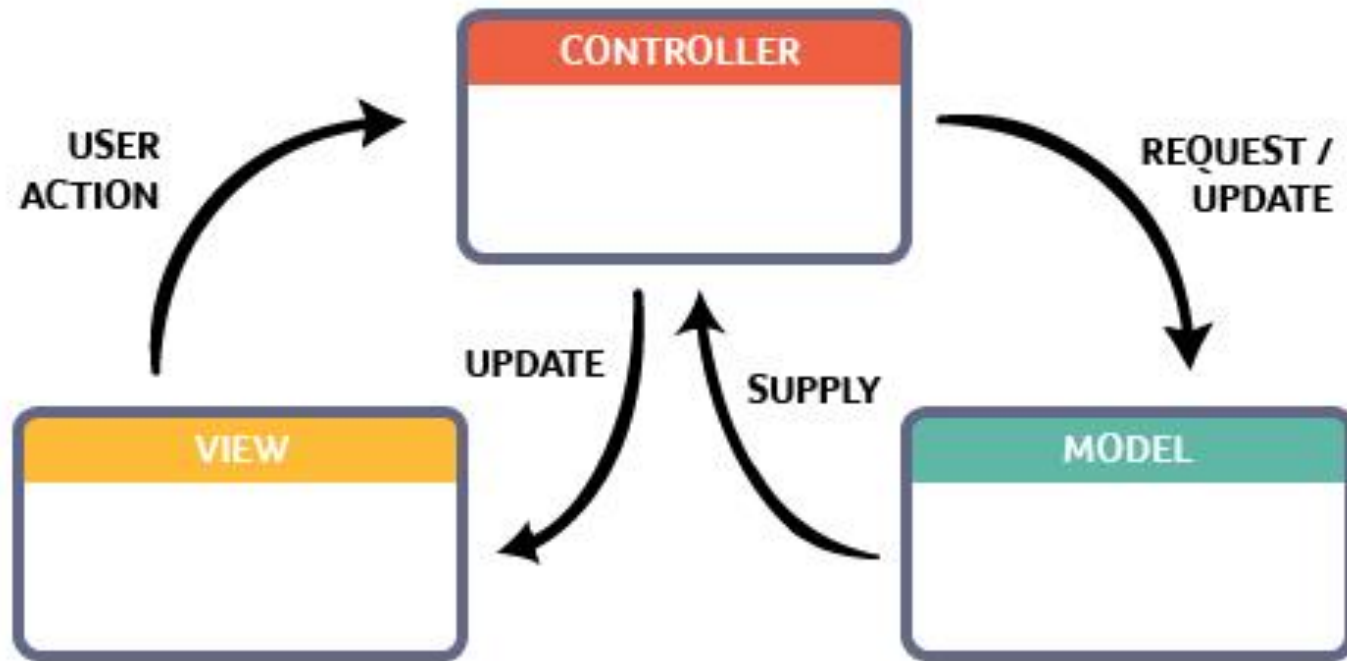
- ▶ Scalability
- ▶ Reliability
- ▶ Fault tolerance
- ▶ Security
- ▶ Maintainability
- ▶ Reusability

Software Architectural Patterns

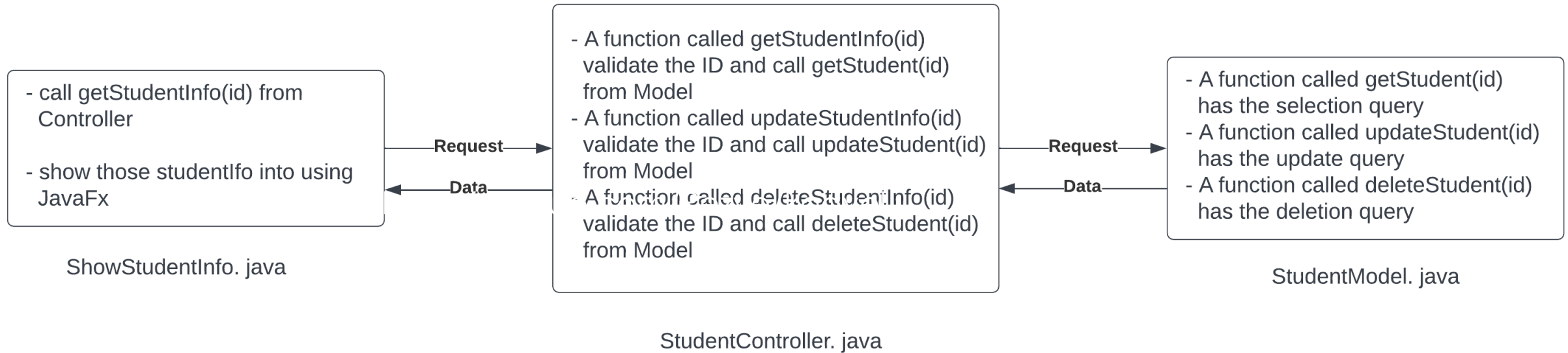
Types of Layered Architectural Pattern

- ▶ MVC
- ▶ HMVC
- ▶ MVVM
- ▶ MVP

MVC



Example: Student Module



MVC Components

- ▶ **Model:** The Model component corresponds to all the **data-related logic** that the user works with. This can represent the related data that is being transferred between the View and Controller components.
- ▶ **View:** The View component is used for all the **UI logic** of the application. For example, the Customer view will include all the UI components such as text boxes, dropdowns, etc.
- ▶ **Controller:** Controllers act as an interface between Model and View components to process all the business logic and incoming requests, manipulate data using the Model component and interact with the Views to render the final output.

MVC frameworks

- ▶ Spring (Java)
- ▶ Django (python)
- ▶ Rails (Ruby)
- ▶ Laravel (PHP)

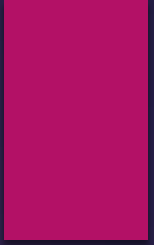

Advantages

- ▶ Rapid development
- ▶ Low coupling
- ▶ Efficient debugging
- ▶ High maintainability

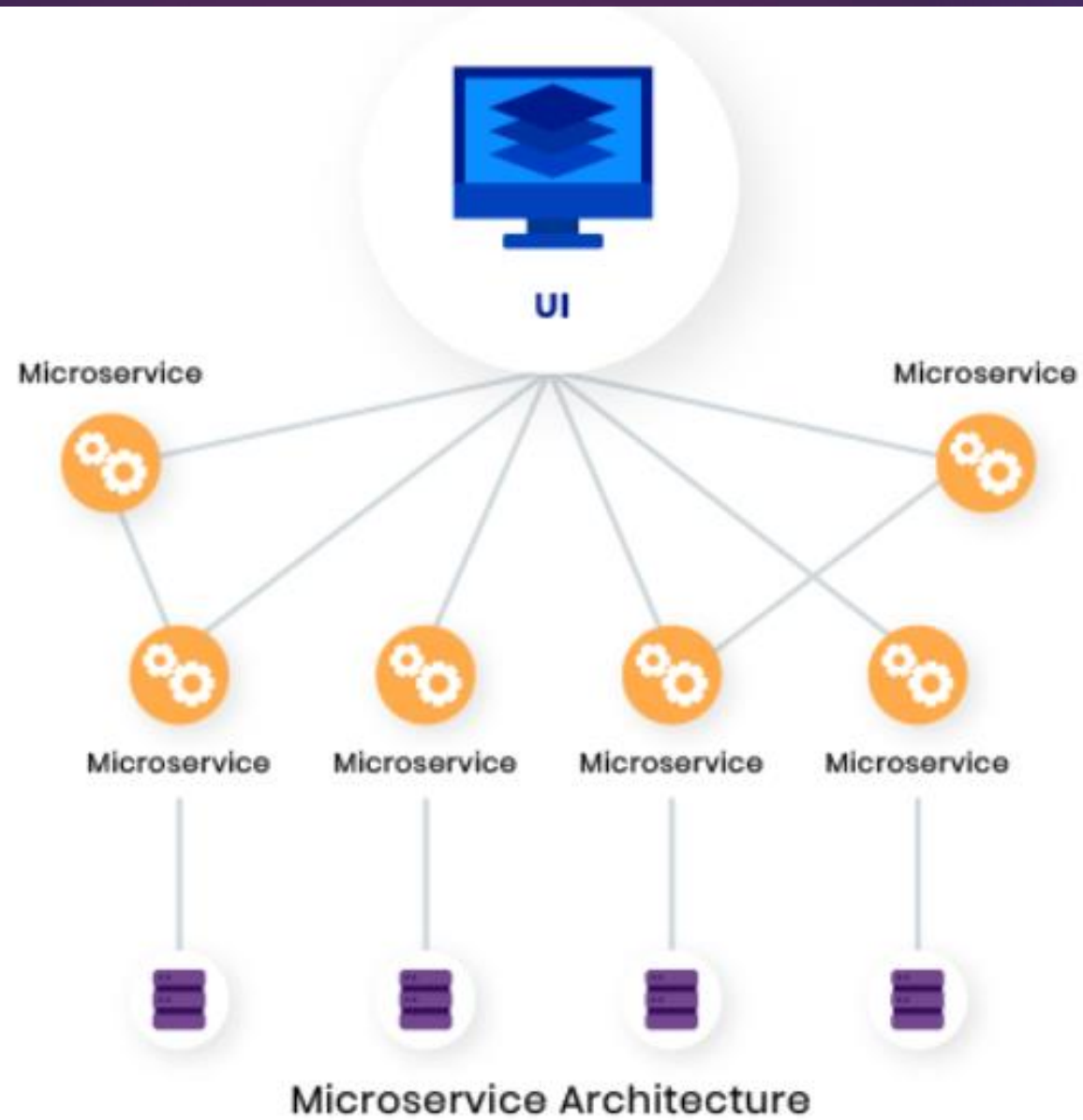
Monolithic Architecture

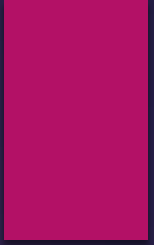



Monolithic Architecture

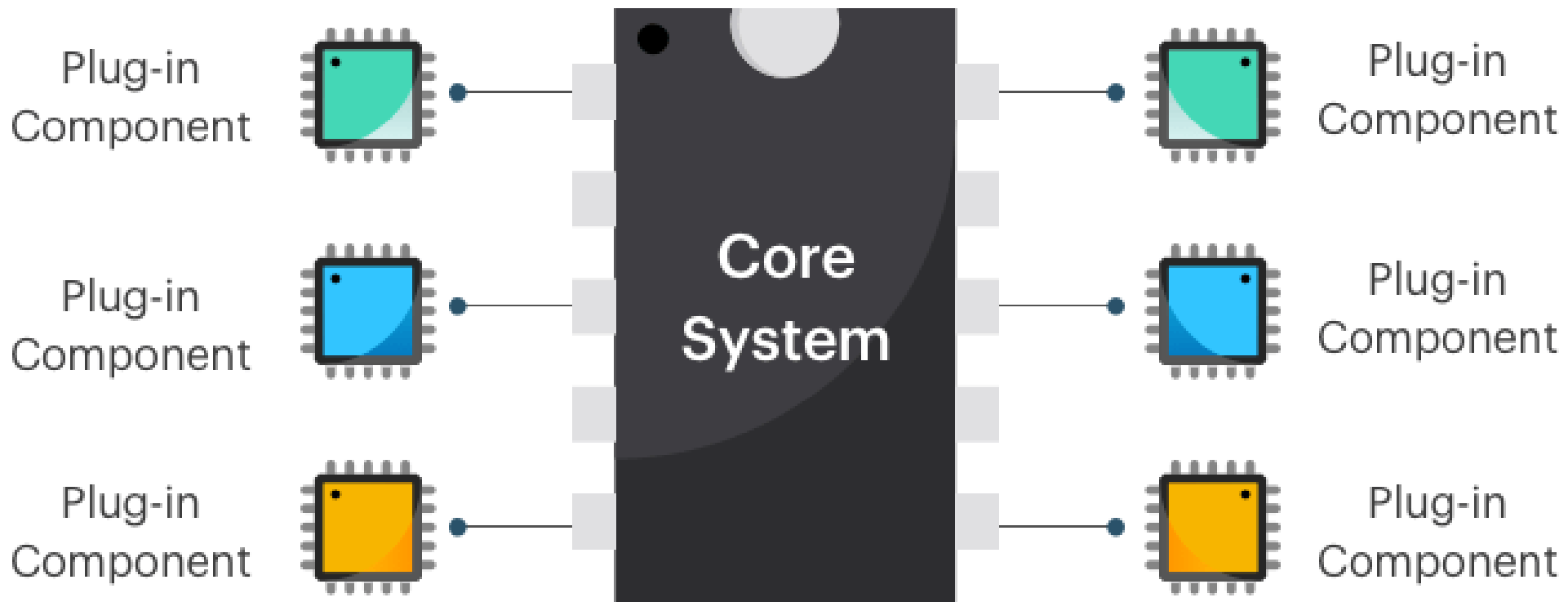
- 
- 
- ▶ It is built as a single and indivisible unit and all the functions are managed and served in one place.
 - ▶ Low cost architecture
 - ▶ Simple for development and deployment.
 - ▶ Difficult for maintenance, Scalability
 - ▶ Tight coupling

Microservice Architecture



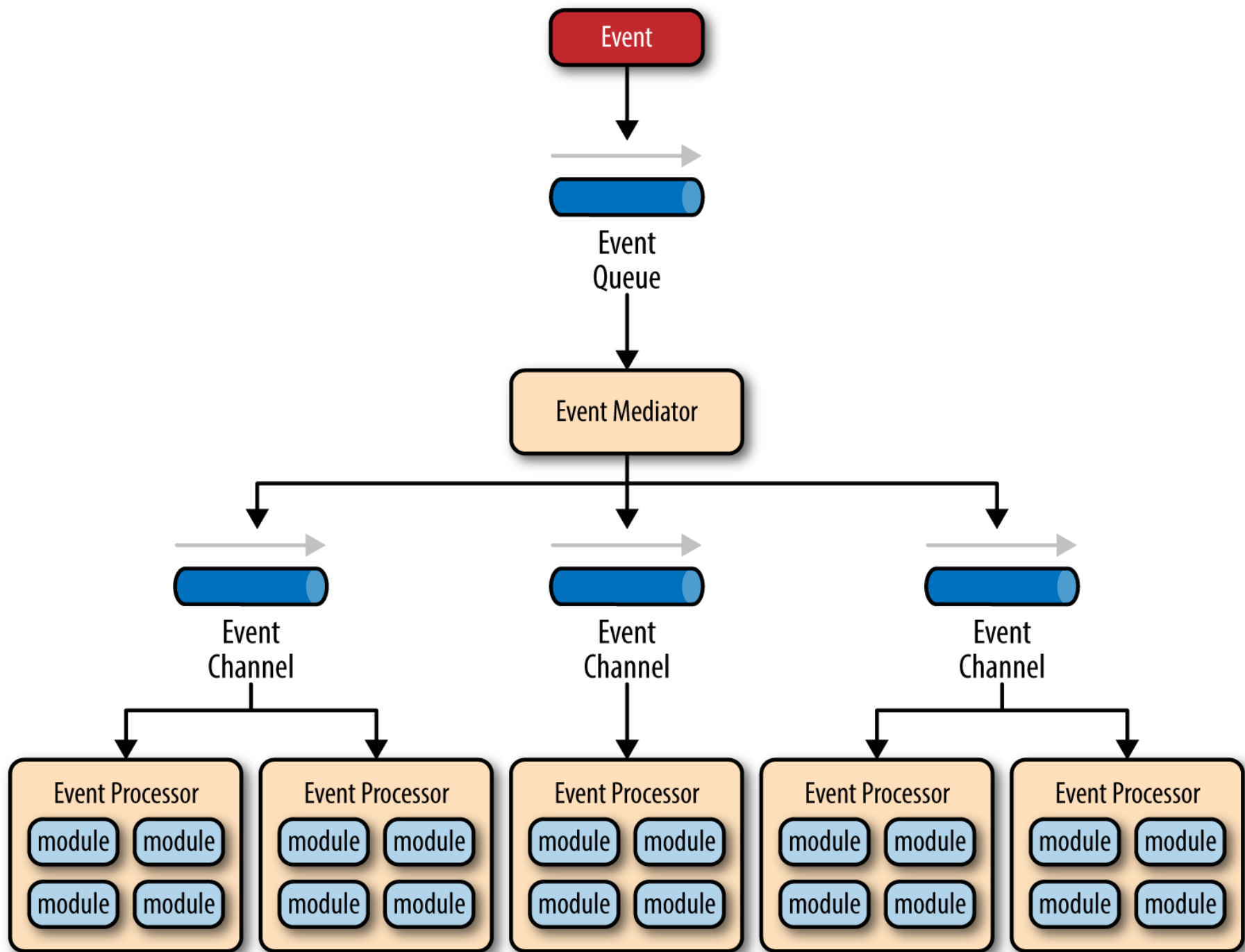
- 
- 
- ▶ Breaks it down into a collection of smaller independent units.
 - ▶ Each service has its own logic and the database as well as perform the specific functions.
 - ▶ Services intercommunicate with some lightweight mechanisms, often an HTTP resource API
 - ▶ Easy maintainability, Scalability, and low coupling
 - ▶ Extra complexity, cross-cutting concerns, and cost

Micro-Kernel Architecture Pattern



- ▶ Also called plug-in architecture which consists of two types of components – a core system and several plug-in modules.
- ▶ The core system works on minimal functionality to keep the system operational, the plug-in modules are independent components with specialized processing.
- ▶ Suitable for applications that have a fixed set of core routines and dynamic set of rule that needs frequent updates.
- ▶ The plugins must have good handshaking code so that the microkernel is aware of the plugin installation and is ready to work.

Event-Driven Architecture



- 
- 
- ▶ It is made up of decoupled, single-purpose event processing components that asynchronously receive and process events.
 - ▶ Agile and highly performant
 - ▶ Easy debugging, fast development and low coupling



Thank You