تحلیل و طراحی سیستم‌ها

System Design and Analysis

Chapter 7:
Software Design

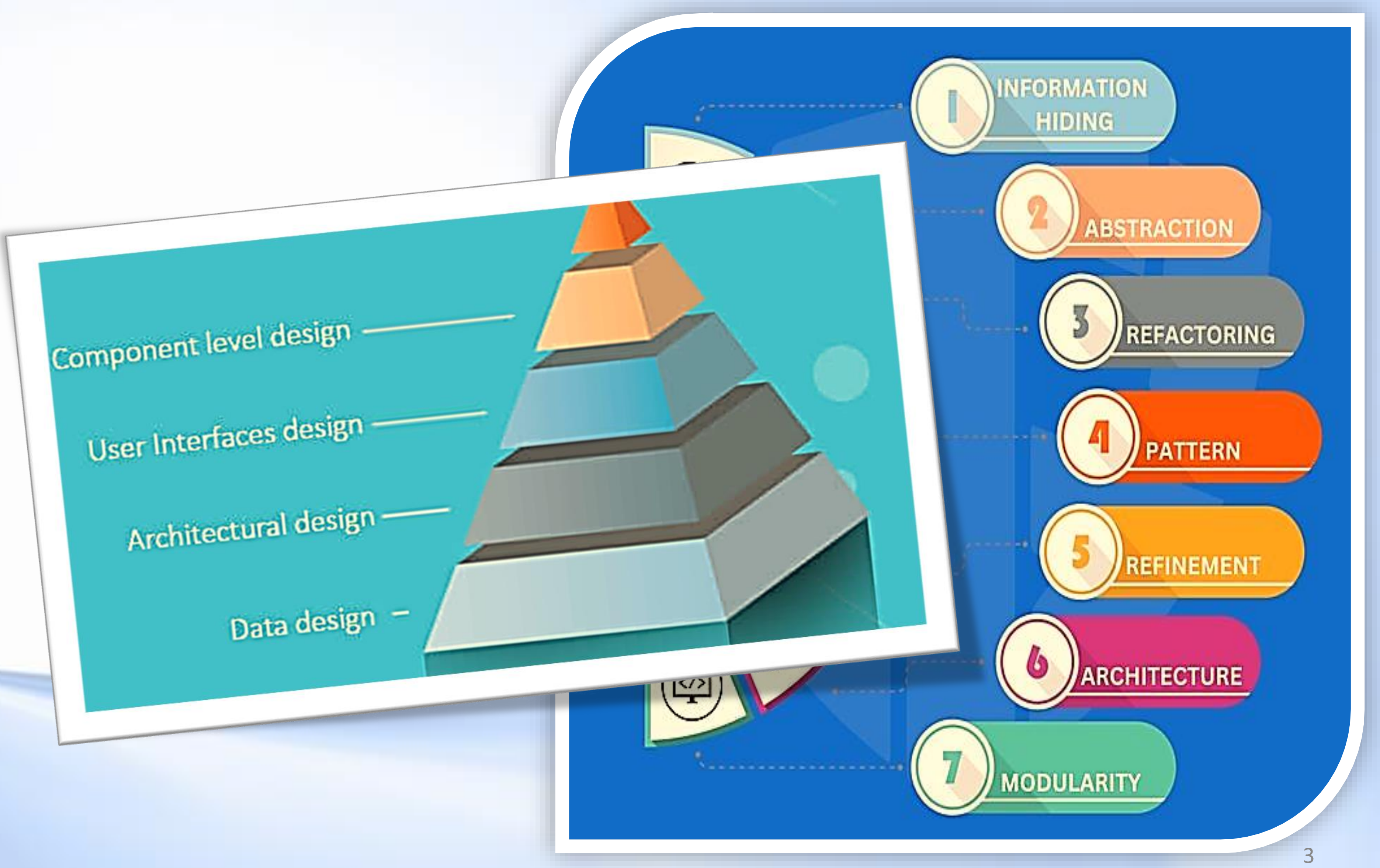
Presented by: Elahe Habibi

The 2nd semester of 1402-1403

Table of Contents

- Review of Software Design (previous session)
- Introduction to Architectural Design
- Architectural Views
- An architectural Style

Review of Software Design (previous session)



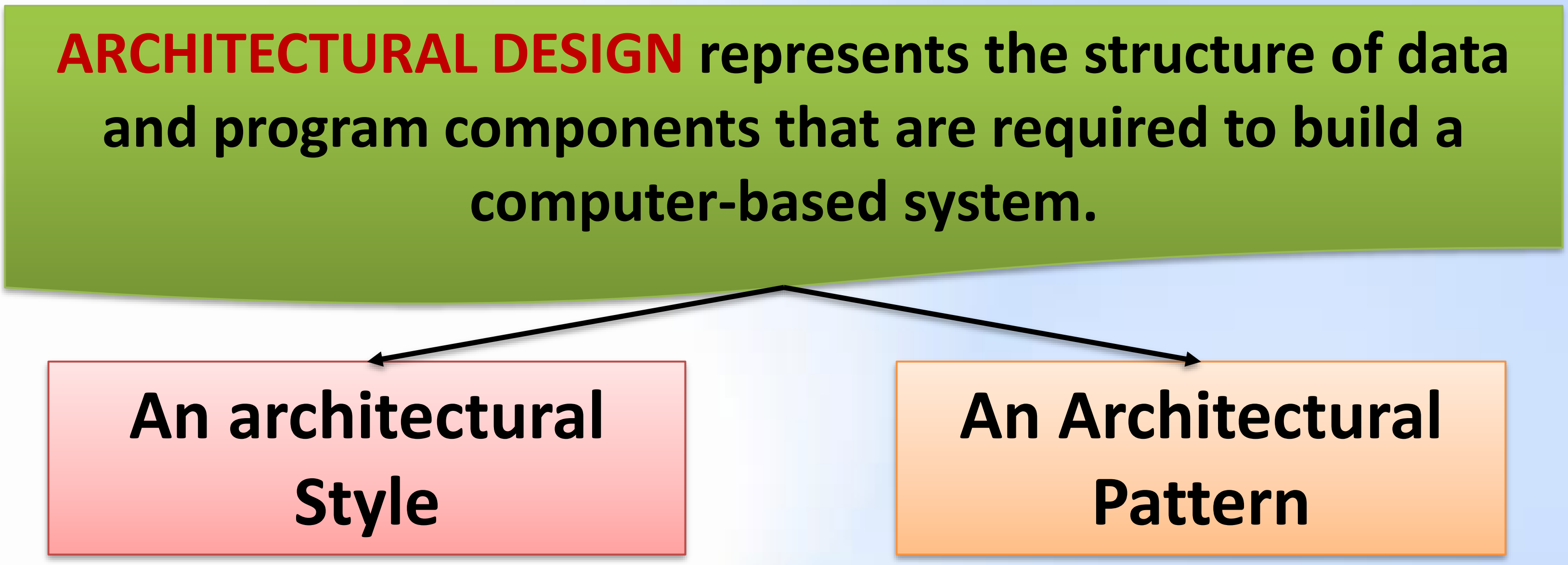
Introduction to Software Architecture (1)

عمارت وینچستر
نمونه ای از یک عدم معماری

- ❖ تعداد اطاقها: ١٦٠
- ❖ هزینه ساخت: ٥.٥ میلیون دلار
- ❖ مدت ساخت: ٣٨ سال (١٨٨٤-١٩٢٢)

- ❑ تعداد درها: ٤٦٧
- ❑ تعداد درهائی که بجائی باز نمی شود: ٩٥٠
- ❑ تعداد پنجره: ١٢٥٧(١٠٠٠٠قطعه شیشه)

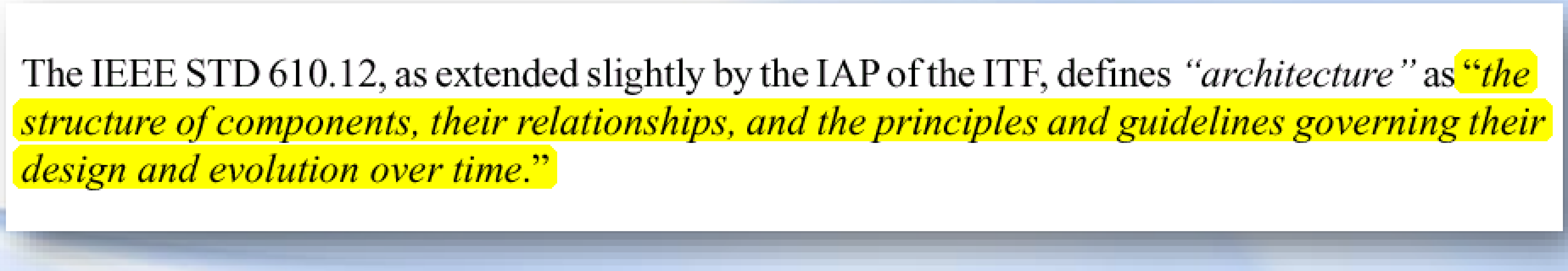
Architectural Design (1)



ARCHITECTURAL DESIGN represents the structure of data and program components that are required to build a computer-based system.

An architectural Style

An Architectural Pattern



The IEEE STD 610.12, as extended slightly by the IAP of the ITF, defines *"architecture"* as *"the structure of components, their relationships, and the principles and guidelines governing their design and evolution over time."*

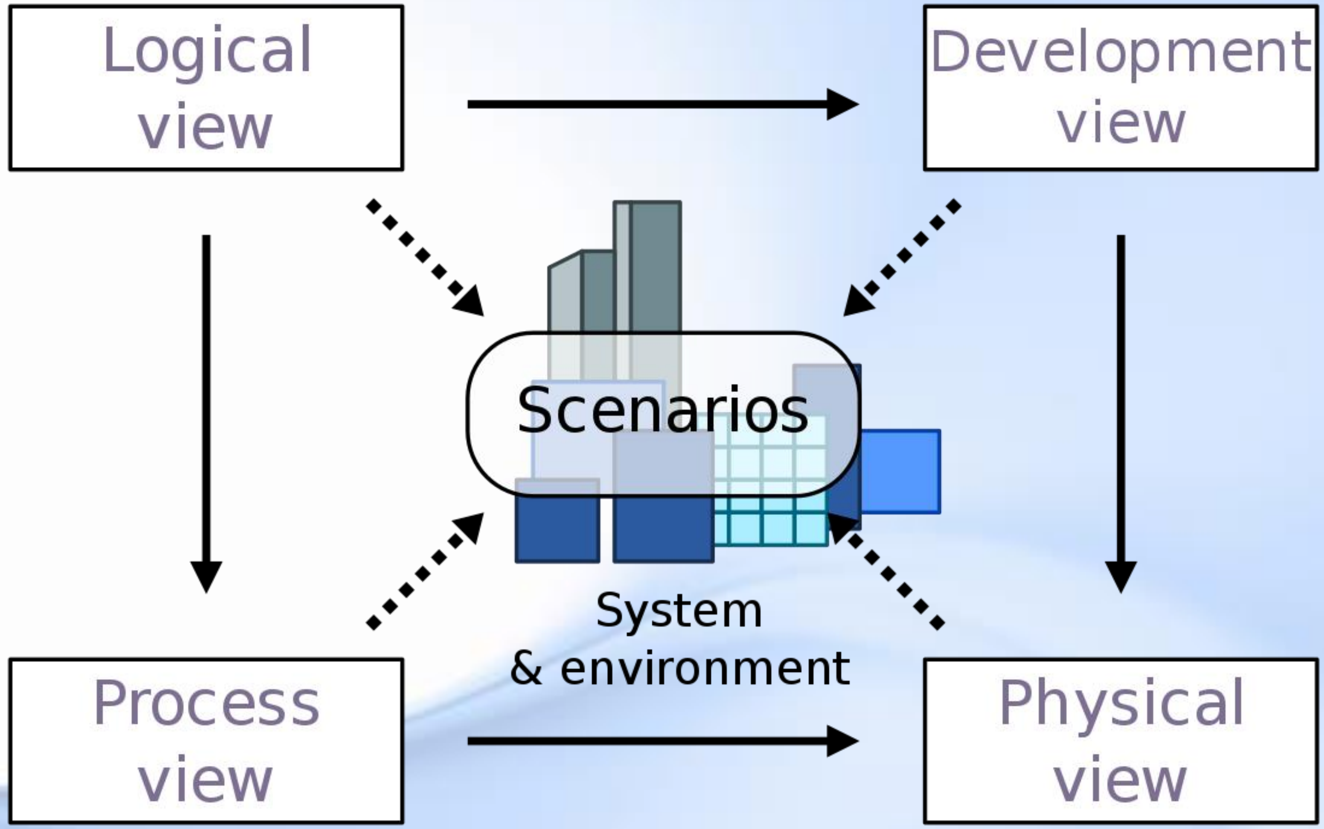
Software architecture = Components + Connectors

Architectural Design (2)

- **Structures** is a set of coherent elements and the relations among them. For each structure these we can specify:
  - Types of elements
  - Types of relations
  - A set of constraints
  - Semantics of the diagram
  - Principles, and guidelines
  - For what purposes it is useful

- **View** is a representation of software architecture based on an structure as written by the architect and read by stakeholders (an instance of the structure)

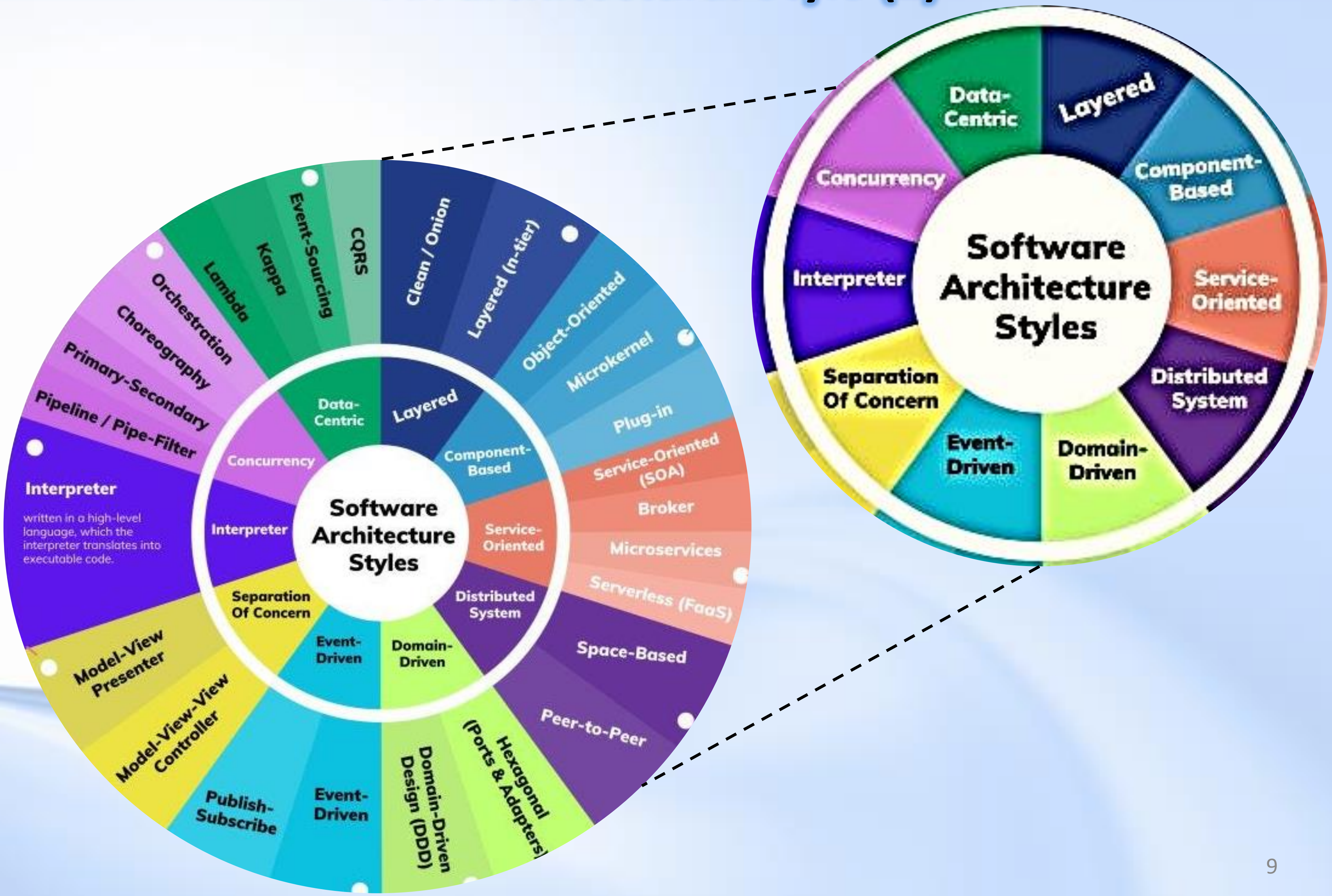
Architectural Views (1)



Architectural Views (2)

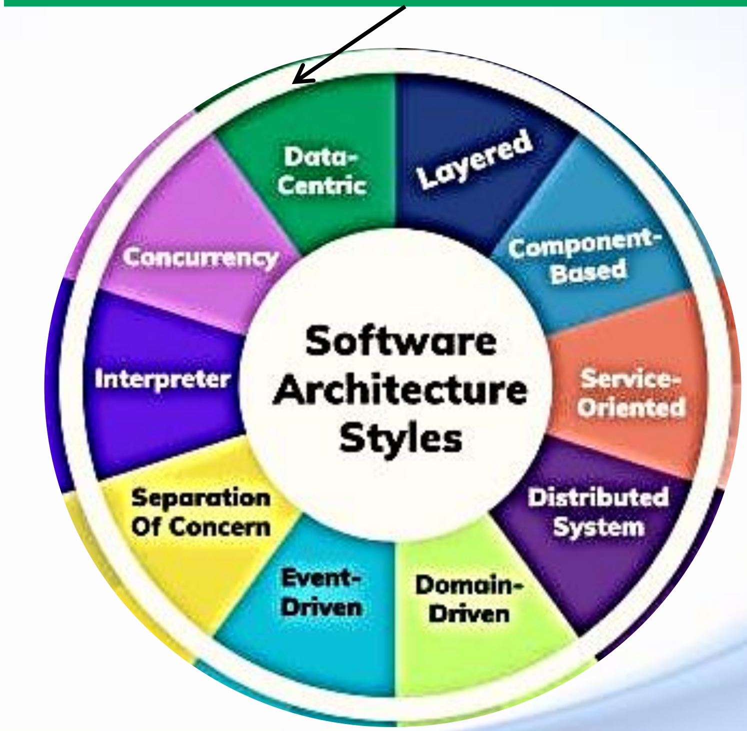
- **Process**: addresses concurrency & distribution of functionality. (a component and connector view)
- **Logical**: objects and classes, elements are “key abstractions” that are objects or classes in OO. (a module view)
- **Development**: shows organization of software modules, libraries, subsystems, and units of development. (an allocation view)
- **Physical**: maps other elements onto processing & communication nodes, also an allocation view, but usually referred to specifically as the deployment view. (an allocation view)

An architectural Style (1)

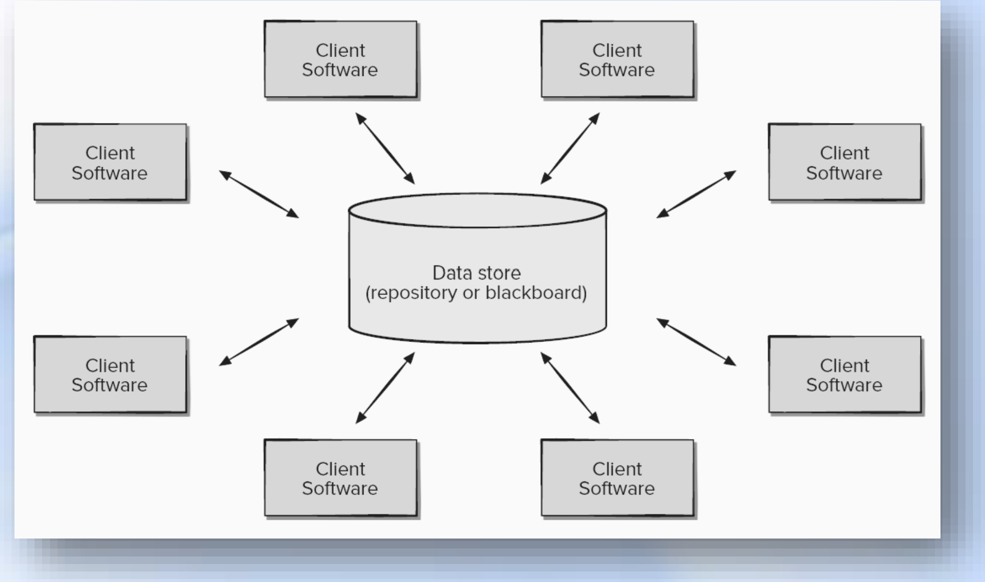


An architectural Style (2)

Data-Centered Architectures

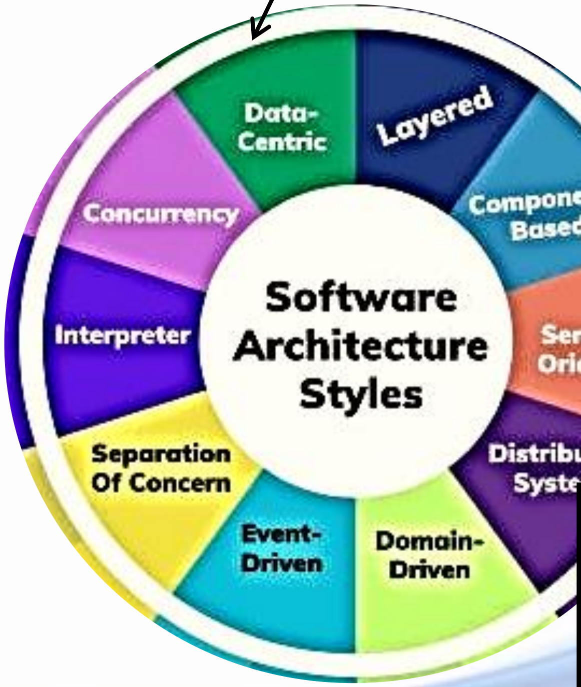


A data store (e.g., a file or database) resides at the center of this architecture and is accessed frequently by other components that update, add, delete, or otherwise modify data within the store.

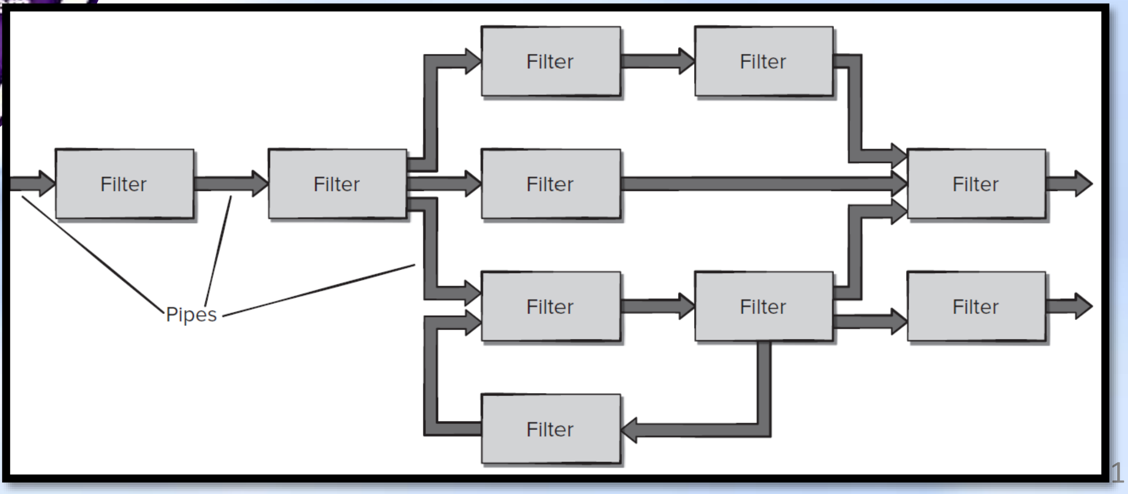


An architectural Style (3)

Data-flow architecture

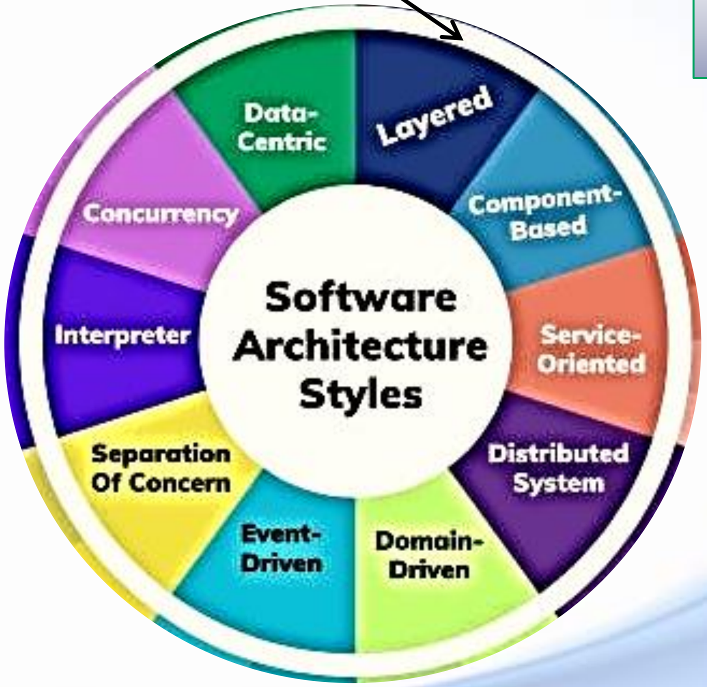


This architecture is applied when input data are to be transformed through a series of computational or manipulative components into output data. A pipe-and-filter has a set of components, called filters, connected by pipes that transmit data from one component to the next.

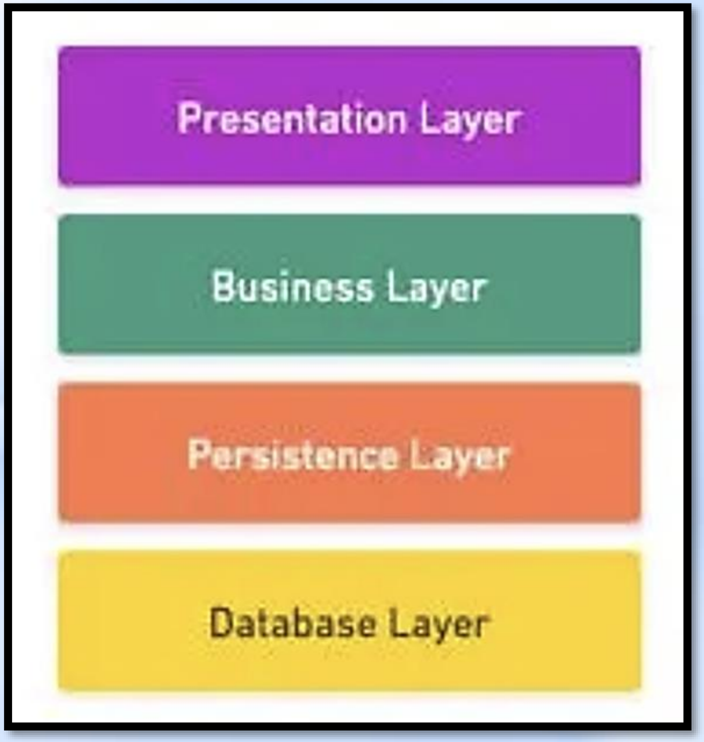


An architectural Style (4)

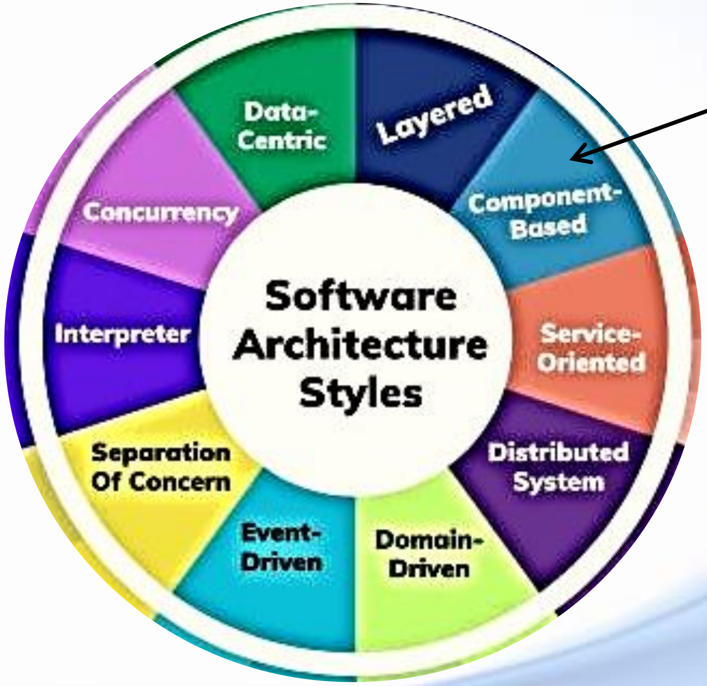
Layer (n-tier) architecture



Separate Software into logical layers

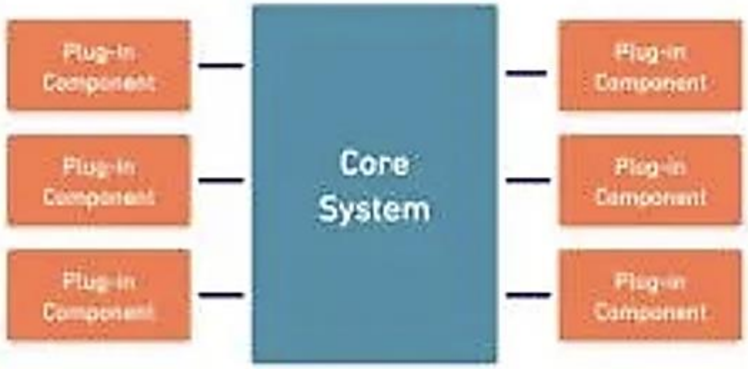


An architectural Style (5)

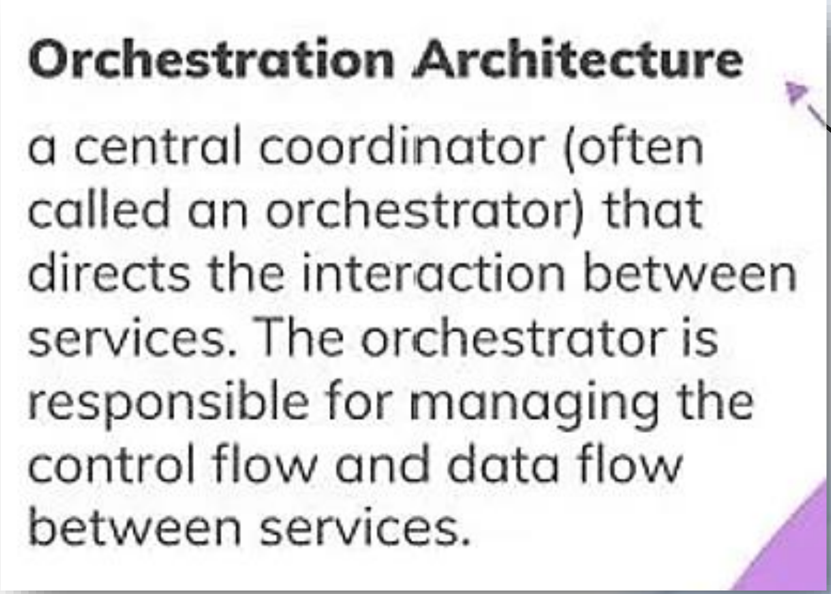


Microkernel Architecture

separates a minimal functional core from extended functionality and customer-specific parts.

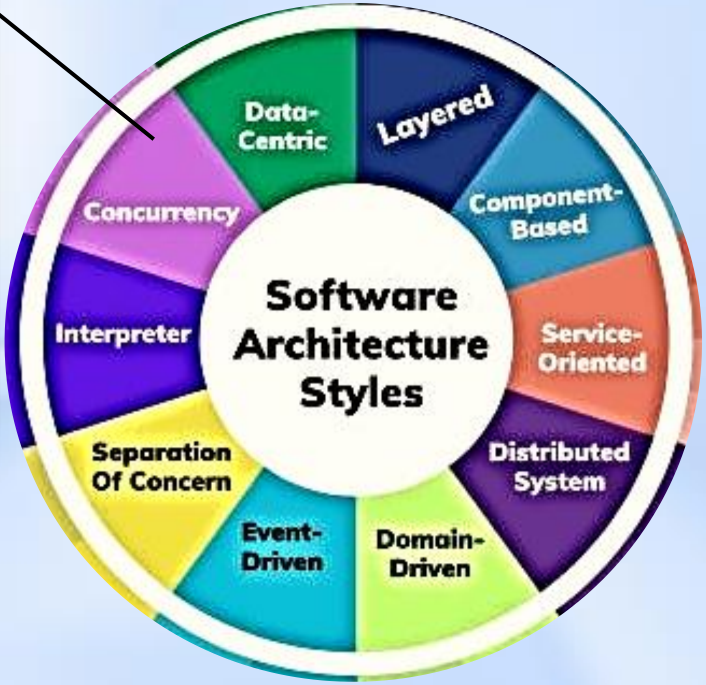


An architectural Style (6)

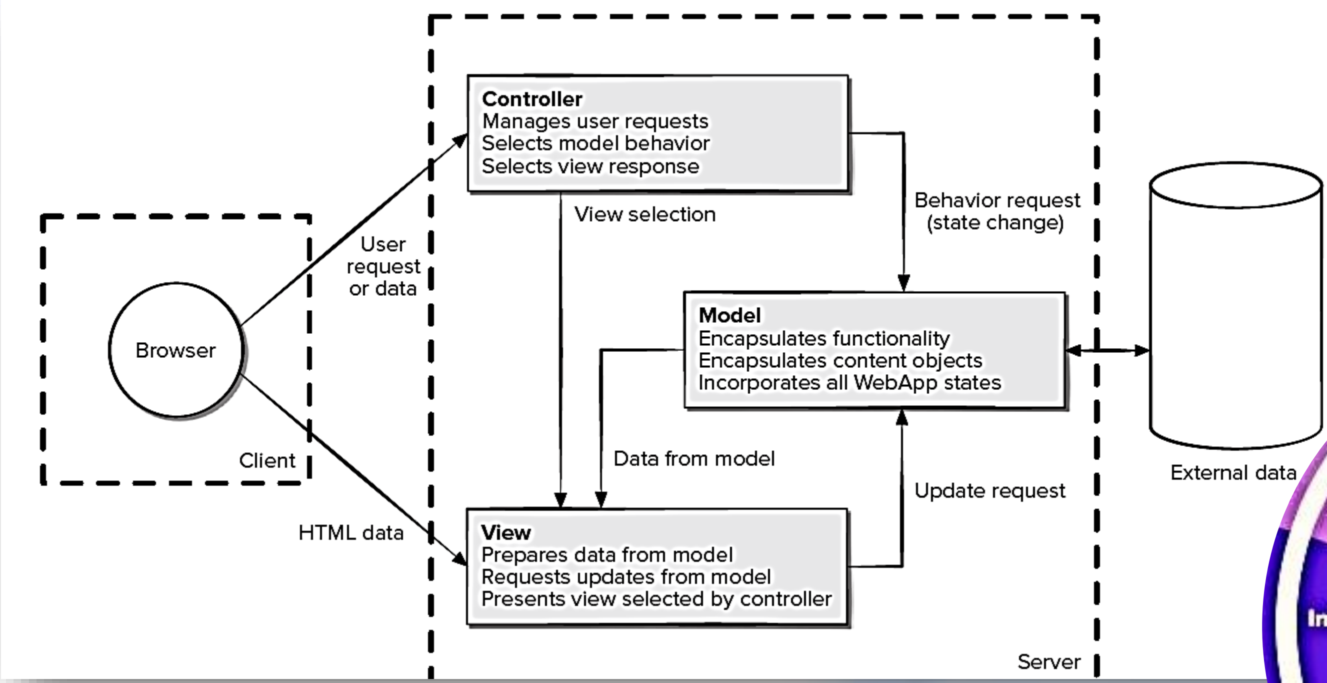


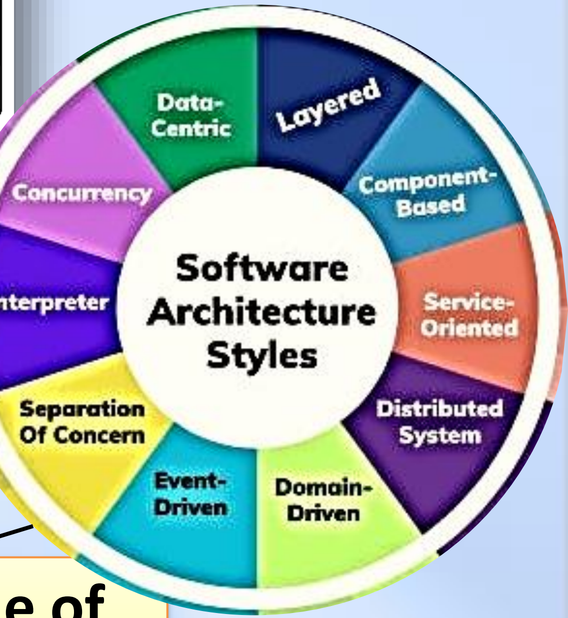
Orchestration Architecture

a central coordinator (often called an orchestrator) that directs the interaction between services. The orchestrator is responsible for managing the control flow and data flow between services.



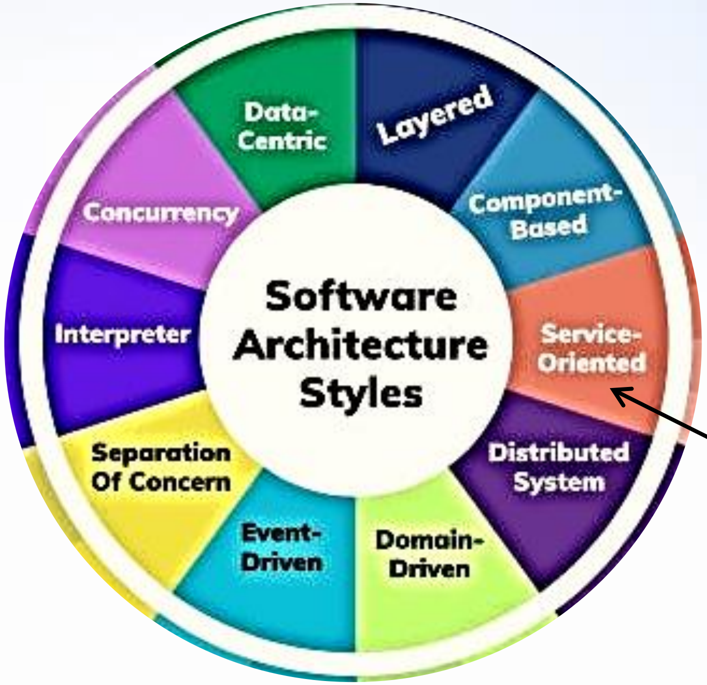
An architectural Style (7)

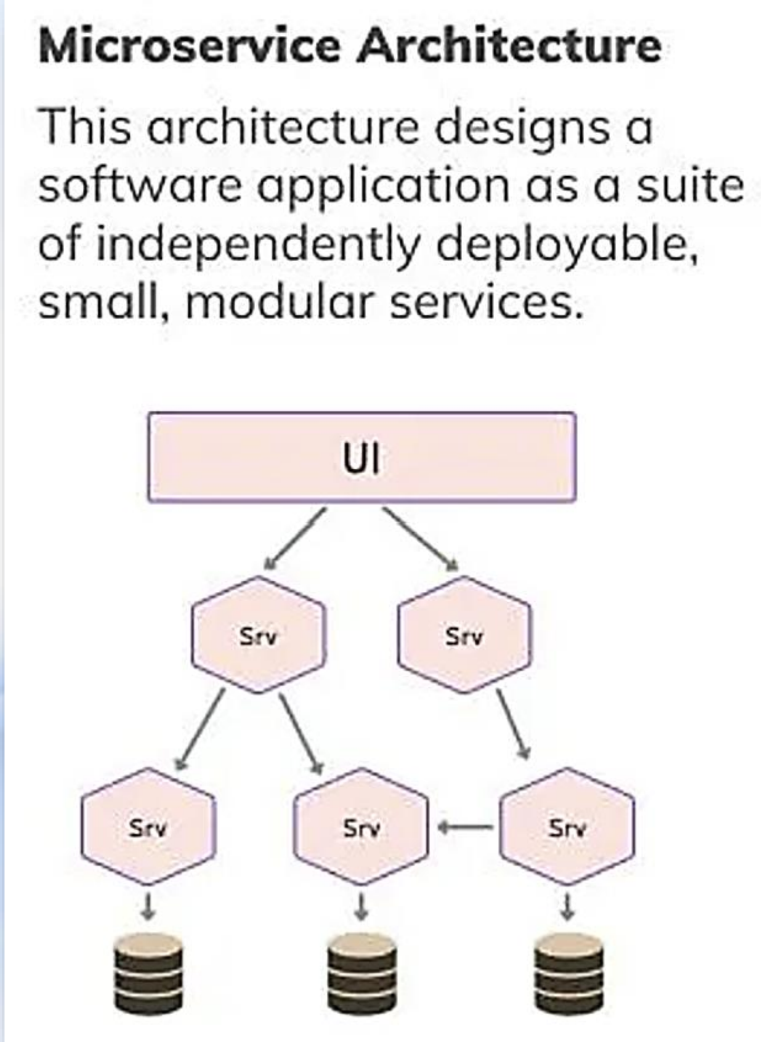




Model-View-Controller (MVC) architecture is one of a number of suggested mobile infrastructure models often used in Web development.

An architectural Style (8)

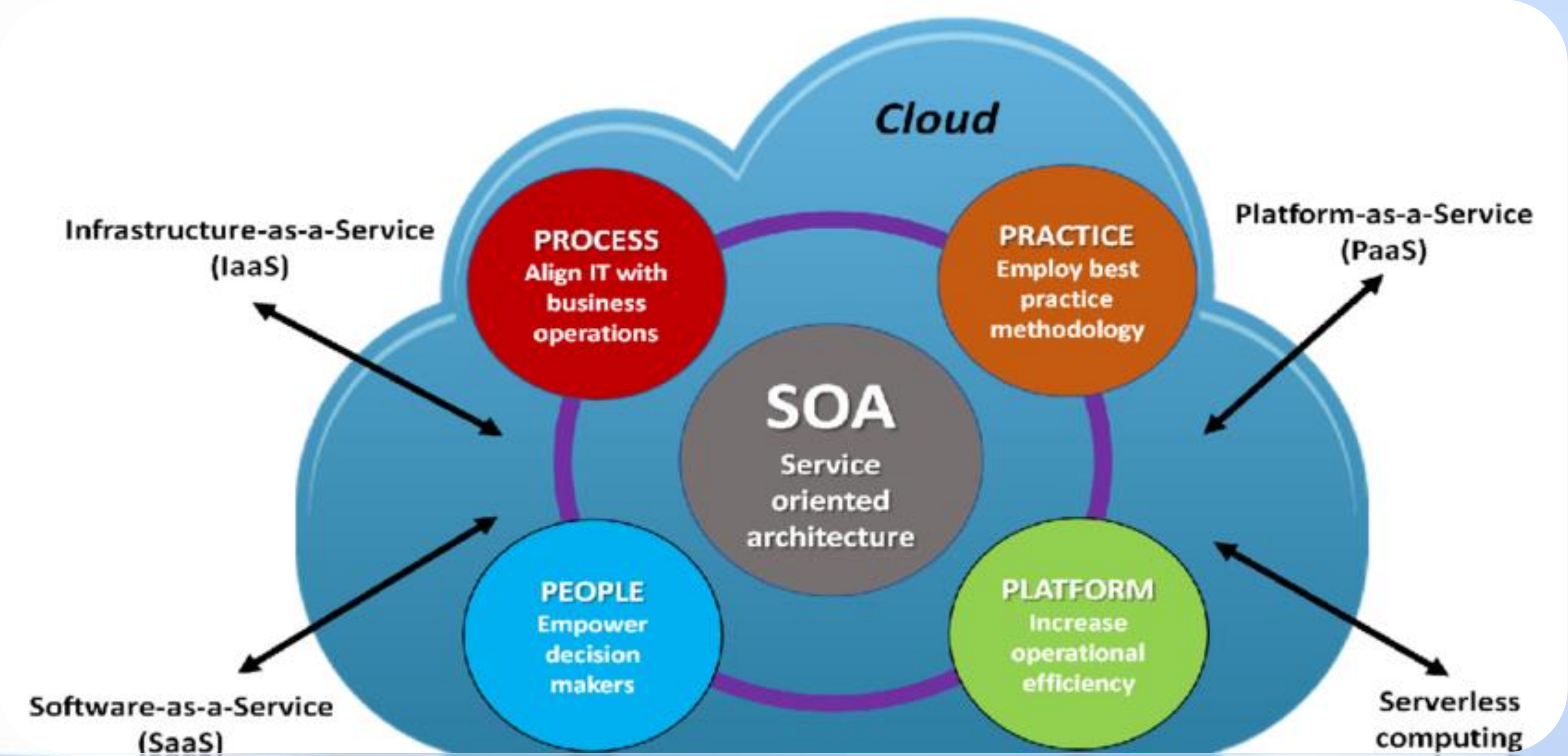




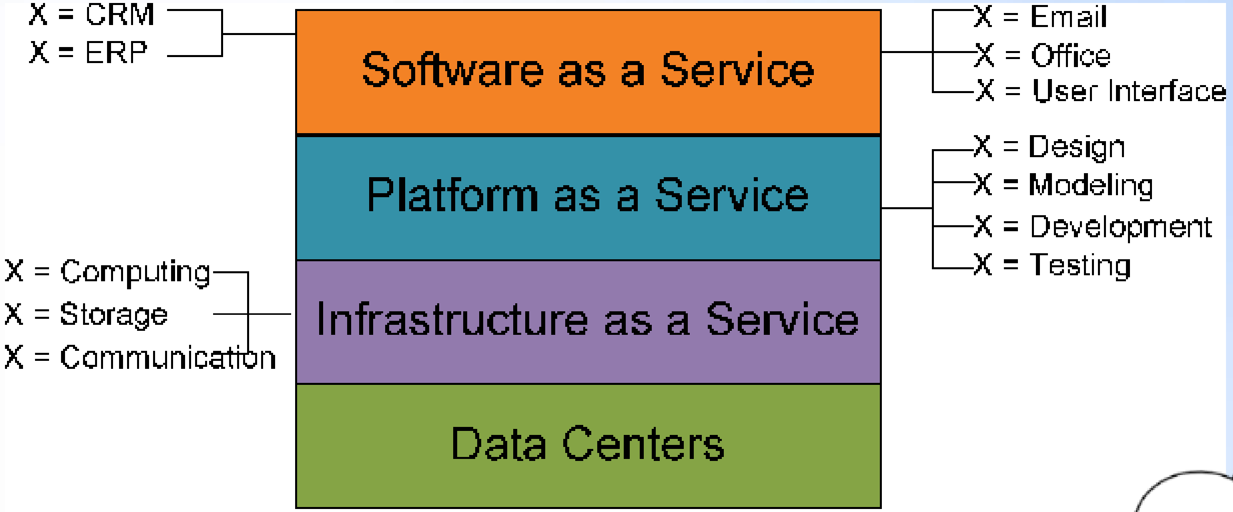
Microservice Architecture

This architecture designs a software application as a suite of independently deployable, small, modular services.

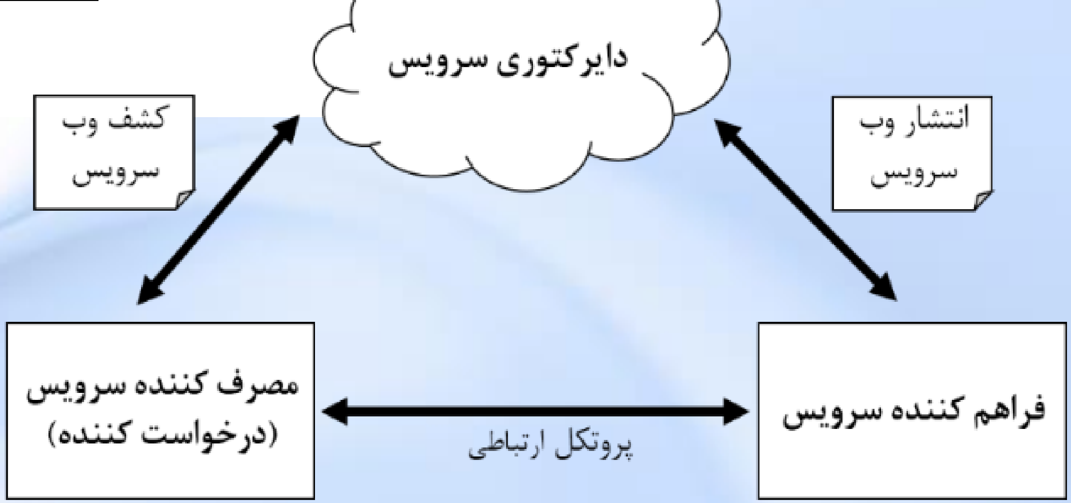
Service-Oriented Architecture (1)


Cloud
Infrastructure-as-a-Service (IaaS)
Platform-as-a-Service (PaaS)
PROCESS Align IT with business operations
PRACTICE Employ best practice methodology
SOA Service oriented architecture
PEOPLE Empower decision makers
PLATFORM Increase operational efficiency
Software-as-a-Service (SaaS)
Serverless computing

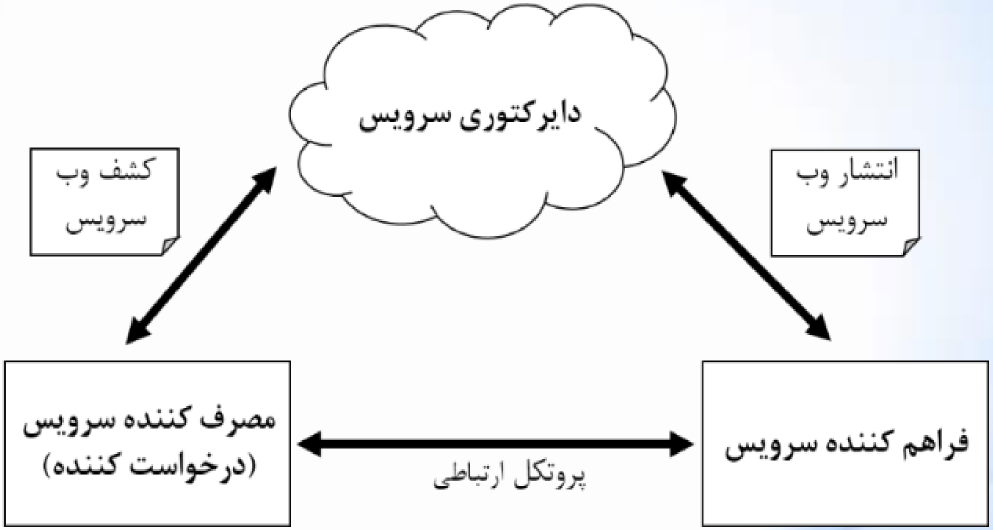
Service-Oriented Architecture (2)



Everything as a Service
XaaS X = ?



Service-Oriented Architecture (3)



معماری سرویس گرا

۱ سیستم‌های اینترنتی (تحت وب)
سیستم‌های متمرکز سازمانی

۲ سرویس و وب‌سرویس

۳ ذینفعان در سرویس‌گرایی

٤ انواع وب‌سرویس

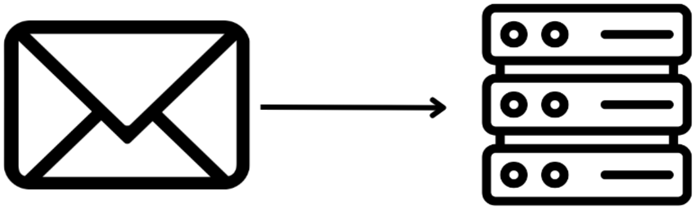
۱- ضروری، کسب‌وکار، هماهنگ‌کننده

۲- جست‌وجوگر، پردازش‌گر، محاسبه‌گر

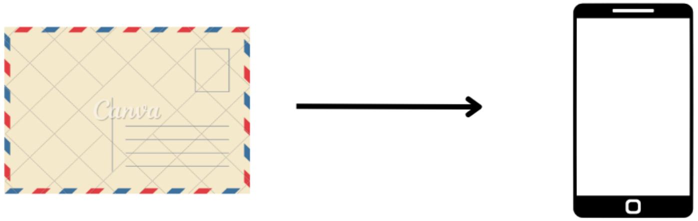
۵ تولید وب‌سرویس

Service-Oriented Architecture (4)

Simple Object Access Protocol

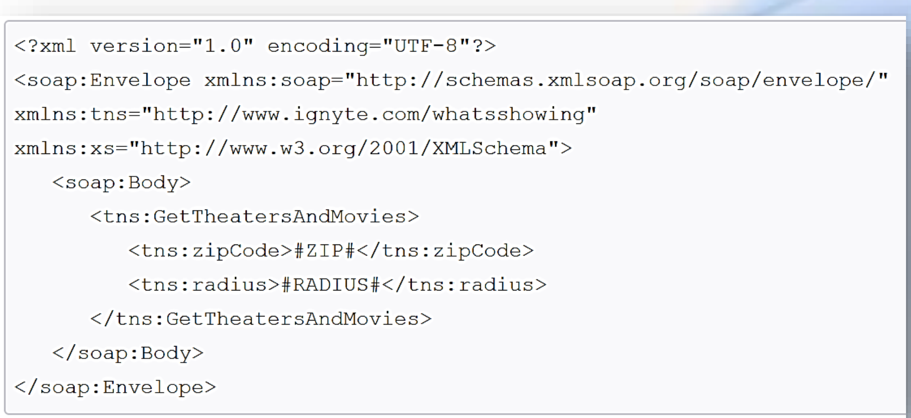


Representational State Transfer

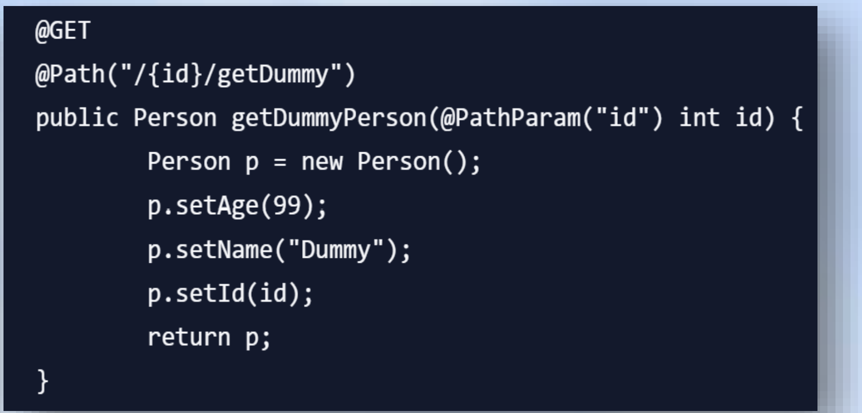


SOAP is like using an Envelope
Extra overhead, more bandwith required ,more work on both ends(sealing and opening)

REST is like a Postcard
Lighterweight, can be cached, easier to update



```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:tns="http://www.ignyte.com/whatsshowing"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
   <soap:Body>
      <tns:GetTheatersAndMovies>
         <tns:zipCode>#ZIP#</tns:zipCode>
         <tns:radius>#RADIUS#</tns:radius>
      </tns:GetTheatersAndMovies>
   </soap:Body>
</soap:Envelope>
```



```
@GET
@Path("/{id}/getDummy")
public Person getDummyPerson(@PathParam("id") int id) {
        Person p = new Person();
        p.setAge(99);
        p.setName("Dummy");
        p.setId(id);
        return p;
}
```

End of Session 16

Any Question?!

THANKS FOR
YOUR
ATTENTION