الجامعة السورية الخاصة
SYRIAN PRIVATE UNIVERSITY

Topic 6

Software Architecture

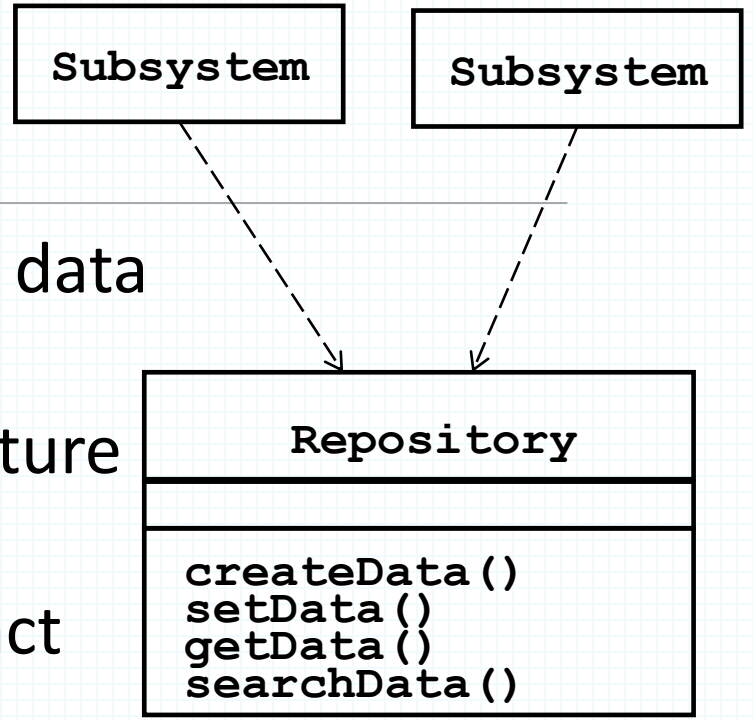
More Architecture Styles

Dr. Riad Sonbol

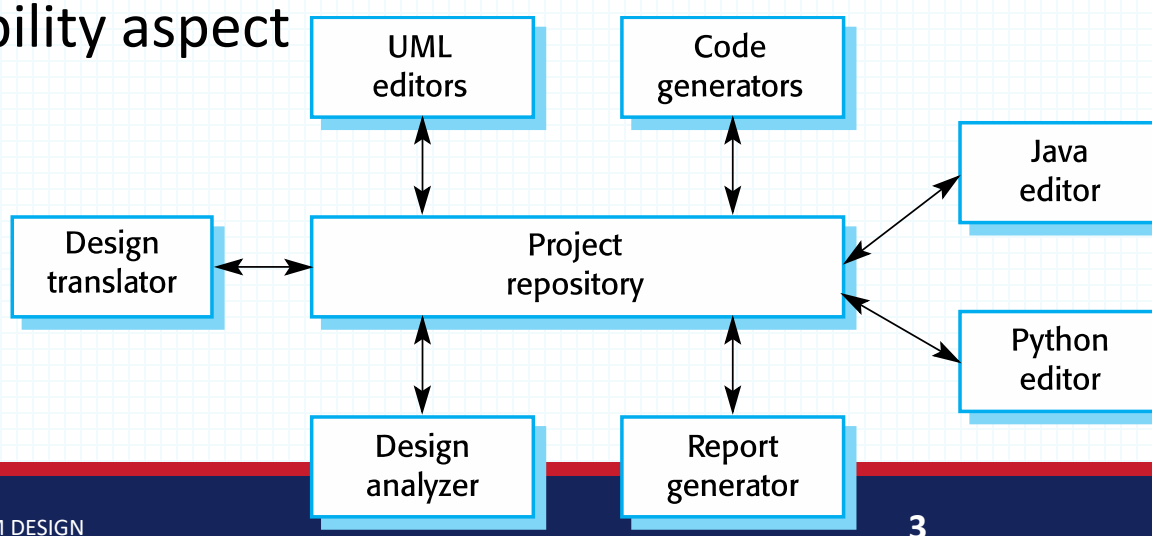
Course Roadmap

- General Principles in Software Design and Architecture: *The Context of Software Architecture, System Decomposition, What is Good Design?*
- Layered Architecture
- Practical Concerns 1: Communication between Layers, Façade Design Pattern, and Dependency Injection
- N-Tier, Client-server, Peer-to-Peer, MVC
- Microservices
- Practical Concerns 2: Message Broker, Related Design Patterns: CQRS, API Gateway, Strangler Pattern, Externalized Configuration, case study
- Clean Architecture: The Hexagonal Architecture, Ports and Adapters
- More Architectural Styles: Repository, Service-Oriented Architecture (SOA), Pipes and Filters

Repository Architectural Style

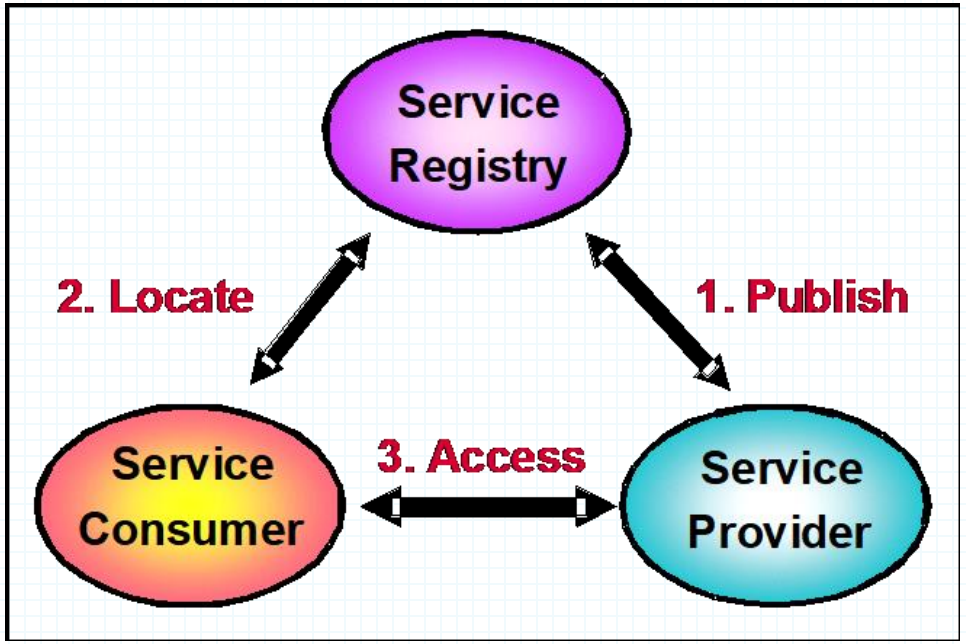


- Subsystems access and modify data from a single data structure called the repository.
- subsystems access and modify a single data structure called the central repository.
- Subsystems are relatively independent and interact only through the repository.
  - the central repository can quickly become a bottleneck, both from a performance aspect and a modifiability aspect
- Sample Application?
  - DB management.
  - IDE (save artifacts which working on software projects management)



Service-Oriented Architecture (SOA) Architectural Style

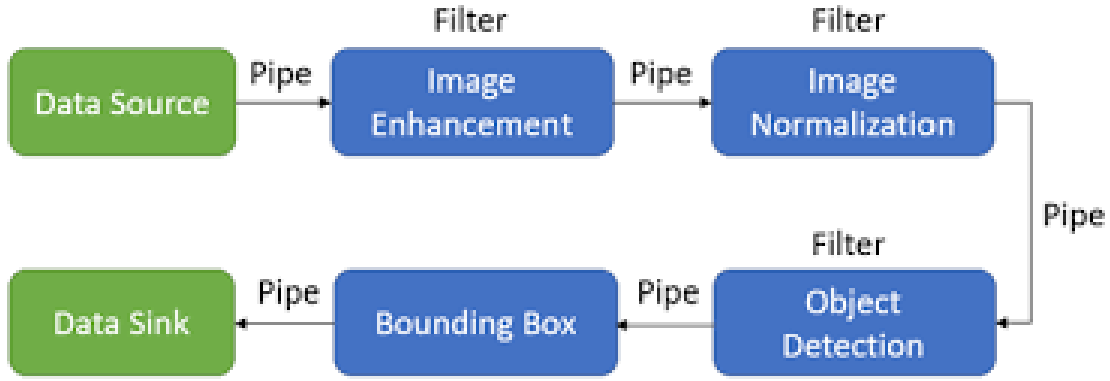
- SOA is an architectural approach in which applications make use of services available in the network.
- Each service in SOA is a complete business function in itself
- Usually, the services are published in such a way that it makes it easy for the developers to assemble their apps using those services
- Standard protocols have been developed to support service communication and information exchange



Three key roles: consumer, provider, registry

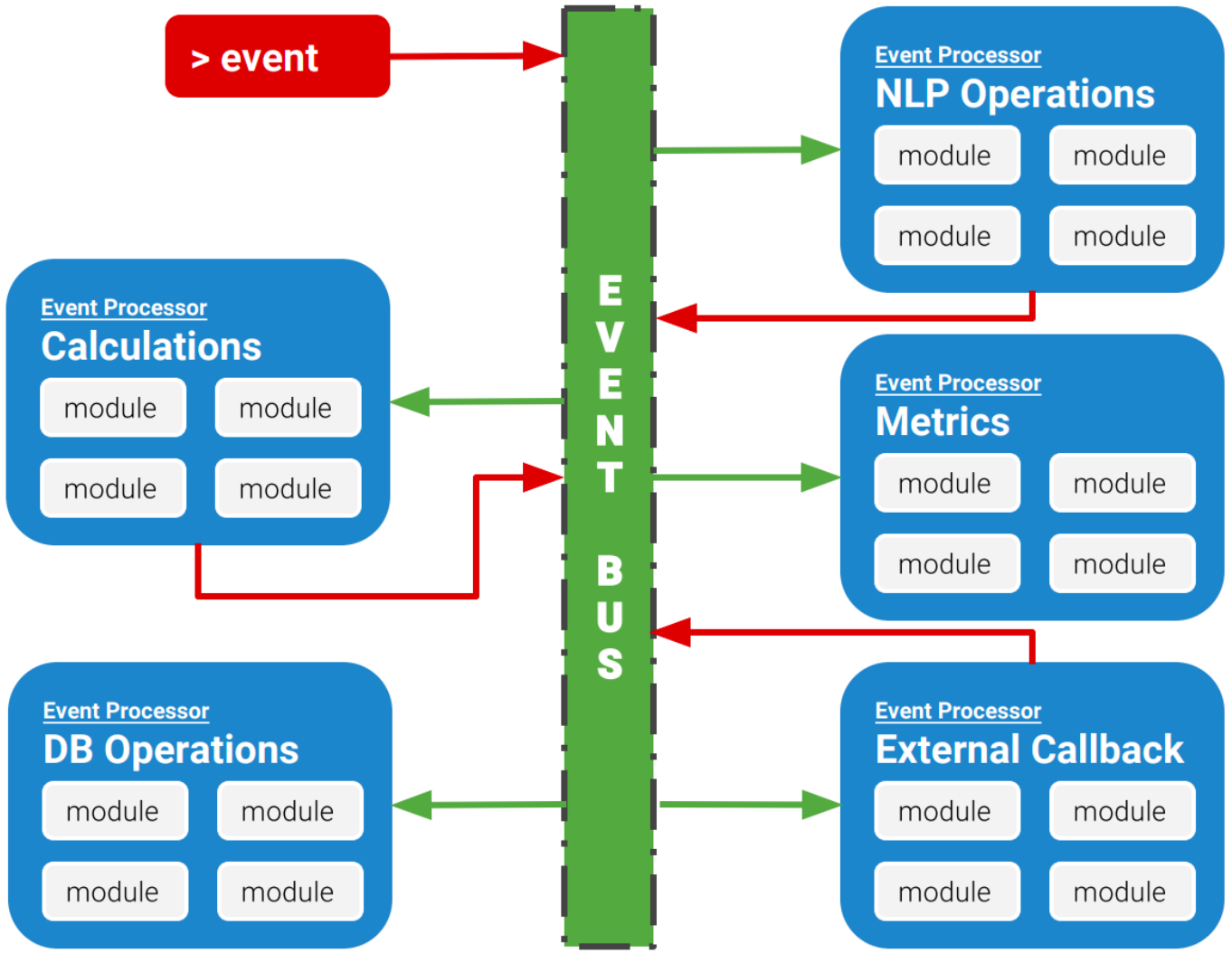
Pipes and Filters

- A pipeline consists of a chain of processing elements (processes, threads, etc.), arranged so that the output of one element is the input to the next element
  - Usually some amount of buffering is provided between consecutive elements
  - The information that flows in these pipelines is often a stream of records, bytes or bits.



Event-Driven Architecture

- Event-Driven Architecture is a **communication style** where:
  - Components communicate by publishing events
  - Other components subscribe and react
  - Communication is asynchronous
- **Key Characteristics**
  - Loose coupling
  - Asynchronous messaging
  - Event broker (Kafka, RabbitMQ, etc.)



Event-Driven vs Microservices

- Microservices is a system structure, Event-Driven Architecture (EDA) is a communication style.
- You can have:
  - **Microservices WITHOUT EDA**
    - REST calls between services
    - Synchronous request/response
    - Tight runtime coupling
  - **EDA WITHOUT Microservices**
    - A monolith using internal event bus
    - Modular monolith reacting to events
  - **Microservices + EDA (Modern Cloud Systems)**
    - Most scalable pattern
    - But most complex

Example

- **Can you recommend an architectural design** for a system that enables users to upload images, apply various computer vision functionalities offered by the system, and subsequently save and edit the resulting images? Additionally, the system should encompass features for account management and billing to handle user subscriptions effectively.