

Architectural Design

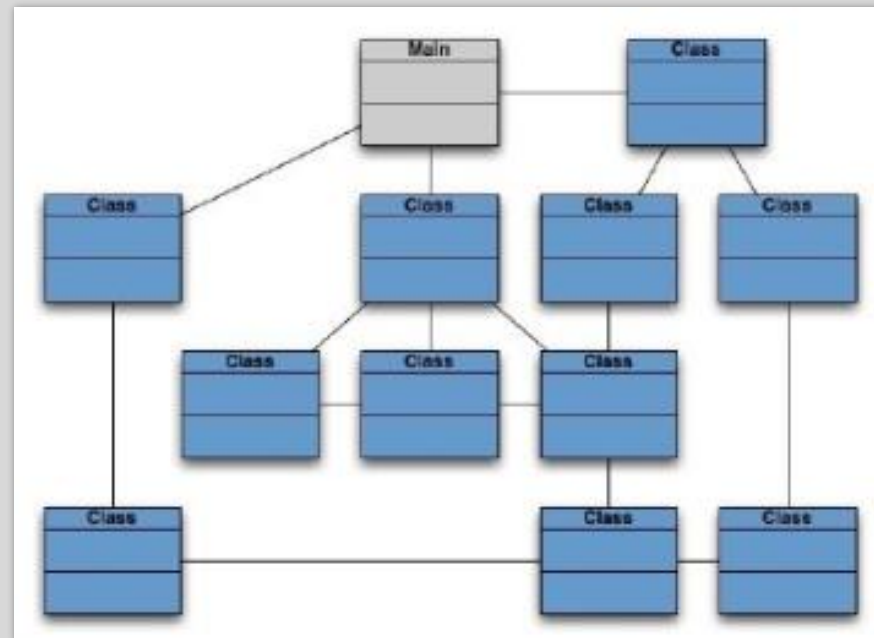
Lecture # 27



Call and Return Architecture

Object-oriented architectures

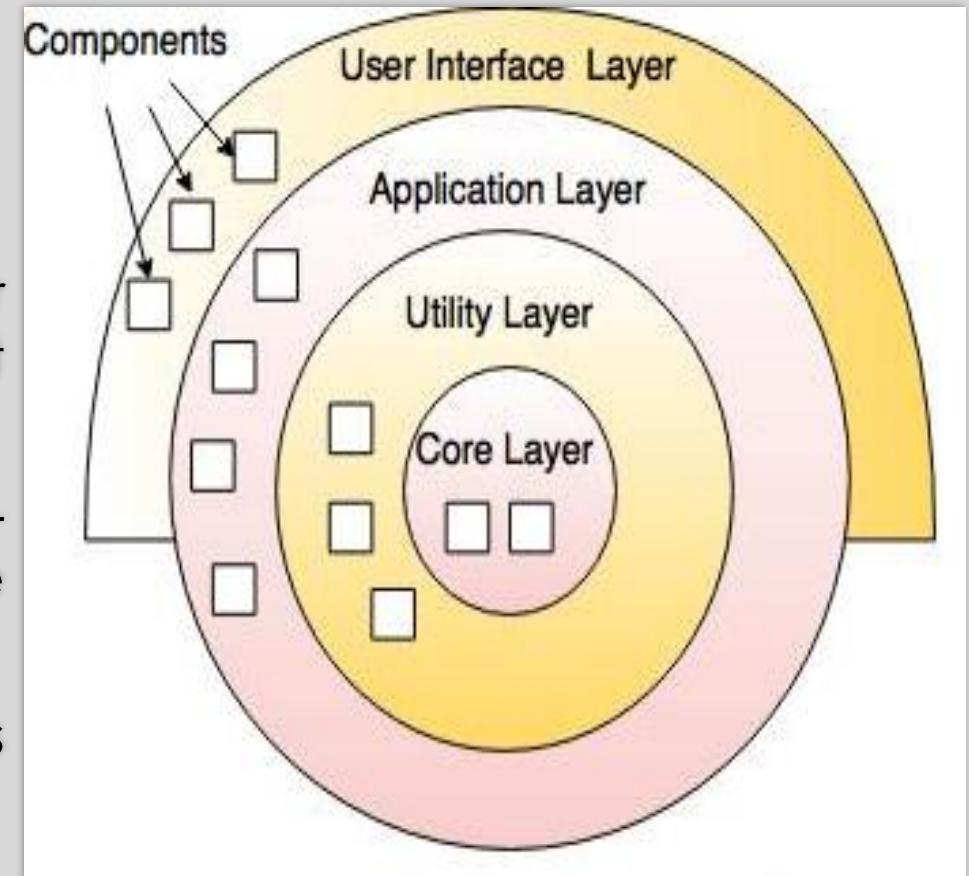
- This architecture is the latest version of call-and-return architecture.
- It consist of the bundling of data and methods.
- It is a design paradigm based on the division of responsibilities for an application or system into individual reusable and self-sufficient objects.



Call and Return Architecture

○ Layered Architecture

- Used to model the interfacing of sub-systems.
- Organizes the system into a set of layers (or abstract machines) each of which provide a set of services.
- Supports the incremental development of sub-systems in different layers. When a layer interface changes, only the adjacent layer is affected.
- However, often artificial to structure systems in this way.



A Generic Layered Architecture

User interface

User interface management
Authentication and authorization

Core business logic/application functionality
System utilities

System support (OS, database etc.)



The Architecture of LIBSYS System



Architecture Reuse

- Systems in the same domain often have similar architectures that reflect domain concepts.
- Application product lines are built around a core architecture with variants that satisfy particular customer requirements.
- The architecture of a system may be designed around one of more architectural patterns or styles.
 - These capture the essence of an architecture and can be instantiated in different ways.



Architectural Patterns

- Patterns are a means of representing, sharing and reusing knowledge.
- An architectural pattern is a stylized description of good design practice, which has been tried and tested in different environments.
- Patterns should include information about when they are and when they are not useful.
- Patterns may be represented using tabular and graphical descriptions



Architectural Patterns (Example)

- **Concurrency**—applications must handle multiple tasks in a manner that simulates parallelism
 - operating system process management pattern
 - task scheduler pattern
- **Persistence**—Data persists if it survives past the execution of the process that created it. Two patterns are common:
 - a database management system pattern that applies the storage and retrieval capability of a DBMS to the application architecture
 - an application level persistence pattern that builds persistence features into the application architecture
- **Distribution**— the manner in which systems or components within systems communicate with one another in a distributed environment
 - A broker acts as a 'middle-man' between the client component and a server component.

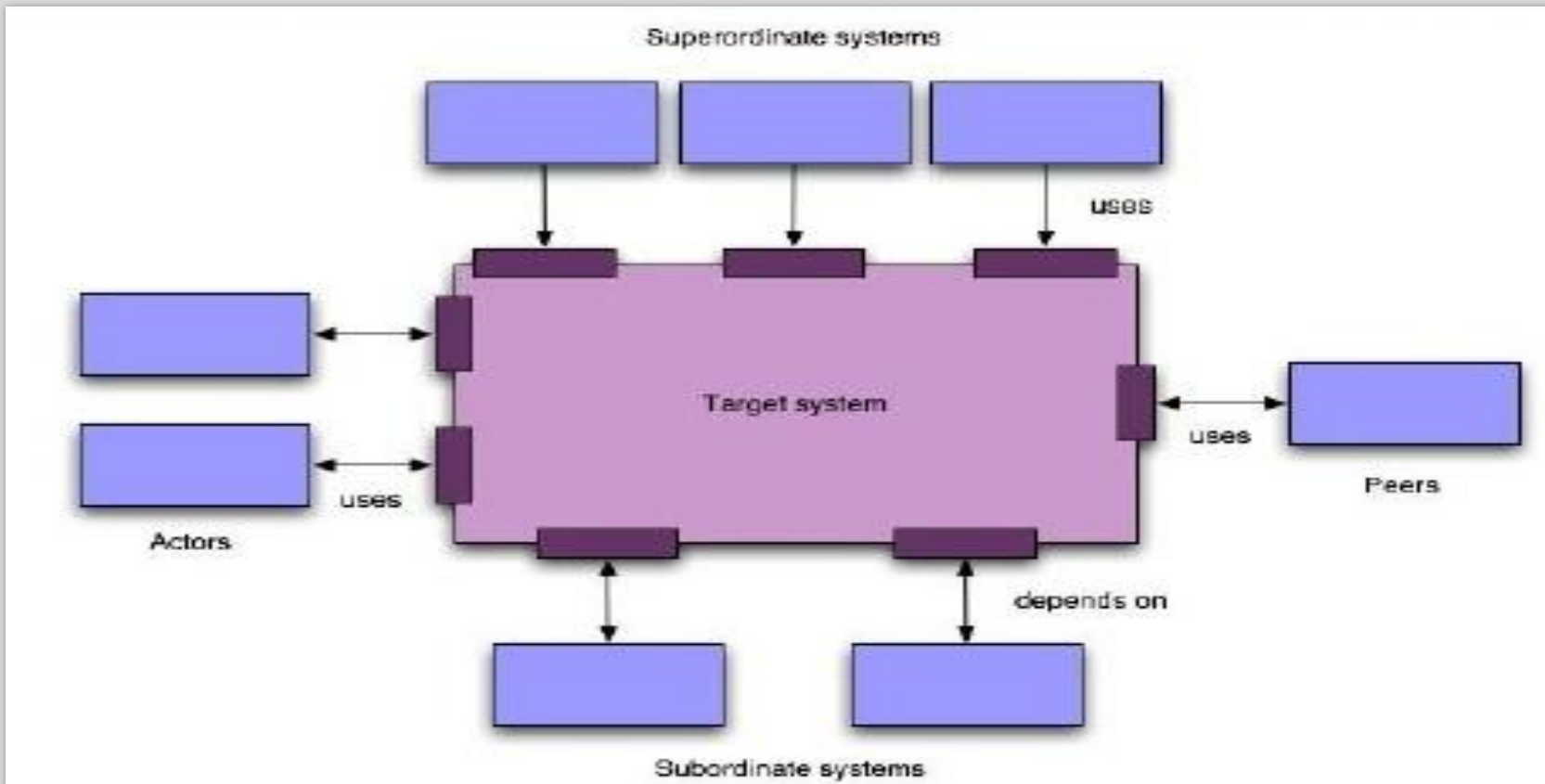


Architectural Design

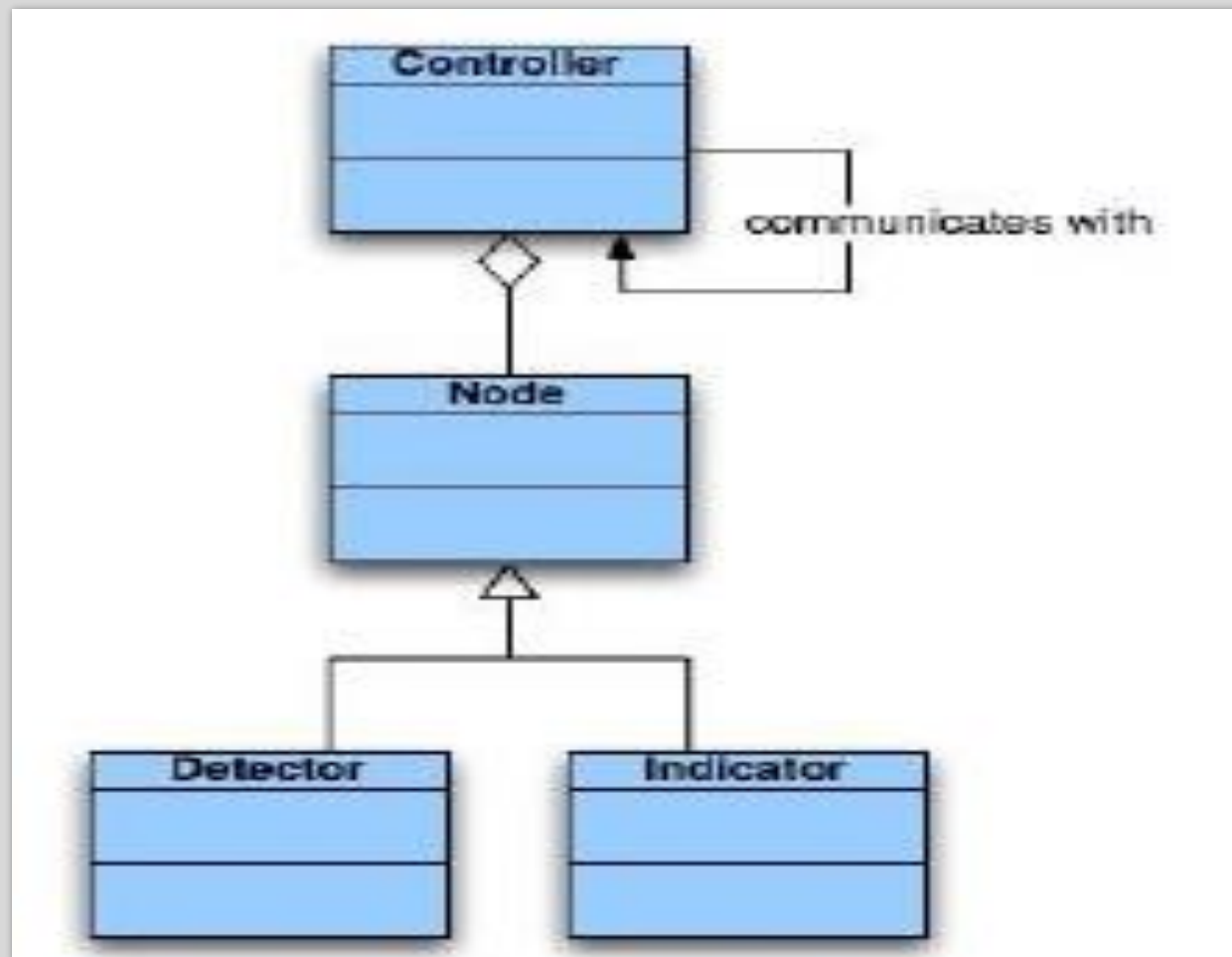
- Architectural context diagrams model how software interacts with external entities
- Archetypes are classes or patterns that represent an abstraction critical to the system
- Architectural components are derived from the application domain, the infrastructure, and the interface.



Arch. Context Diagram



Archetype



Component Structure

