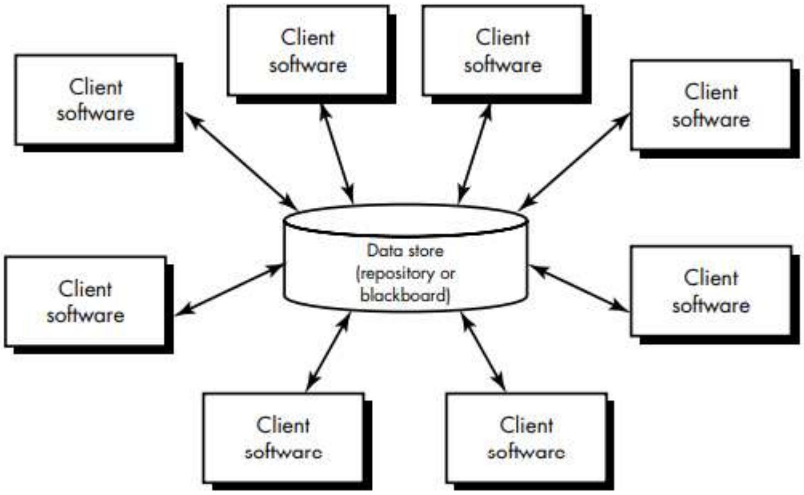
Unit: II
Lecture: 6
Architectural Styles

The various architectural styles are to establish a structure for all the components of the system. The software that is built for computer-based systems can exhibit one of many architectural styles.
Each architectural style describes a system category that encompasses:

- a set of components
- a set of connectors that enables "communication and coordination"
- constraints that define how components can be integrated to form the system.
- semantic models to understand the overall properties of a system.

Various Architectural Styles:

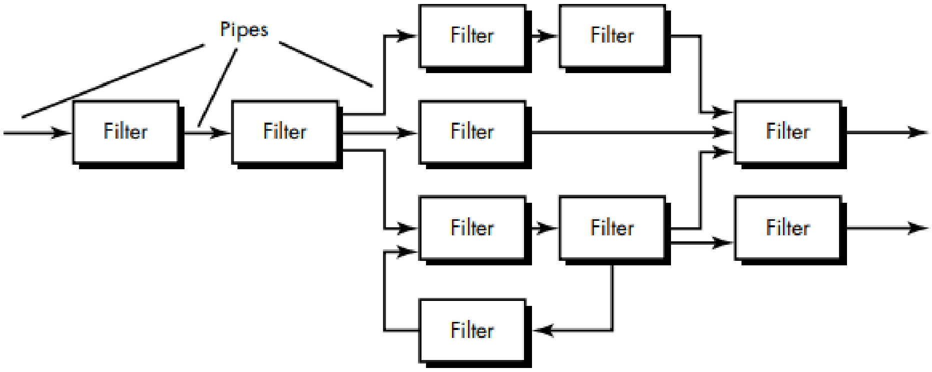
1. **Data Centered Architectures:**
   - A data store will reside at the centre of this architecture and is accessed frequently by the other components that update, add, delete or modify the data present within the store.
   - The figure illustrates a typical data centered style. The client software access a central repository.



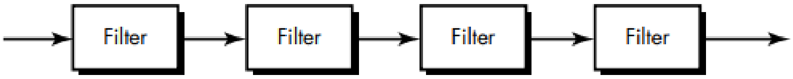
   - The data-centered architecture promotes integrability i.e. the existing components can be changed and a new client component can be added to the architecture without the permission or concern of other clients.

2. Data flow architectures:

- This kind of architecture is used when input data to be transformed into output data through a series of computational manipulative components.



(a) Pipes and filters



(b) Batch sequential

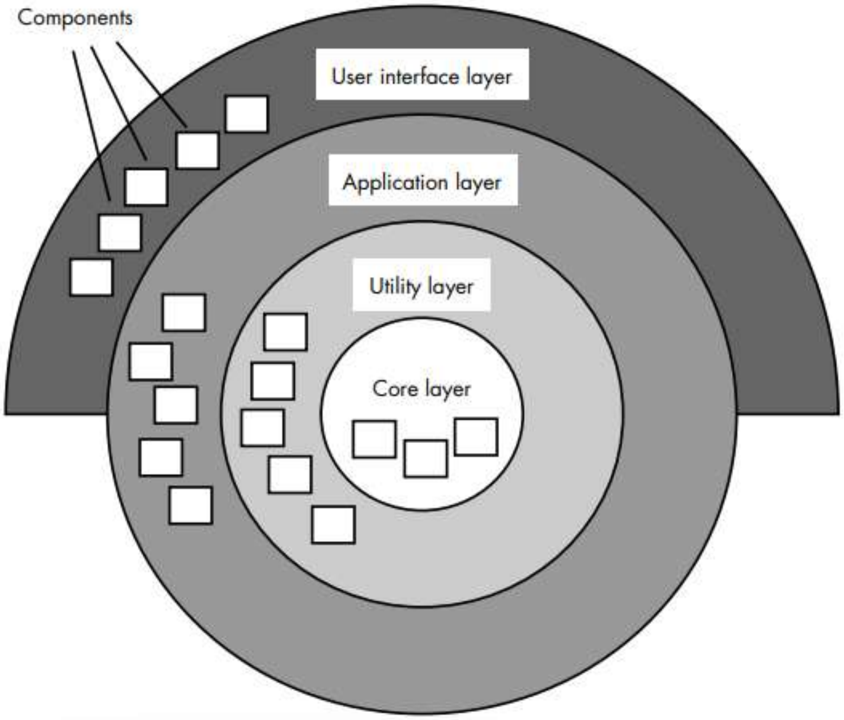
- A pipe and filter pattern (shown Fig (a)) has set of components, called filters, connected by pipes that transmit data from one component to the next.
- Each filter works independently of those components upstream and downstream, is designed to expect data input of a certain from and produces data output (to the next filter) of a specified form.
- If the data flow degenerates into a single line of transforms (Fig(b)), then it is termed as a batch sequential. This structure accepts the batch of data and then applies a series of sequential components to transform it.

3. Call and Return Architectures:

- It is used to create a program that is easy to scale and modify. A number of substyles exist within this category:
- **Main program/sub program architectures:** This classic program structure decomposes function into a control hierarchy where a main program invokes a number of program components, which in turn may invoke still other components.
- **Remote procedure call architectures:** The components of a main program/ subprogram architecture are distributed across multiple computers on a network.

4. **Object-oriented Architectures:**
    - The components of a system encapsulate data and the operations that must be applied to manipulate the data. Communication and coordination between components is accomplished via message passing.

5. **Layered Architectures:**
    - A number of different layers are defined with each layer performing a well-defined set of operations. Each layer will do some operations that becomes closer to machine instruction set progressively.
    - At the outer layer, components will receive the user interface operations and at the inner layers, components will perform the operating system interfacing (communication and coordination with OS).
    - Intermediate layers provide utility services and application software functions.


Components
User interface layer
Application layer
Utility layer
Core layer