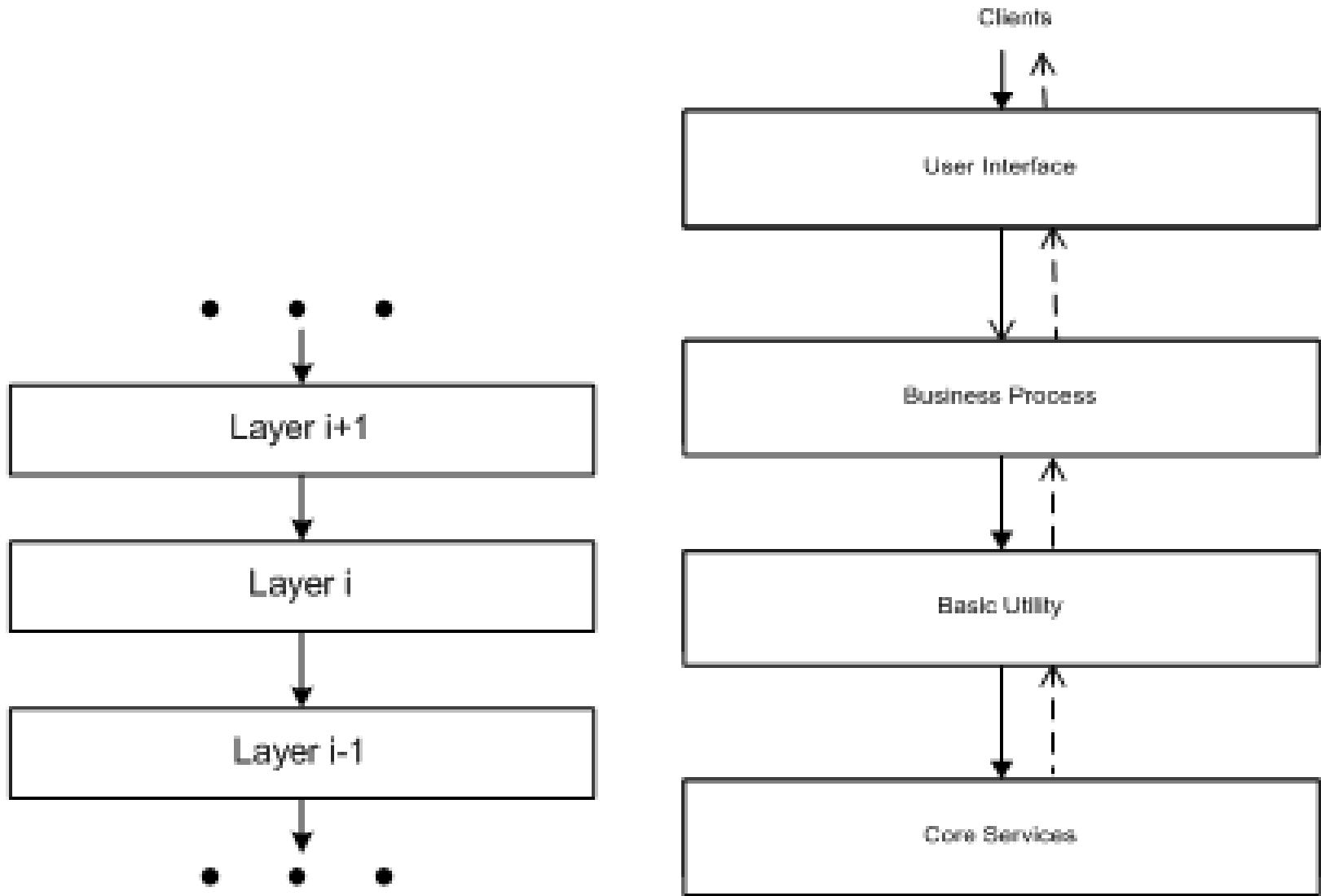


# Software Design and Architecture

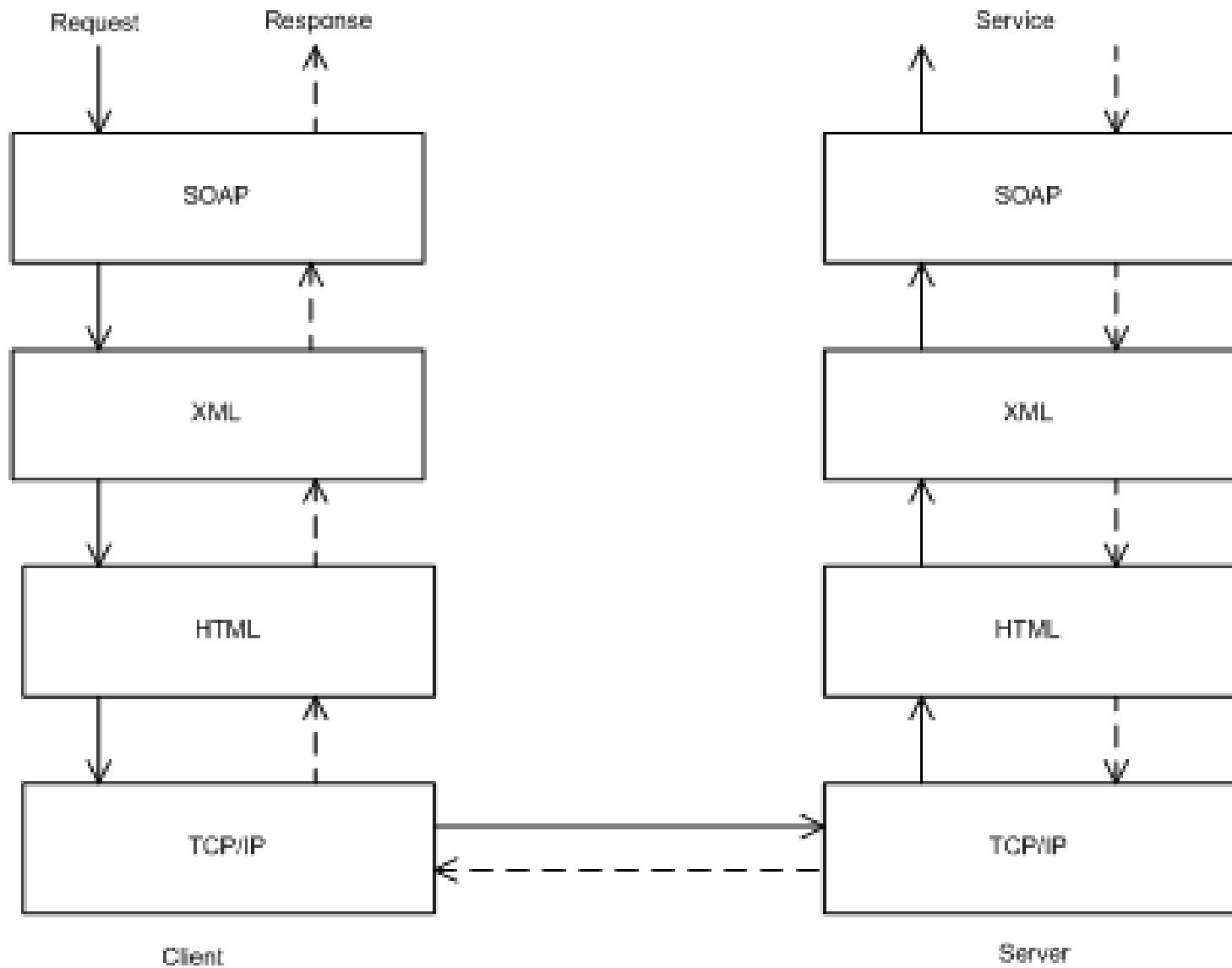
Layered Architecture

# Layered Architecture

- ▶ As its name suggests, in a layered architecture the system is decomposed into a number of higher and lower layers in a hierarchy; each layer consists of a group of related classes that are encapsulated in package, in a deployed component, or as a group of subroutines in the format of method library or header file.
- ▶ Also, each layer has its own sole responsibility in the system.



- ▶ A simple software system may consist of two layers: Interaction Layer and Processing Layer:
- ▶ The Interaction layer provides user Interfaces to clients, takes requests, validates and forwards request to processing layer for processing, and responds to clients.
- ▶ The Processing layer gets the forwarded requests and performs business logic process, accesses database, returns the results to its upper layer, lets upper layer respond to clients since the upper layer has the GUI interface responsibility.



# Applicable Design Domains

- ▶ Any system that can be divided between the application specific portions and platform specific portions which provide generic services to the application of the system.
- ▶ Applications that have clean divisions between core services, critical services, user interface services, etc.
- ▶ Applications that have a number of classes that are closely related to each other so that they can be grouped together into a package to provide the services to others.

# Benefits

- ▶ Incremental software development based on increasing levels of abstraction.
- ▶ Enhanced independence of upper layer to lower layer since there will be no impact from the changes of lower layer services as long as their interfaces remain unchanged.
- ▶ Enhanced flexibility: interchangeability and reusability are enhanced due to the separation of the standard interface and its implementation.
- ▶ Component-based technology is a suitable technology to implement the layered architecture; this makes the system much easier to allow for plug-and-play of new components.
- ▶ Promotion of portability: each layer can be an abstract machine deployed independently.

# Limitations

- ▶ Lower runtime performance since a client's request or a response to client must go through potentially several layers. There are also performance concerns on overhead on the data marshaling and buffering by each layer.
- ▶ Breach of interlayer communication may cause deadlocks and “bridging” may cause tight coupling.
- ▶ Exceptions and error handling is an issue in the layered architecture, since faults in one layer must propagate upwards to all calling layers.

# Summary

- ▶ Introduction to layered architecture
- ▶ Benefits of layered architecture
- ▶ Limitations of layered architecture