CT30A3401
Distributed Systems
Lecture 3

Bilal Naqvi, PhD.
syed.naqvi@lut.fi

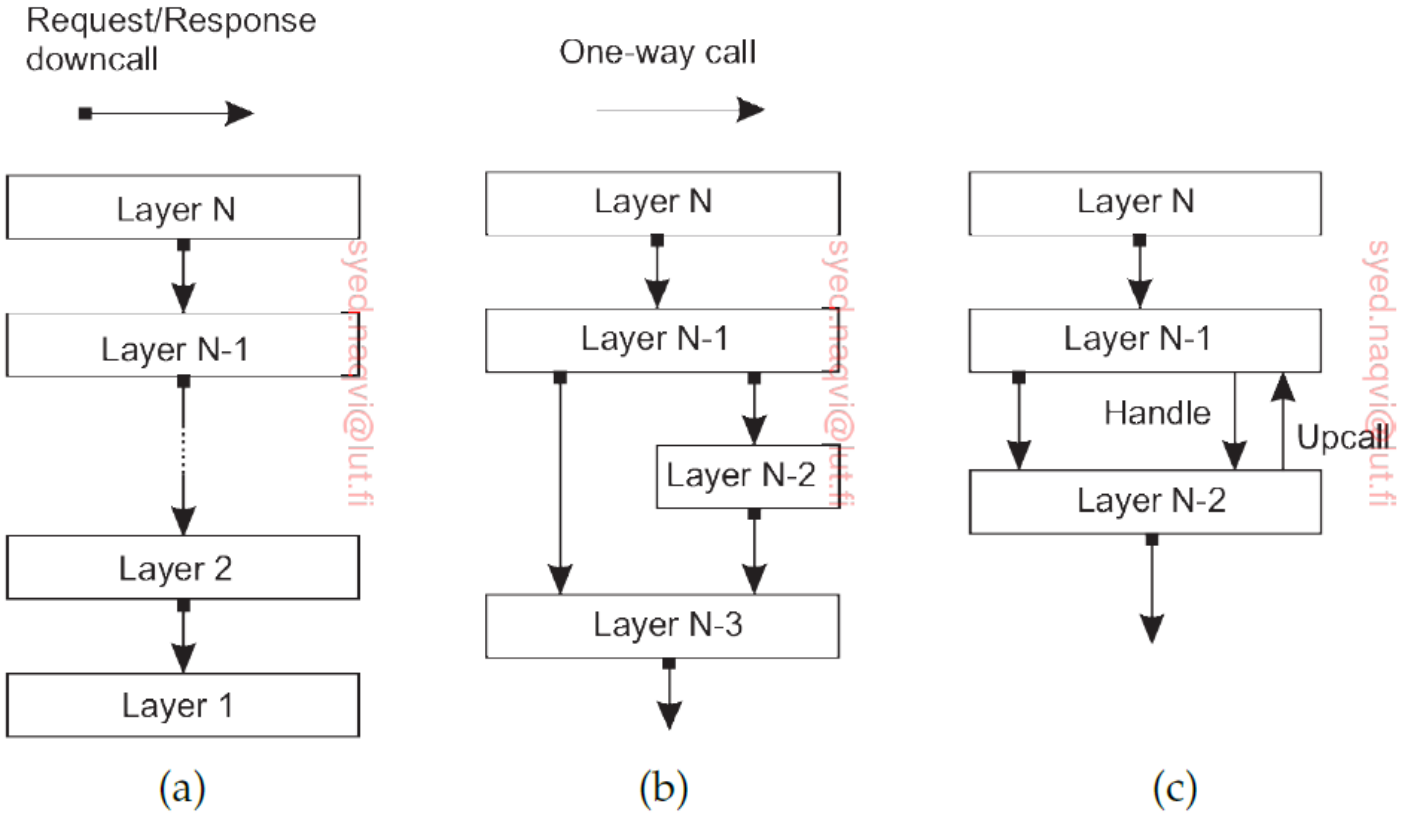
Distributed System Architectures

- Distributed systems are often complex pieces of software which run across multiple machines
  - need for proper organization due to complexity
- Different ways to organize (aka *Architectural styles*)
  - Layered architectures
  - Object-based architectures
  - Resource-centered architectures
  - Event-based architectures

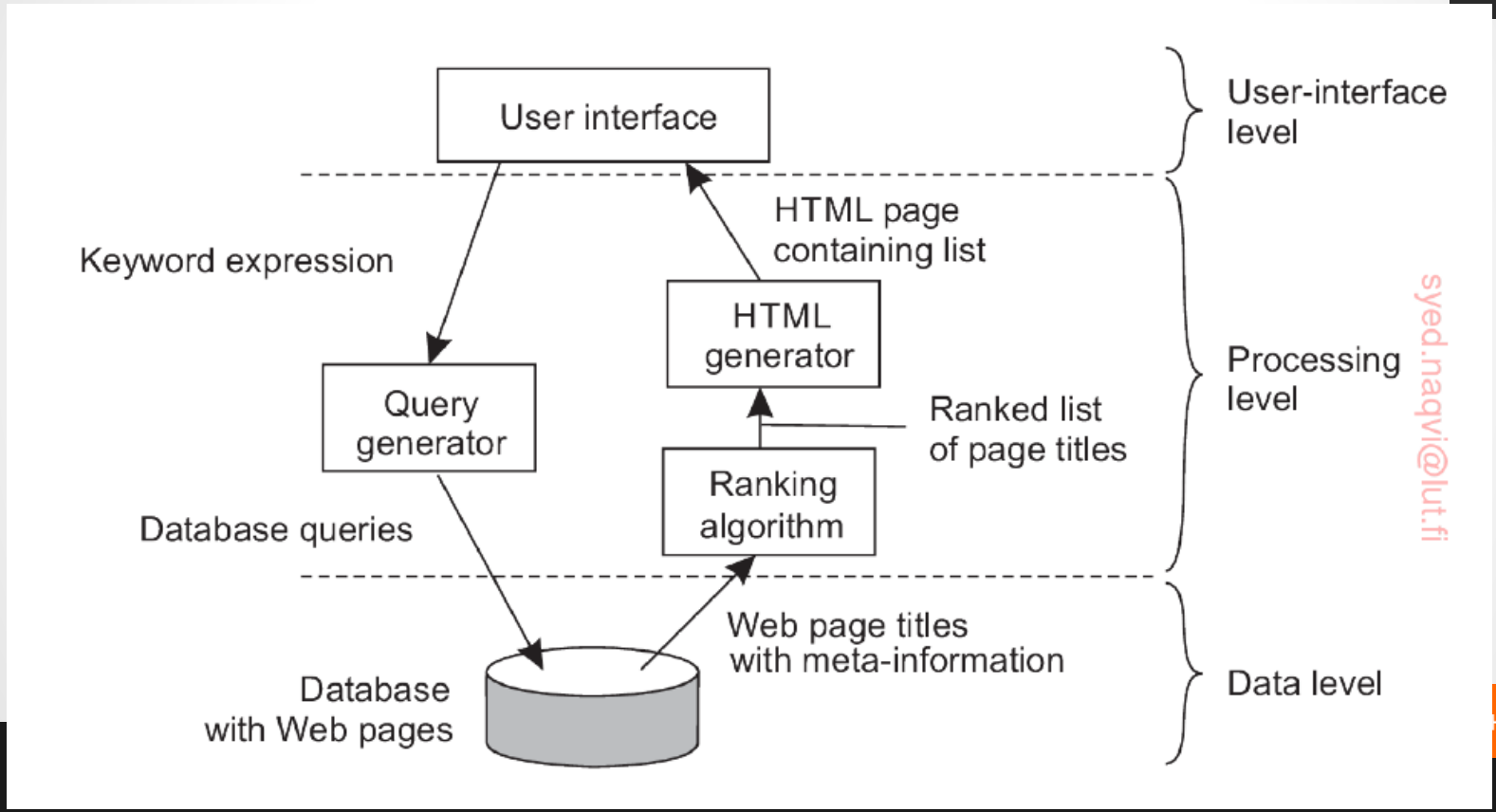
Layered architectures

- Components are organized in a layered fashion
- Principle: Each component at layer L_j can make a down call to a component at a lower-level layer L_i (with $i < j$) and generally expects a response
- Only in exceptional cases will an upcall be made to a higher-level component

Layered architectures



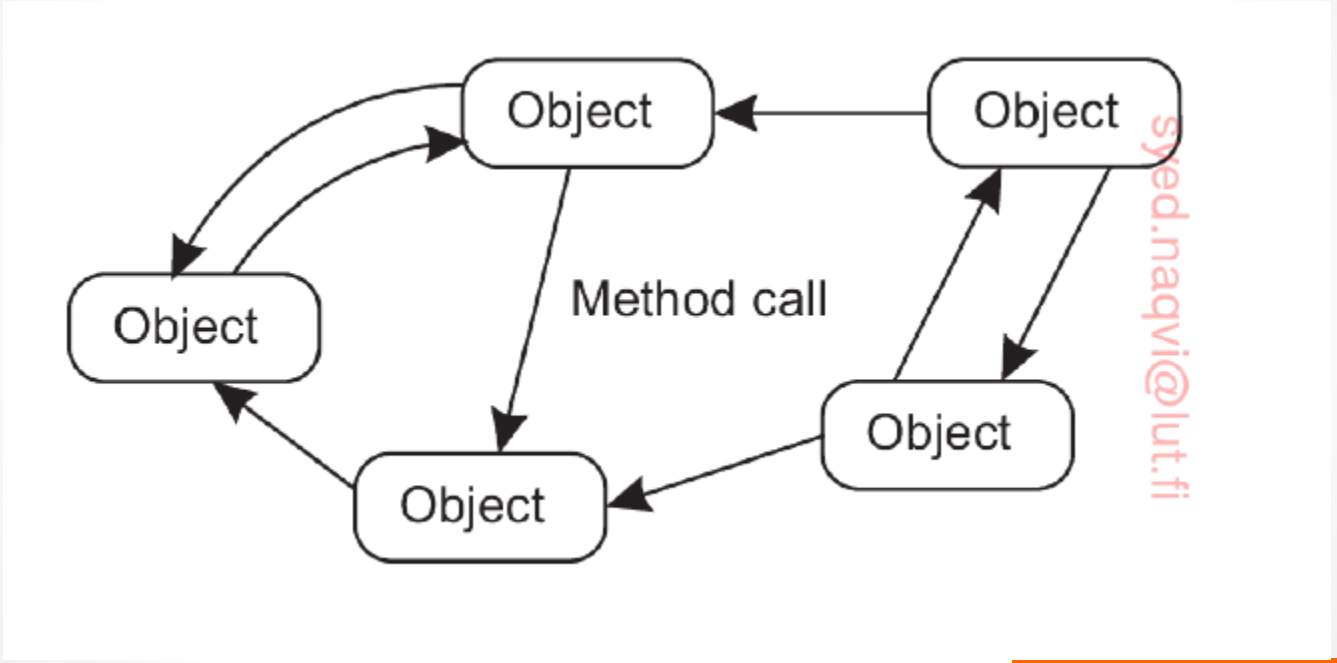
Example: Internet Search Engine



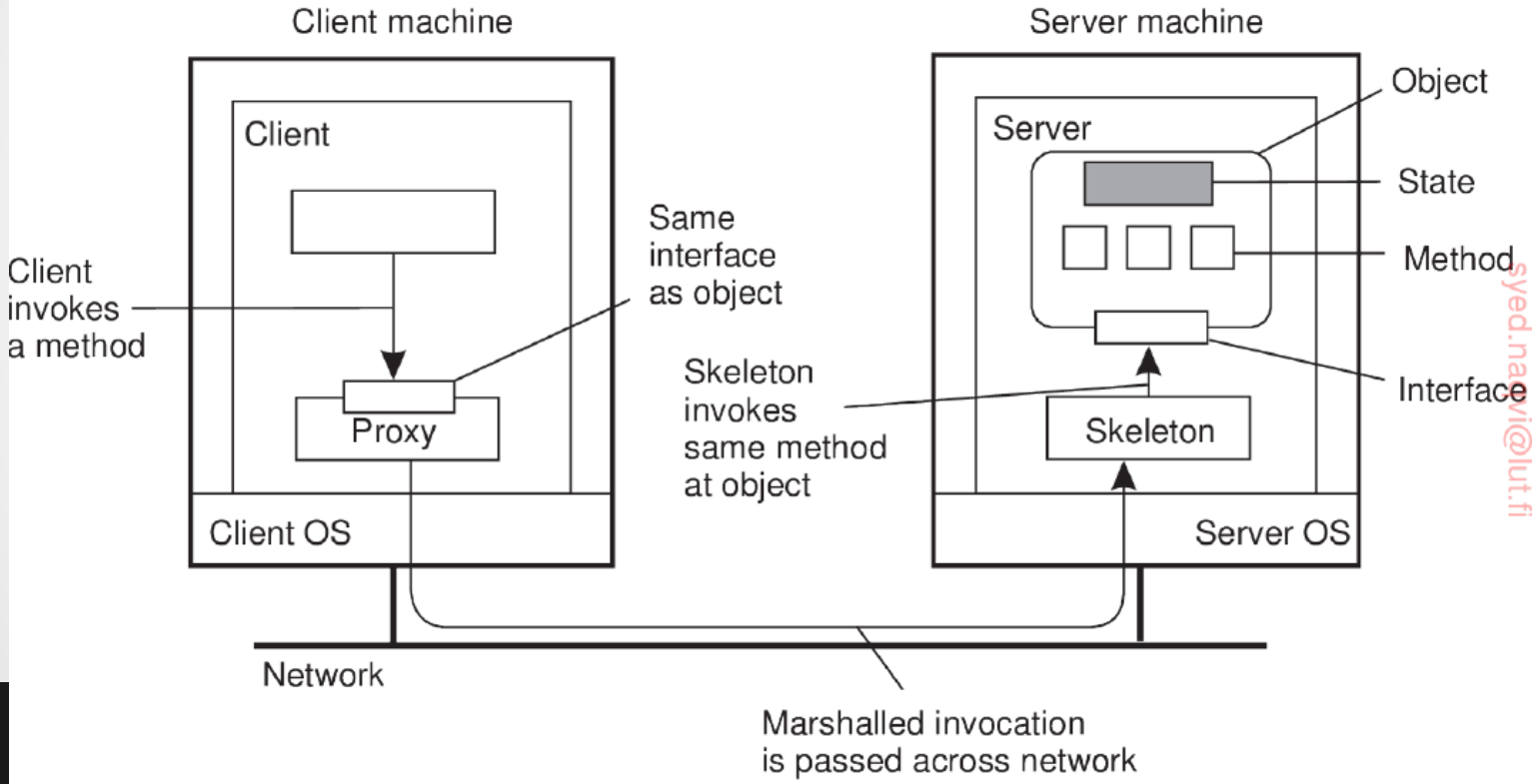
Object-based and service-oriented architectures

- Each object corresponds to a component, and these components are connected through a procedure call mechanism
- A procedure call takes place over a network
- The calling object need not be executed on the same machine as the called object

Object-based and service-oriented architectures



Object-based and service-oriented architectures (Operation)



Resource-based architectures

- Mostly used for Web-based distributed systems
- Resources may be added or removed by (remote) applications, and likewise can be retrieved or modified.
  - implemented for web and known as Representational State Transfer (REST)

RESTful architecture

- Resources are identified through a single naming scheme
- All services offer the same interface, consisting of at most four operations
- Messages sent to or from a service are fully self-described
- After executing an operation at a service, that component forgets everything about the caller (stateless execution)
  - unlike object-oriented architectures

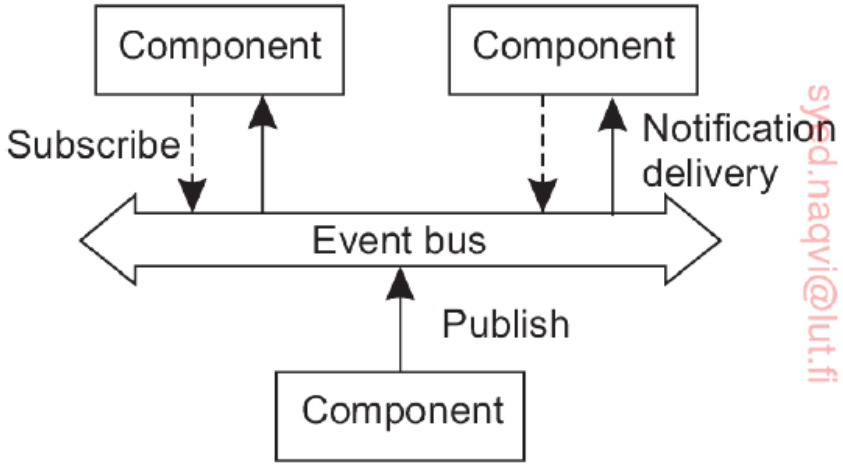
RESTful architectures common operations

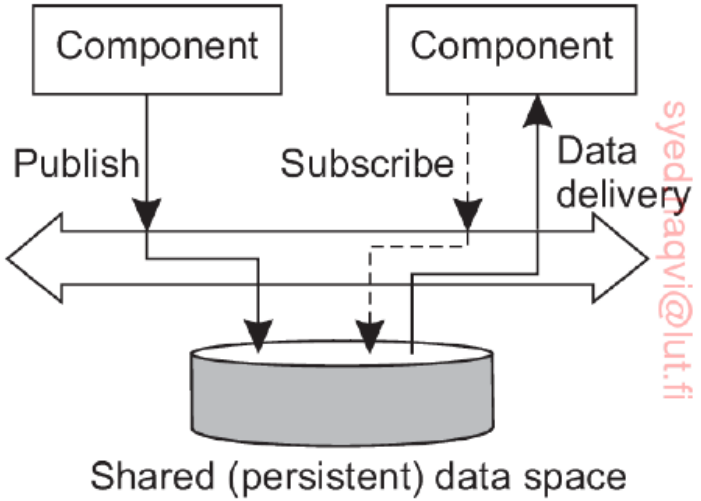
Operation	Description
PUT	Create a new resource
GET	Retrieve the state of a resource in some representation
DELETE	Delete a resource
POST	Modify a resource by transferring a new state

Event-based architectures

- Event-based coordination
- Process can publish a notification describing the occurrence of an event, assuming that notifications come in all sorts and kinds, processes may subscribe to a specific kind of notification

Event-based architectures



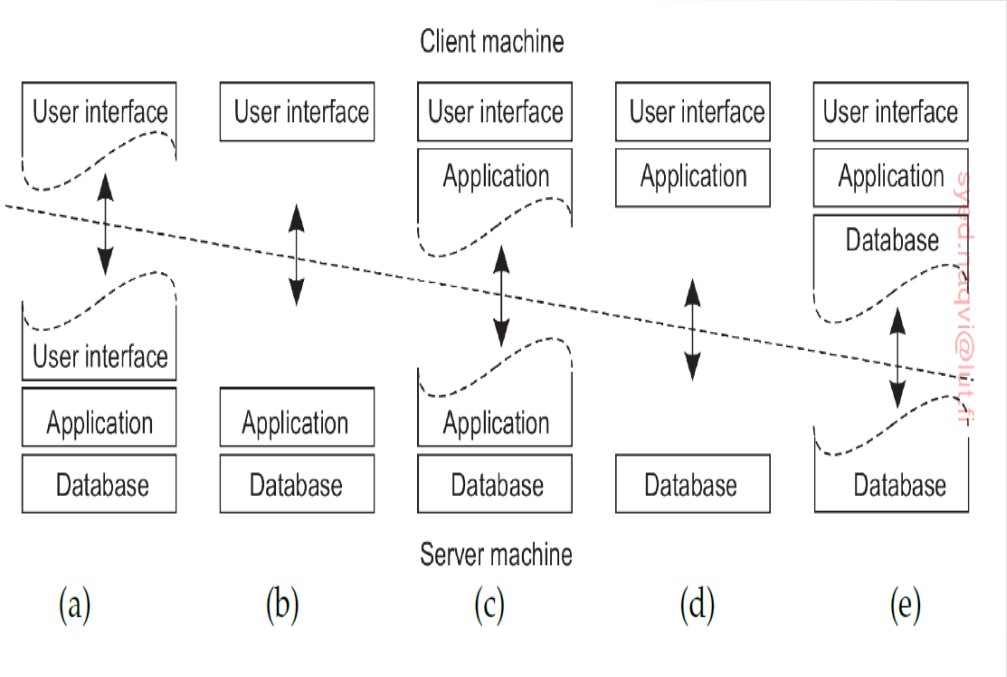


System Architectures

- Centralized organizations
- Decentralized organizations
- Hybrid Architectures

Centralized organizations

- Client-server architecture
  - server is a process implementing a specific service
  - client is a process that requests a service from a server
- Can be simple/multi-tiered

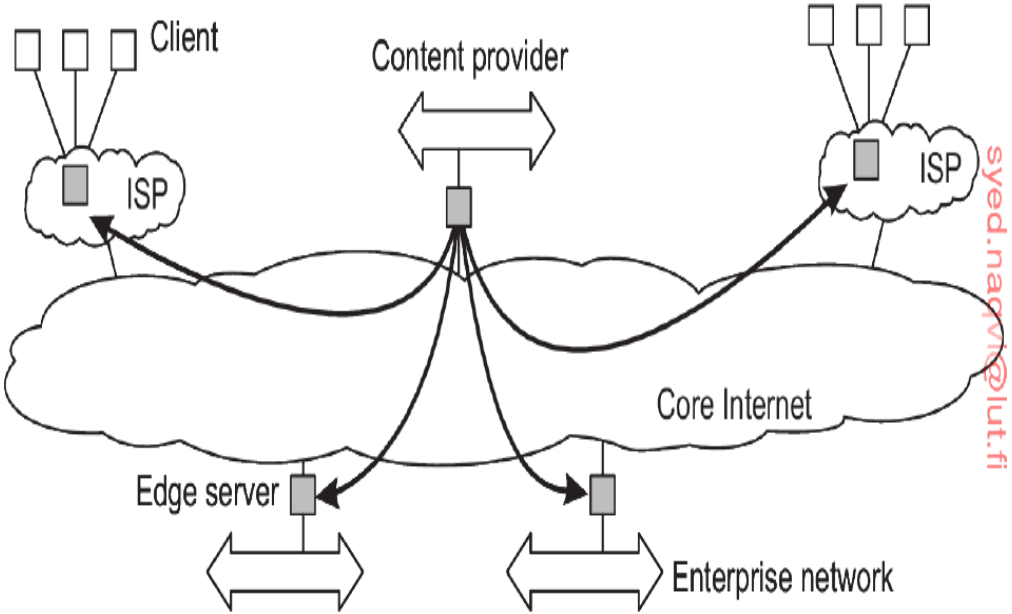


Decentralized organizations

- Peer-to-peer systems
  - each process will act as a client and a server at the same time
- Different classifications
  - structured
  - un-structured
  - hierarchically organized

Hybrid Architectures

- Edge-server systems
  - ISP
- Collaborative distributed systems
  - BitTorrent



Decentralized organizations

- Peer-to-peer systems
  - each process will act as a client and a server at the same time
- Different classifications
  - structured
  - un-structured
  - hierarchically organized