Parallelism

Master 1 International

Andrea G. B. Tettamanzi

Université de Nice Sophia Antipolis

Département Informatique

andrea.tettamanzi@unice.fr

Lecture 4 – Part a

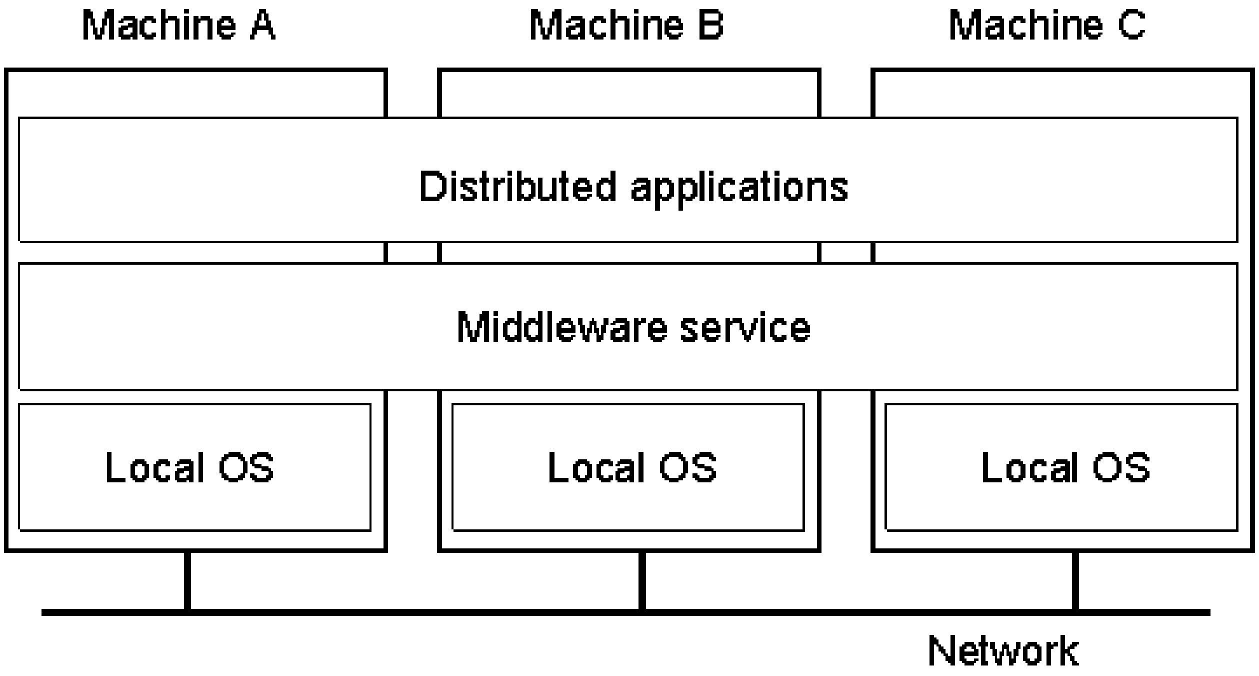
Distributed Architectures

Definition of a Distributed System (1)

A distributed system is:

A collection of independent computers that appears to its users as a single coherent system.

Definition of a Distributed System (2)



A distributed system organized as middleware.
Note that the middleware layer extends over multiple machines.

Goals

- Accessibility of resources
- Transparency
- Openness
- Scalability

Accessibility

- Sharing of resources
- Virtual communities, groupware
- E-banking
- Electronic Commerce
- Social Networks

Critical Aspects:

- Security
- Privacy

Different forms of Transparency

Transparency	Description
Access	Hide differences in data representation and how a resource is accessed
Location	Hide where a resource is located
Migration	Hide that a resource may move to another location
Relocation	Hide that a resource may be moved to another location while in use
Replication	Hide that a resource may be shared by several competitive users
Concurrency	Hide that a resource may be shared by several competitive users
Failure	Hide the failure and recovery of a resource
Persistence	Hide whether a (software) resource is in memory or on disk

Openness

- Compliance to standard rules: syntax & semantics
- Protocols
- Interfaces – IDL
- Semantics: ontologies
- Interoperability
- Portability
- Separation of policies from mechanisms: parameterization

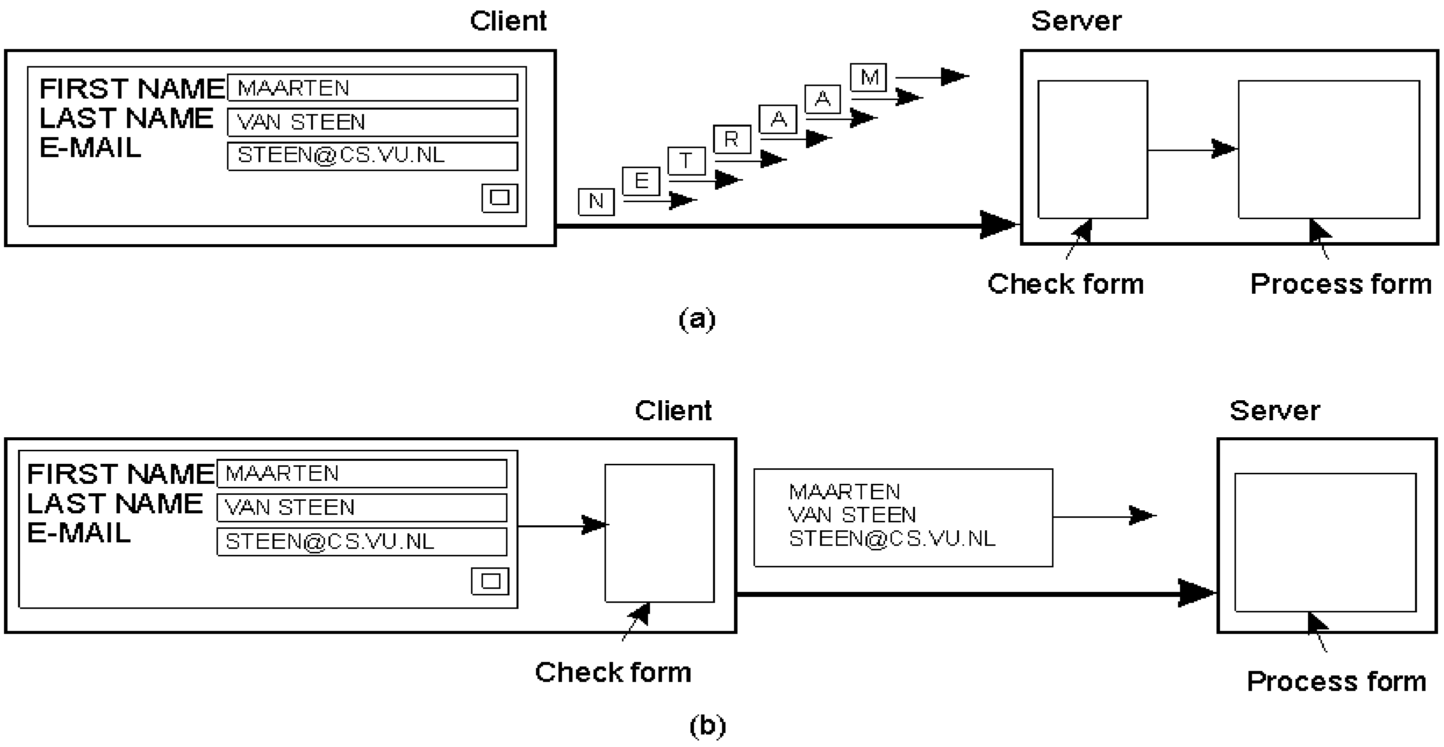
Dimensions of scalability

- With respect to size (users, resources)
- With respect to geographic deployment
- With respect to administration/management

Examples of Scalability Limitations

Concept	**Example**
Centralized services	A single server for all users
Centralized data	A single on-line telephone book
Centralized algorithms	Doing routing based on complete information

Scaling Techniques (1)



(a)

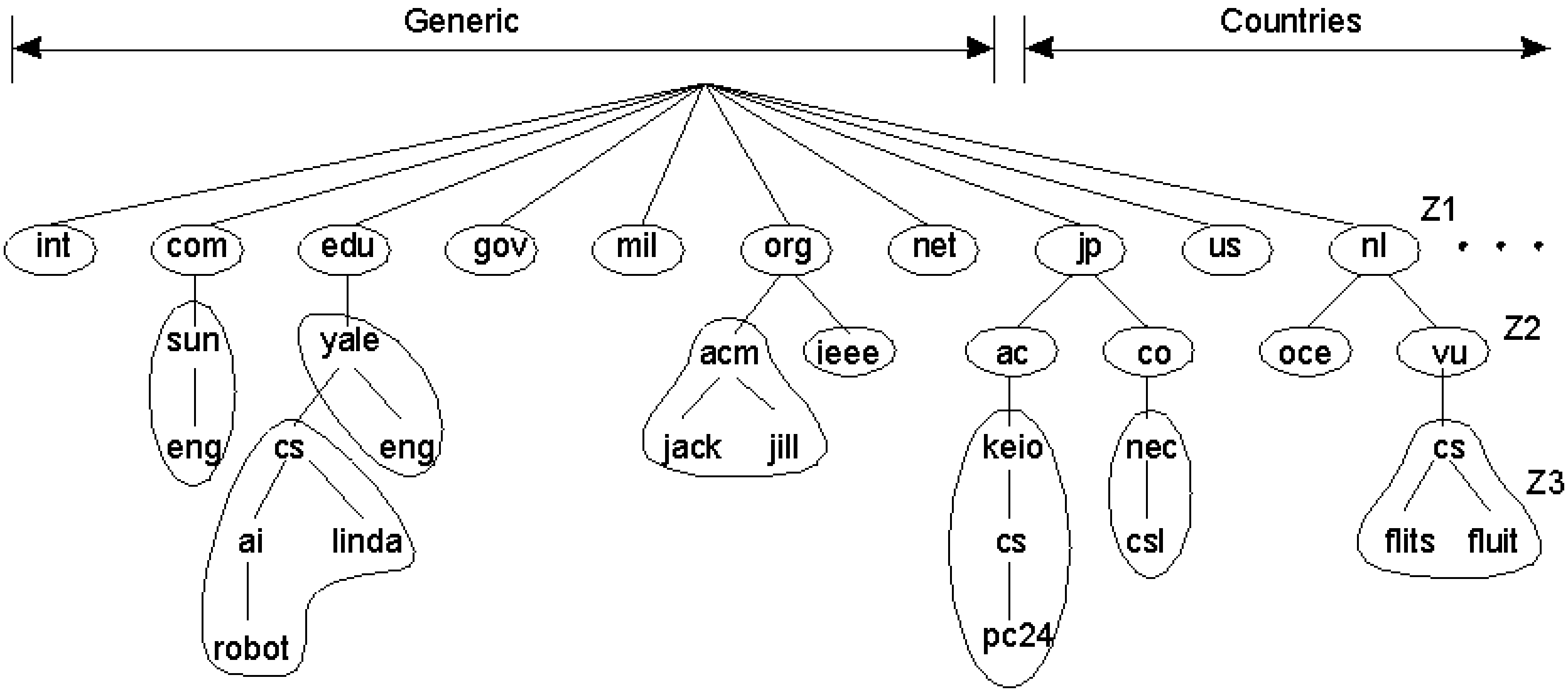
(b)

The difference between letting:

a) a server or

b) a client check forms as they are being filled

Scaling Techniques (2)



An example of dividing the DNS name space into zones.

Pitfalls – False Assumptions

1. The network is reliable
2. The network is secure
3. The network is homogeneous
4. The topology does not change
5. Latency is negligible
6. Bandwith is unlimited
7. The cost of transport is zero
8. There is one administrator

Types of Distributed Systems

- Computing Systems
  - Cluster, Grid
- Information System
  - Transnational systems, Integration of applications
- Pervasive Systems
  - Home automation, Healthcare, Sensor networks

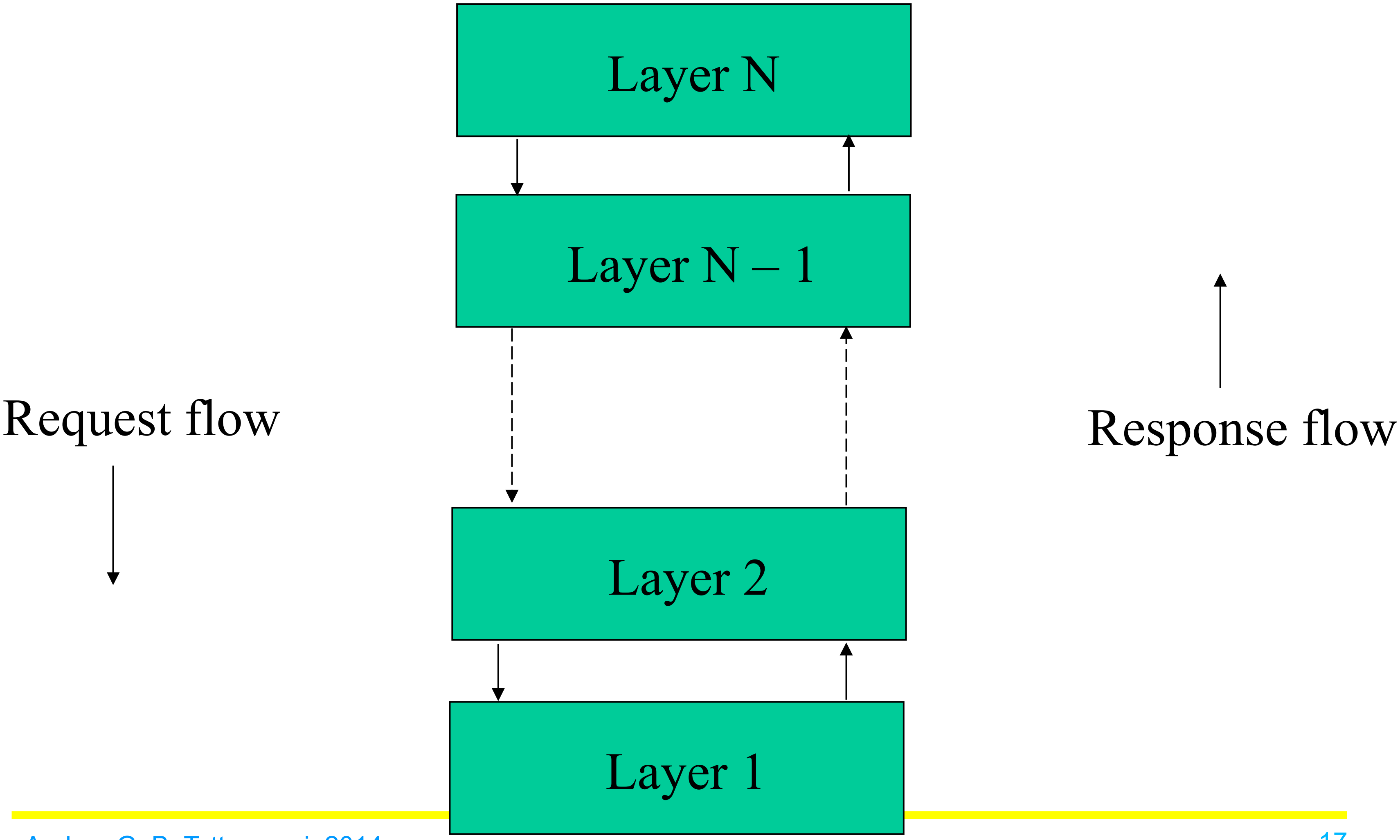
Components and Connectors

- Architectual style expressed in terms of
  - Components
    - A modular unit
    - With well-defined required and supplied interfaces
  - Connectors
    - Any mechanism that mediates
      - Communication
      - Coordination
      - Cooperation

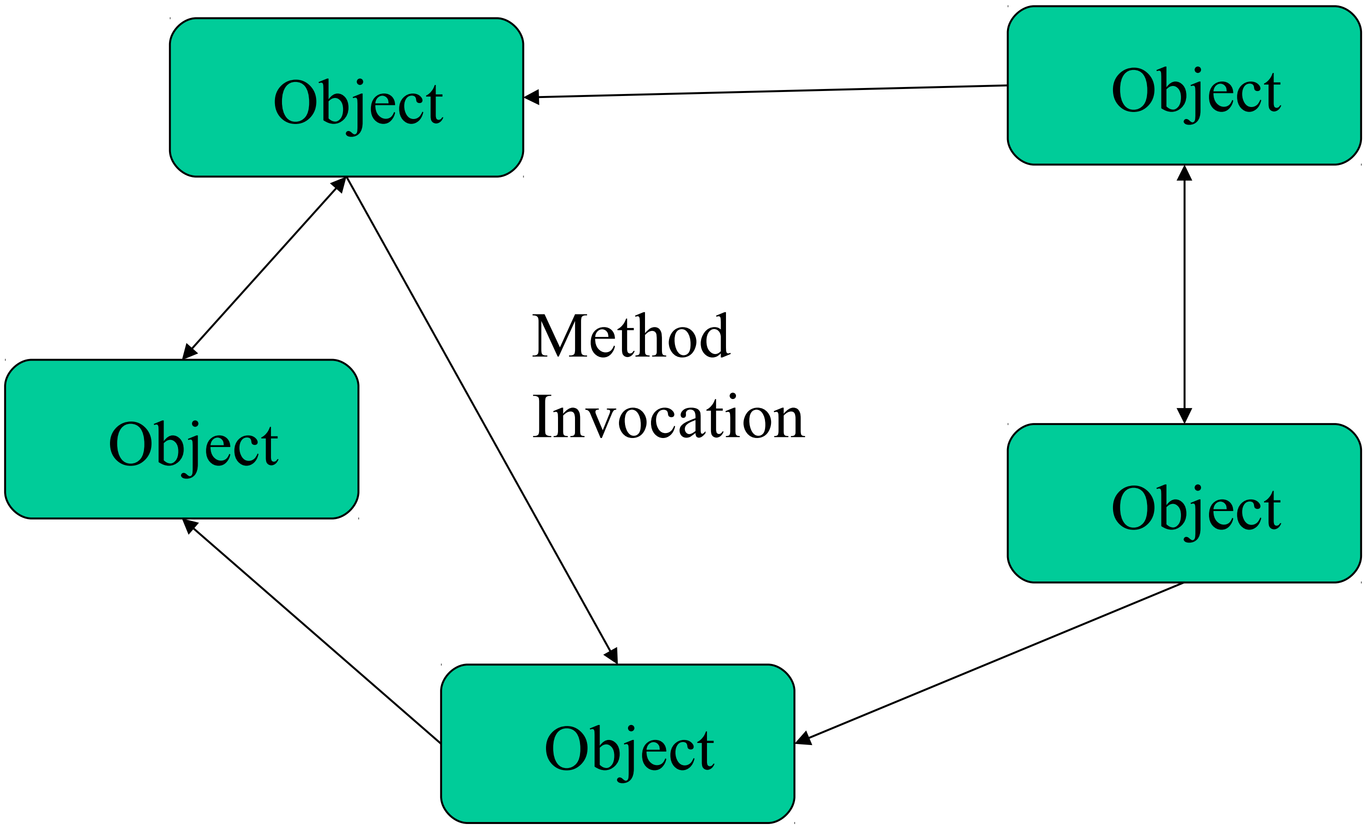
Architectural Styles

- Layered Architecture
- Object-Oriented Architectures
- Data-Centric Architectures
- Event-Based Architectures

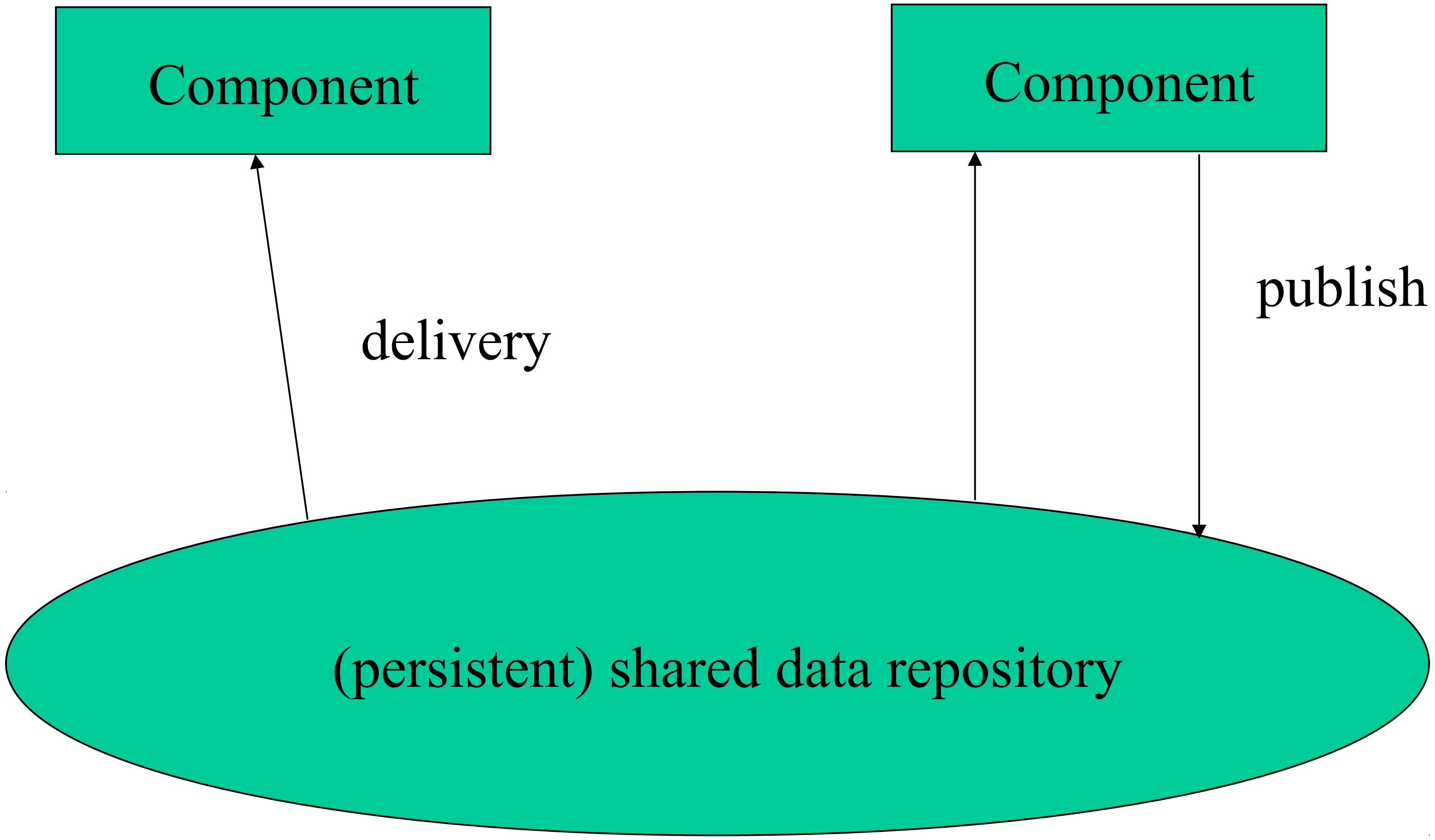
Layered Architectures



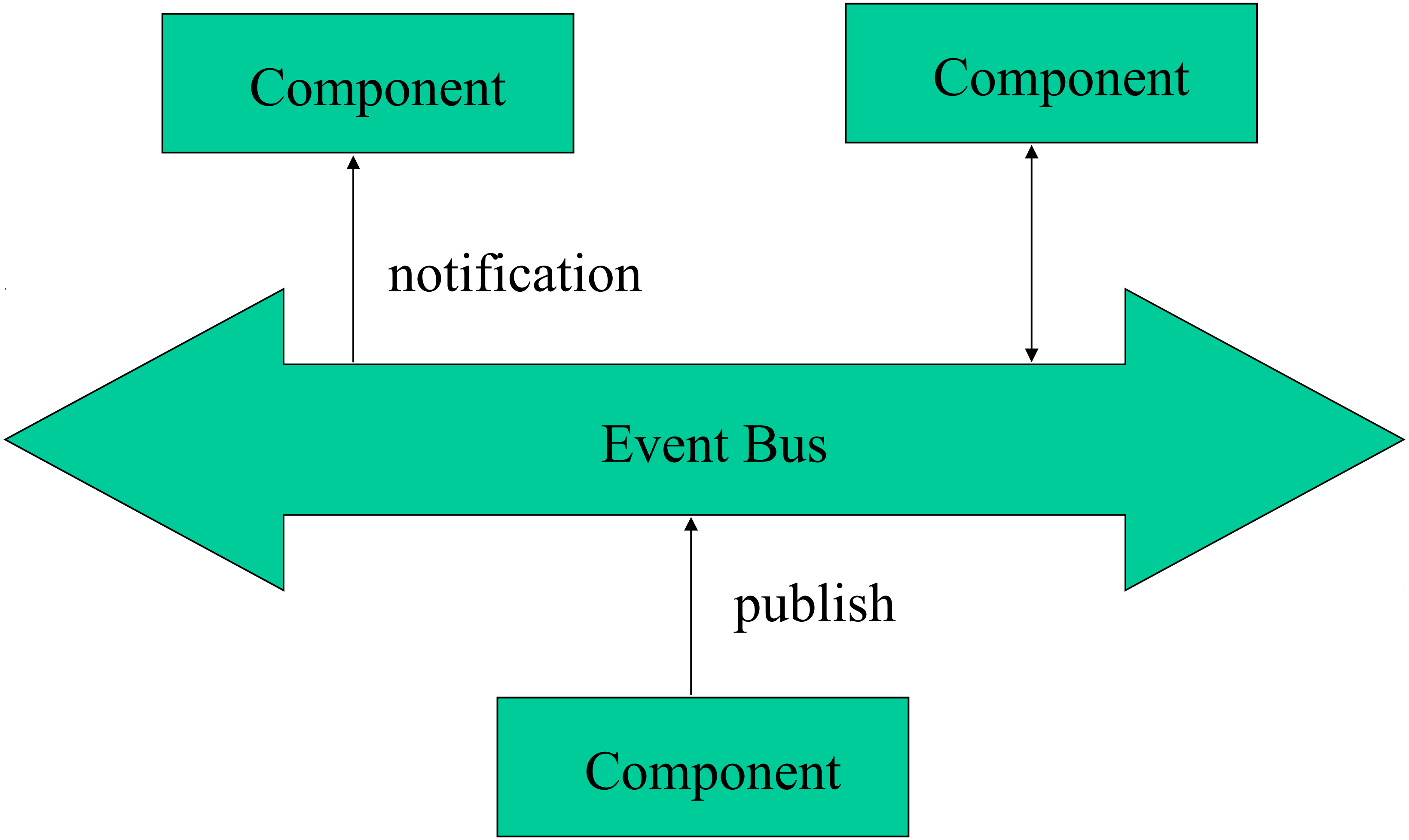
Object-Oriented Architectures



Data-Centric Architectures



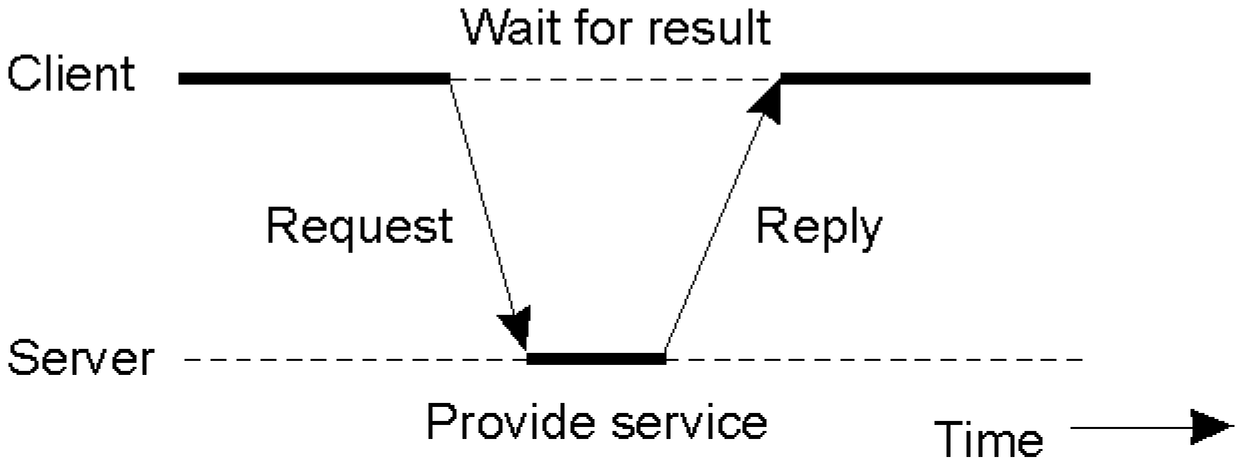
Event-Based Architectures



System Architectures

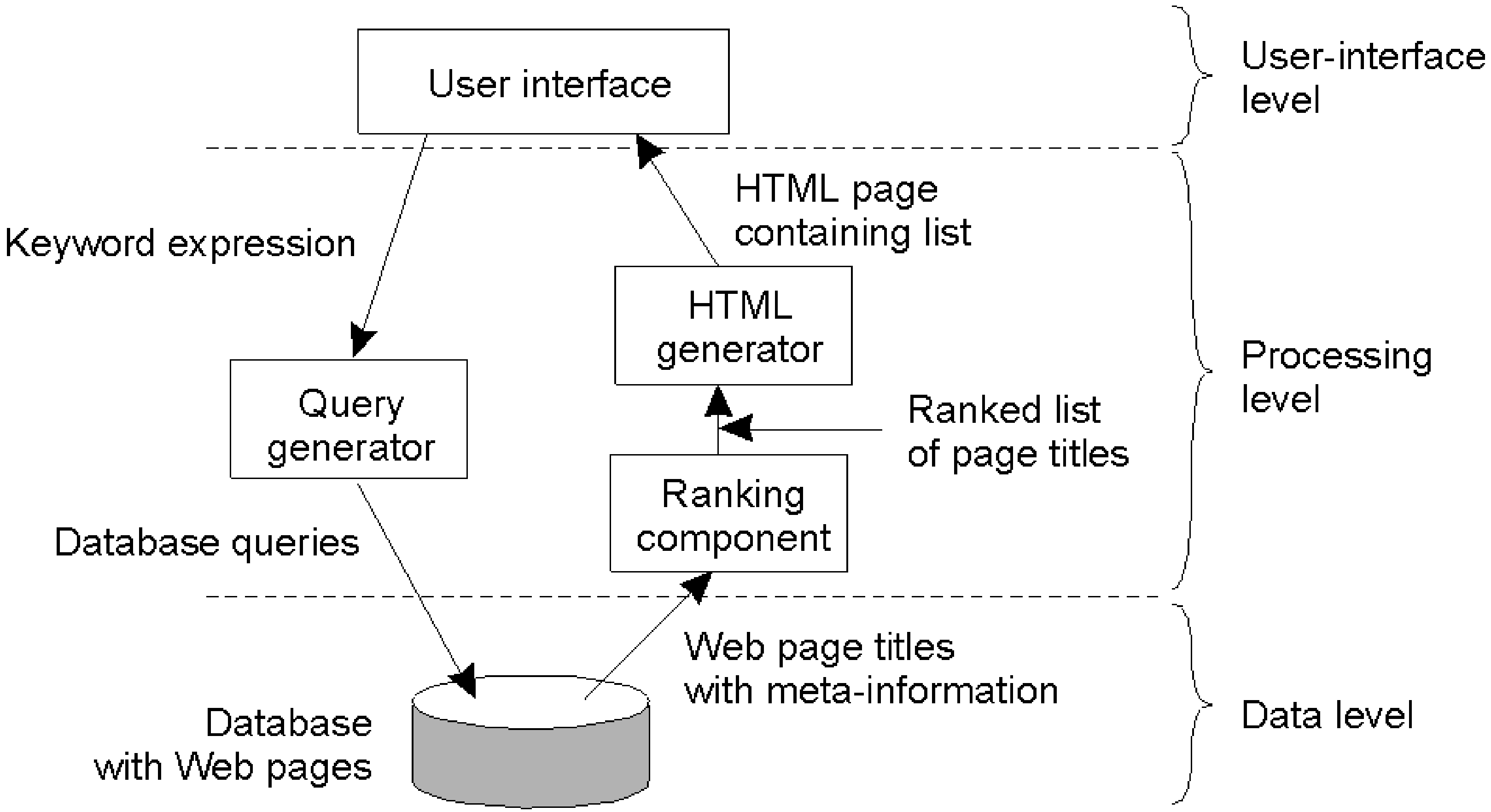
- Centralized Architectures
- Decentralized Architectures
  - Structured Peer-to-Peer Architectures
  - Unstructured Peer-to-Peer Architectures
- Hybrid Architectures

Clients and Servers



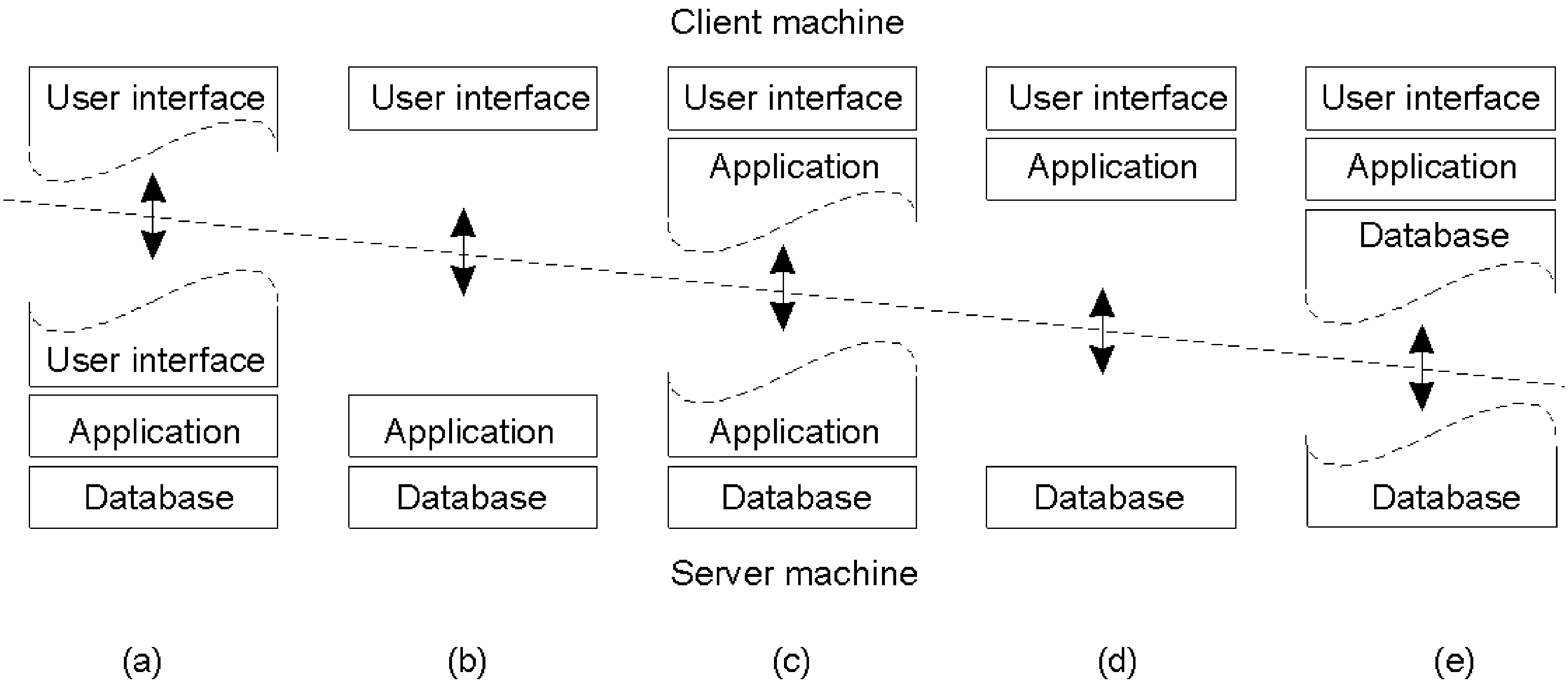
General interaction between a client and a server.

Processing Level



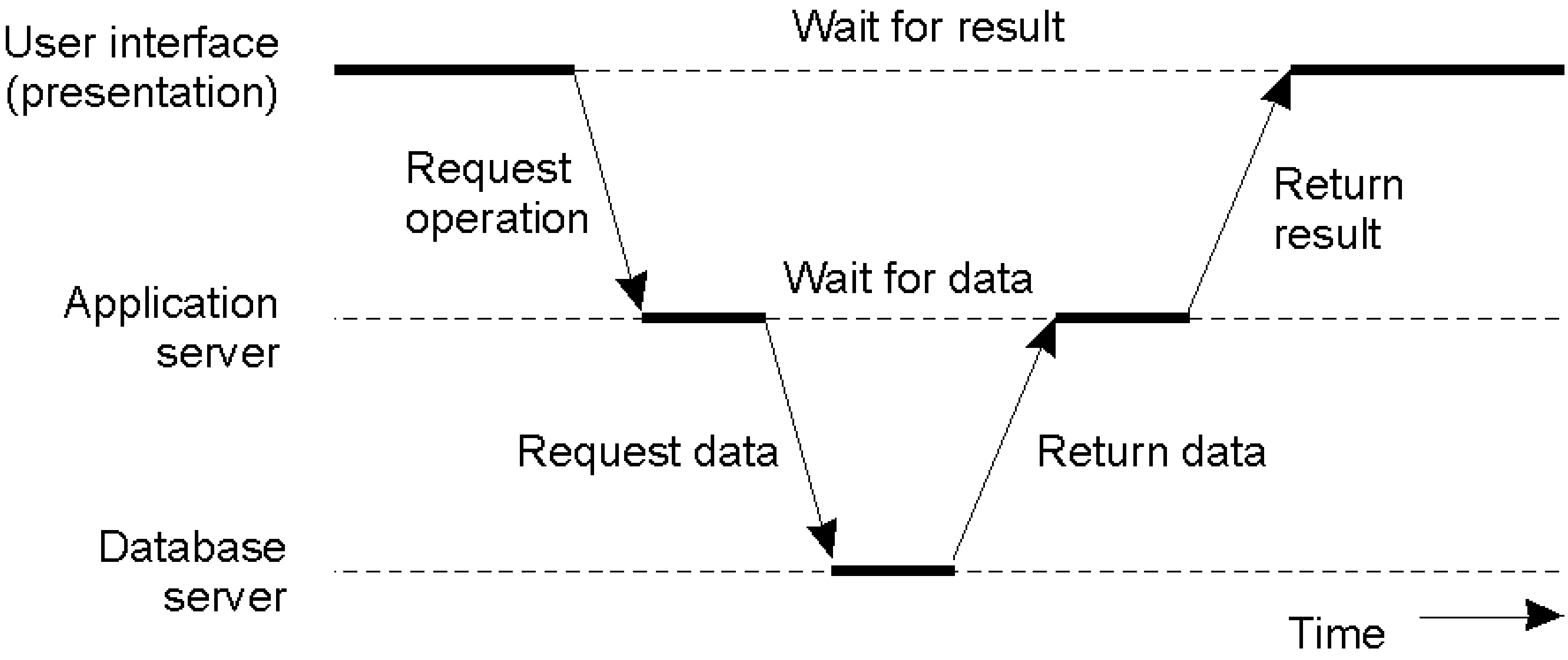
The general organization of an Internet search engine into three different layers

Multitiered Architectures (1)



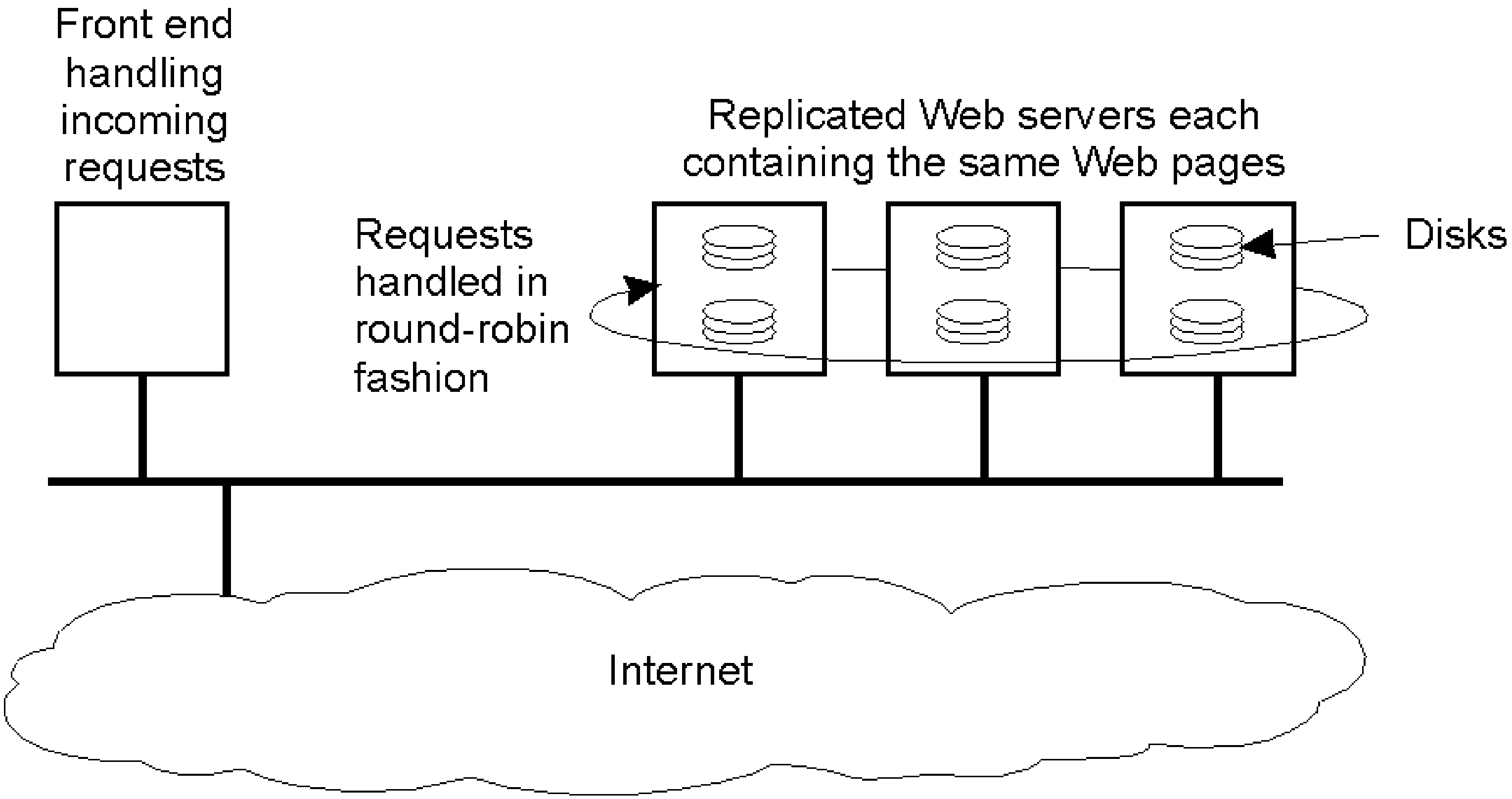
Alternative client-server organizations (a) – (e).

Multitiered Architectures (2)



An example of a server acting as a client.

Modern Architectures


Front end handling incoming requests
Requests handled in round-robin fashion
Replicated Web servers each containing the same Web pages
Disks
Internet

An example of horizontal distribution of a Web service.

Peer-to-Peer Architectures

Horizontal Distribution

Symmetric Interaction among Processes (“Servents”)

Overlay Networks (structured/non-structured)

Applications

Communication and collaboration (IM)

Distributed Computing (...@home)

Internet Service Support

Databases

Content Distribution

Overlay Networks

- Pure Decentralized (all nodes are equal)
- Partially Centralized (nodes + supernodes)
- Hybrid Decentralized (central server + nodes)

- Unstructured (content unrelated to topology)
- Structured (content related to topology)

Structured Architectures

Distributed hash table (DHT):

Data mapped into keys $k \in H$

Nodes choose random identifier $i \in H$

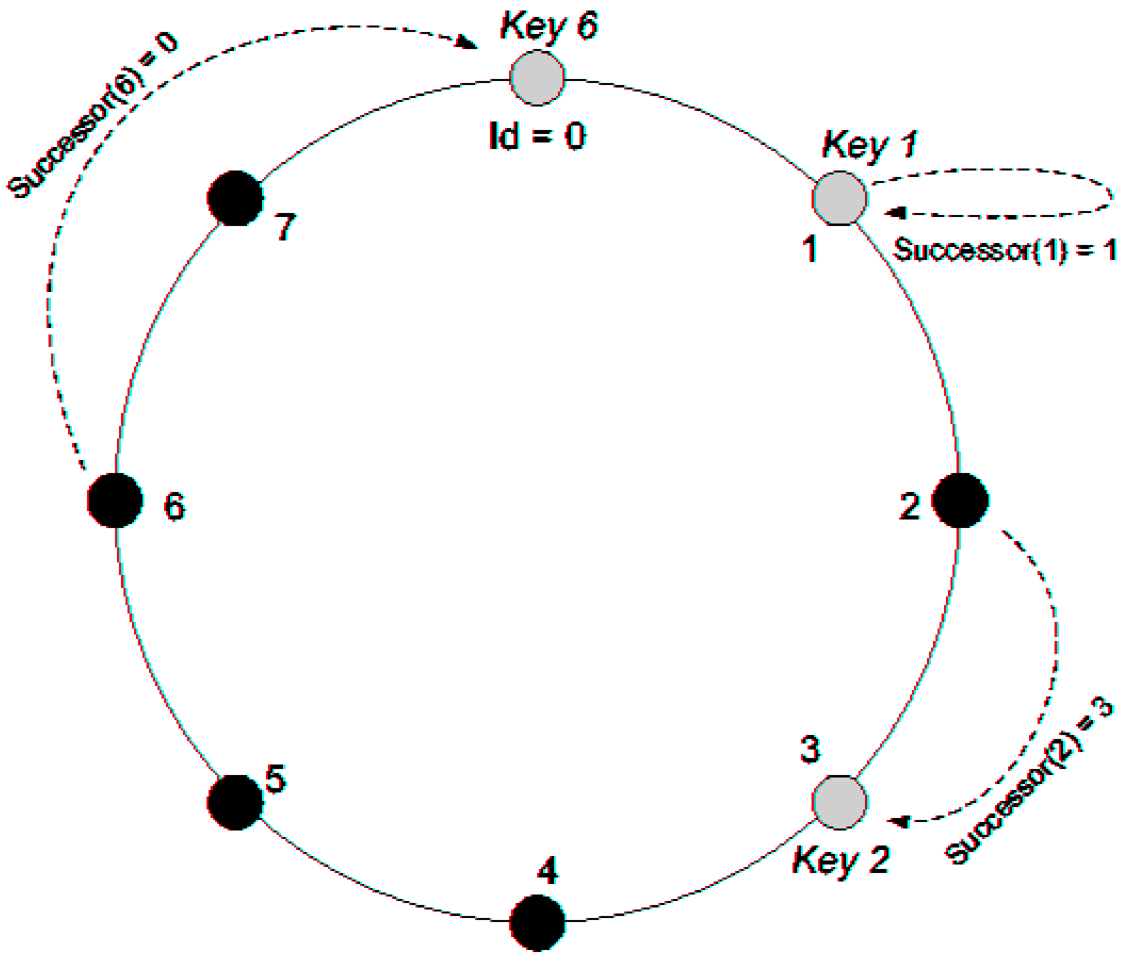
Map $f : H \to H$, which assigns k to i.

Membership Management:

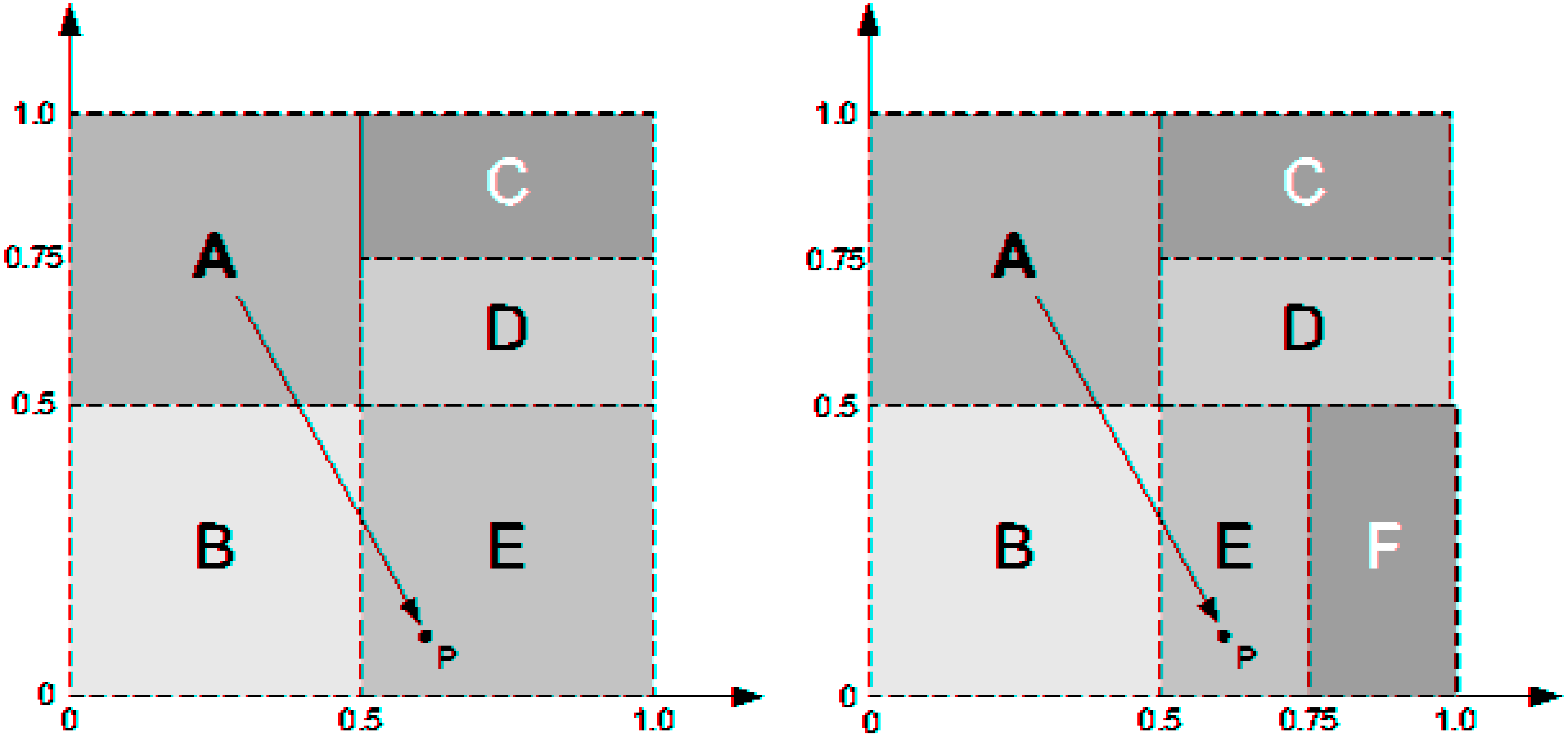
Entry of a new node

Exit of a node

Chord


Key 6
Id = 0
Successor(6) = 0
Key 1
Successor(1) = 1
1
7
6
2
3
Successor(2) = 3
5
4
Key 2

Content-Addressable Network



E's neighbour set: {B,D}

(a)

E's neighbour set: {B,D,F}

F's neighbour set: {D,E}

(b)

Unstructured Architectures

Random algorithms to construct overlay network

Search: flood

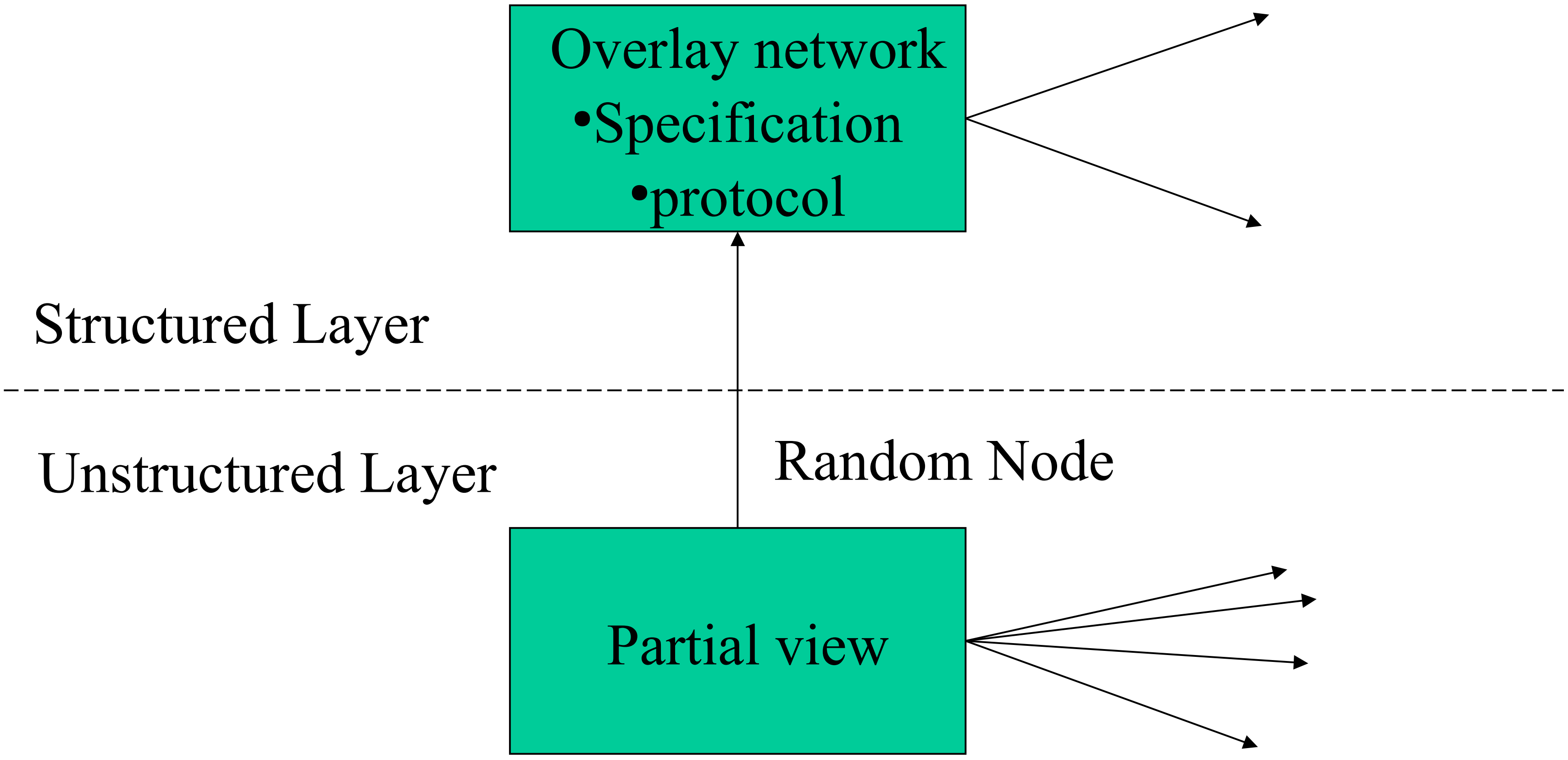
Every node keeps a list of “neighbors”: partial view

Updating a partial view

Two complementary approaches:

1) Exchange half views between two nodes
2) Remove “old” nodes from every list

Topology Management



Overlay network

- Specification
- protocol

Structured Layer

Unstructured Layer

Random Node

Partial view

Overlay Network Specification

Ranking function

e.g.: distance

e.g.: semantic similarity

Partially Centralized Networks

Problem: searching data in unstructured networks

Solution: Superpeers

Hierarchical Organization:

Superpeer networks

Subnetworks of peers constructed around superpeers

Dynamic election of superpeers
(cf. Synchronization)

Thank you for your attention!