CSci 5105

Introduction to Distributed Systems

Architectural Styles

Today

- Architectural Styles
- Chapter 2 TVS
- Homework #1 is on the website

Last Time

- Spoke about transparency

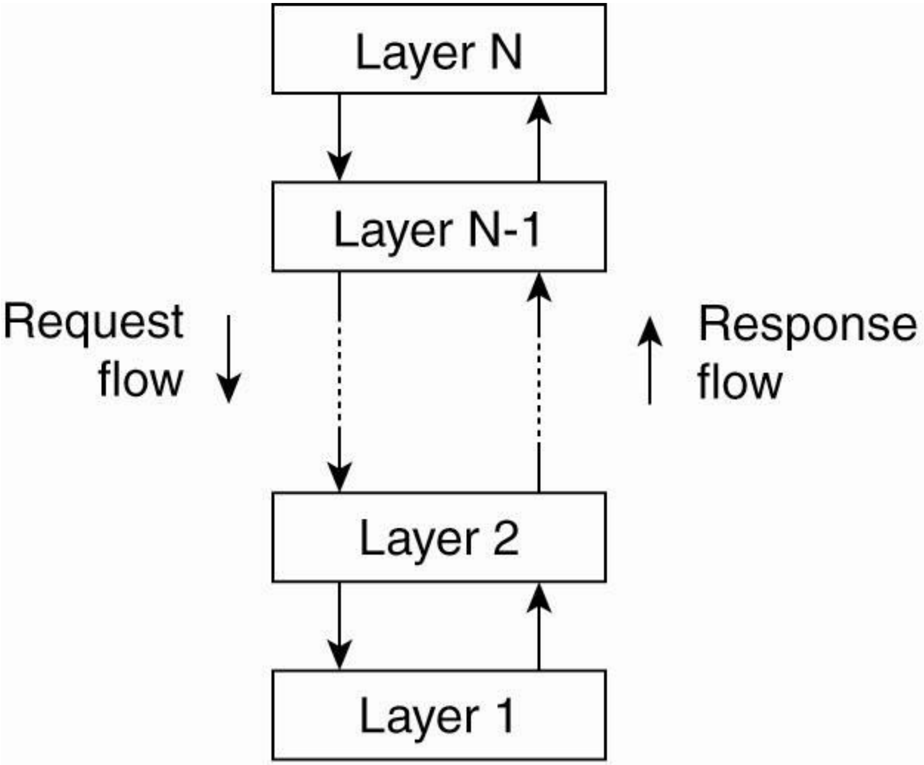
Architectural Styles

- How components are organized?
- How components interact?

- Layered architectures
- Object-based architectures
- Data-centered architectures
- Event-based architectures

Architectural Style 101

OSI Model



Layer N

Layer N-1

Request flow

Response flow

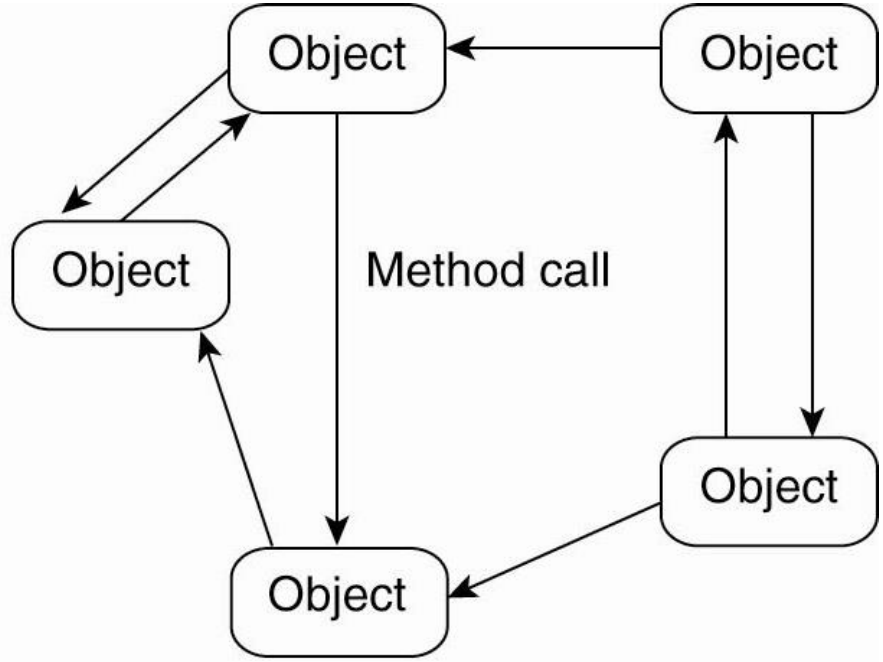
Layer 2

Layer 1

+/- ?

Object-Based (my fave)

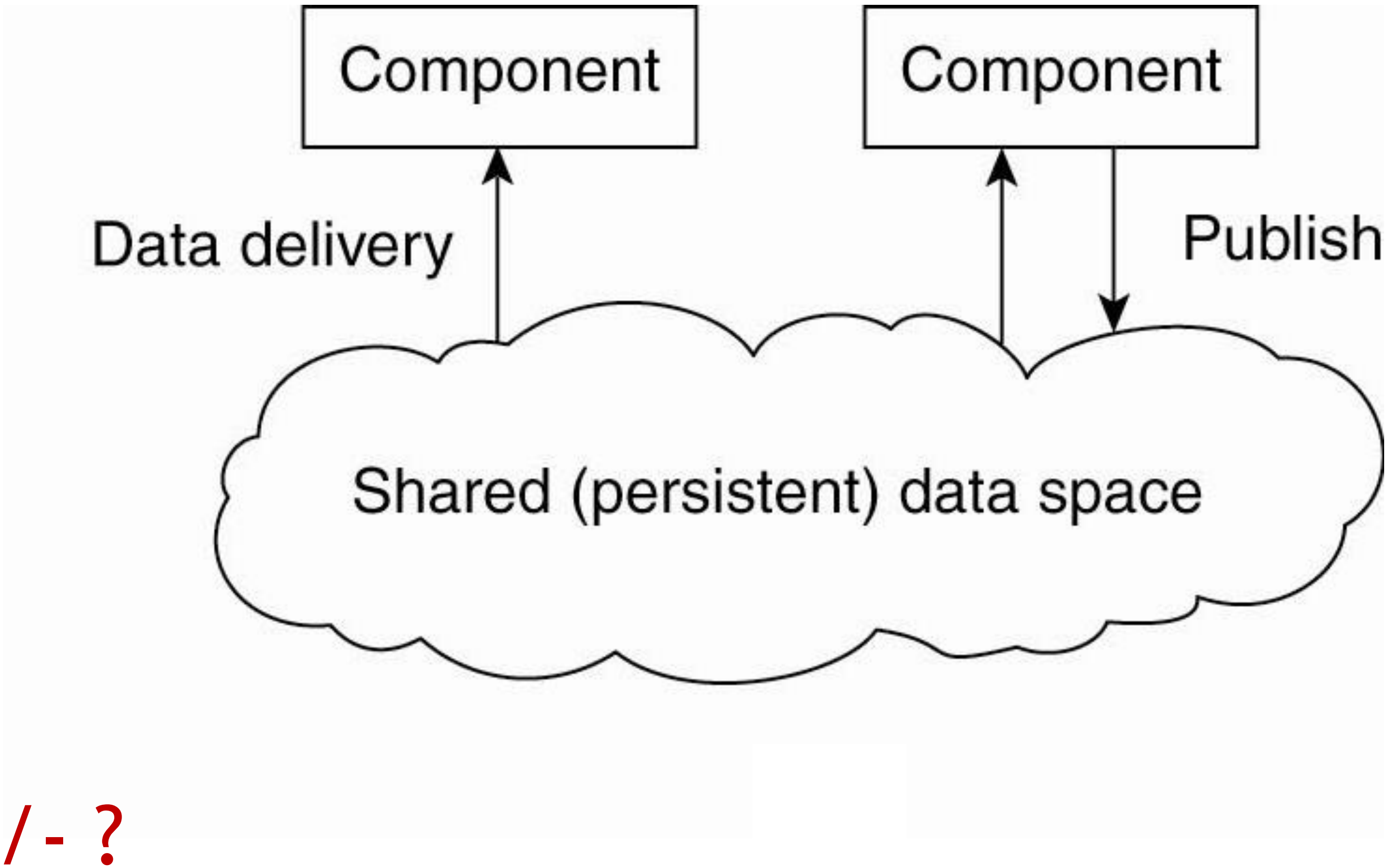
RPC or RMI



+/- ?

Data-Based

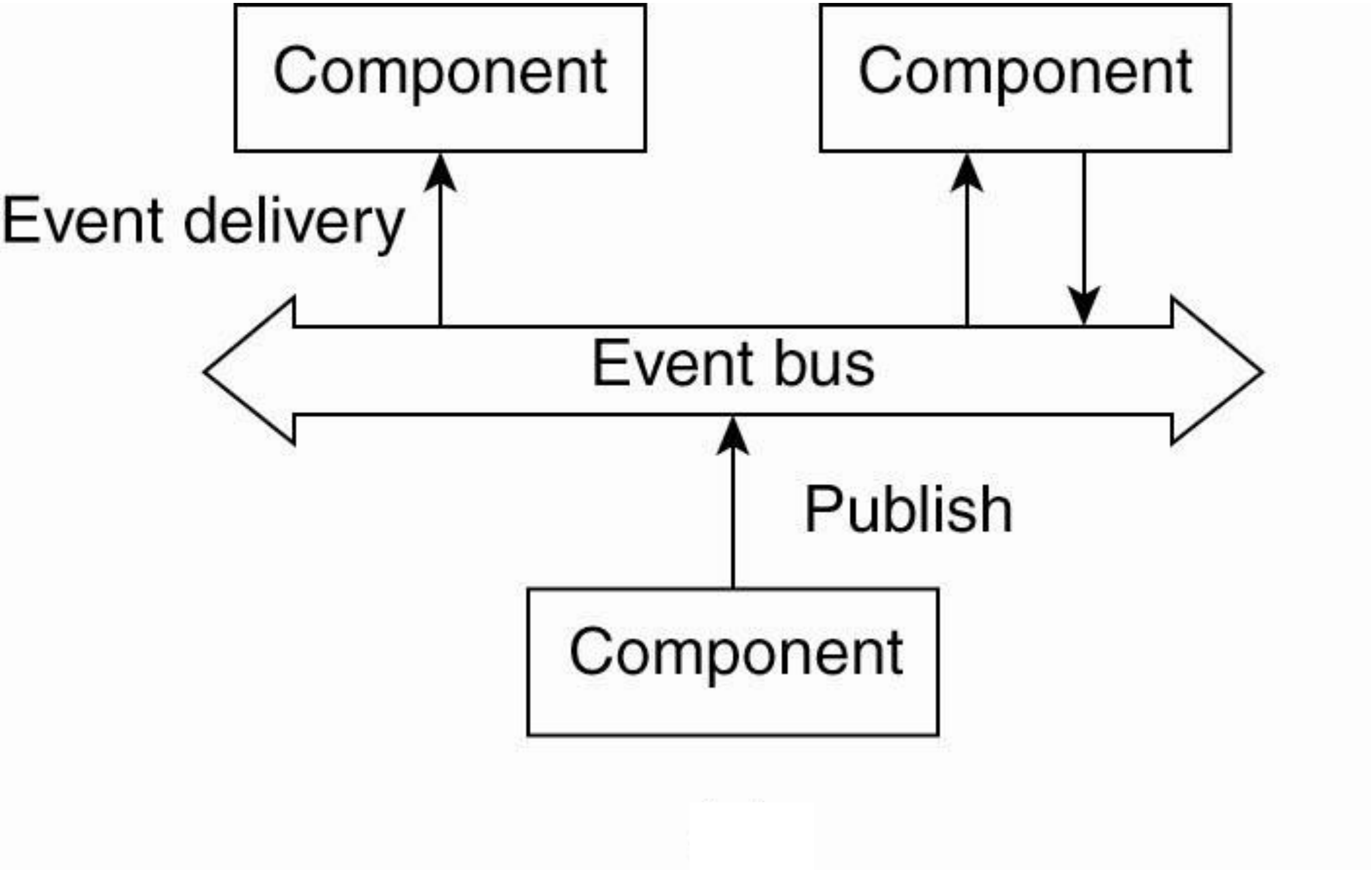
Publish-Subscribe



+/- ?

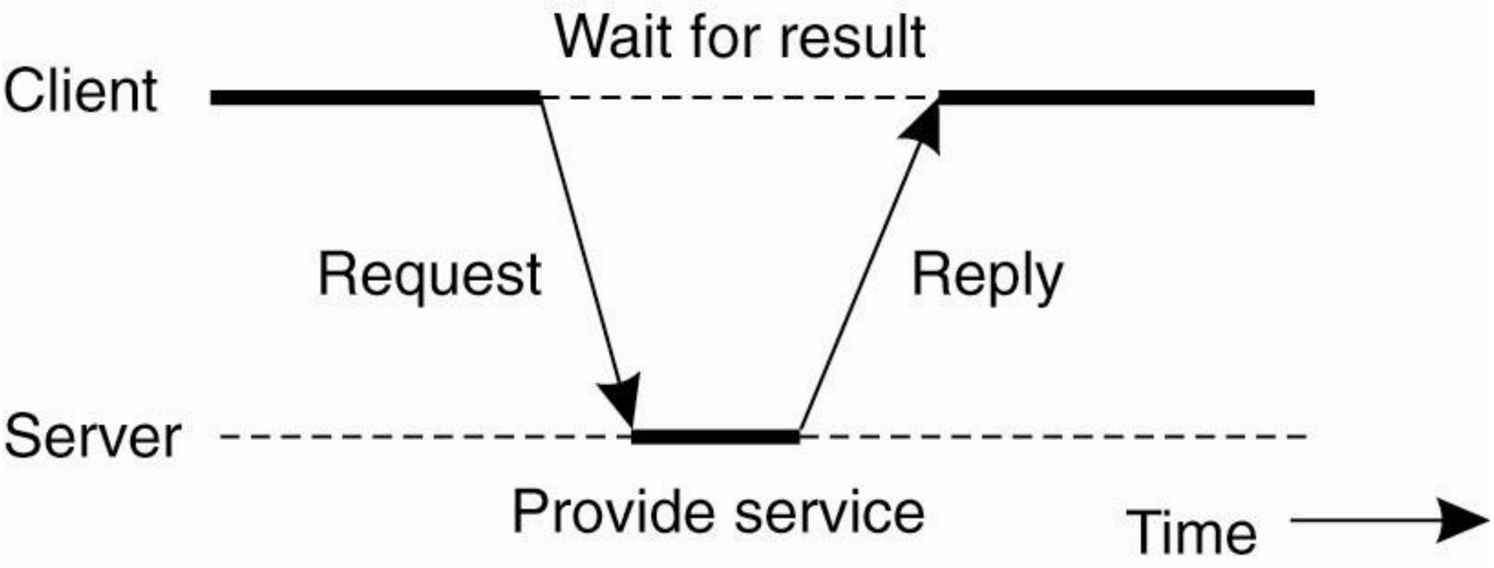
Event-Based

?


Component
Component
Event delivery
Event bus
Publish
Component

+/- ?

Basic Architecture: Client-Server

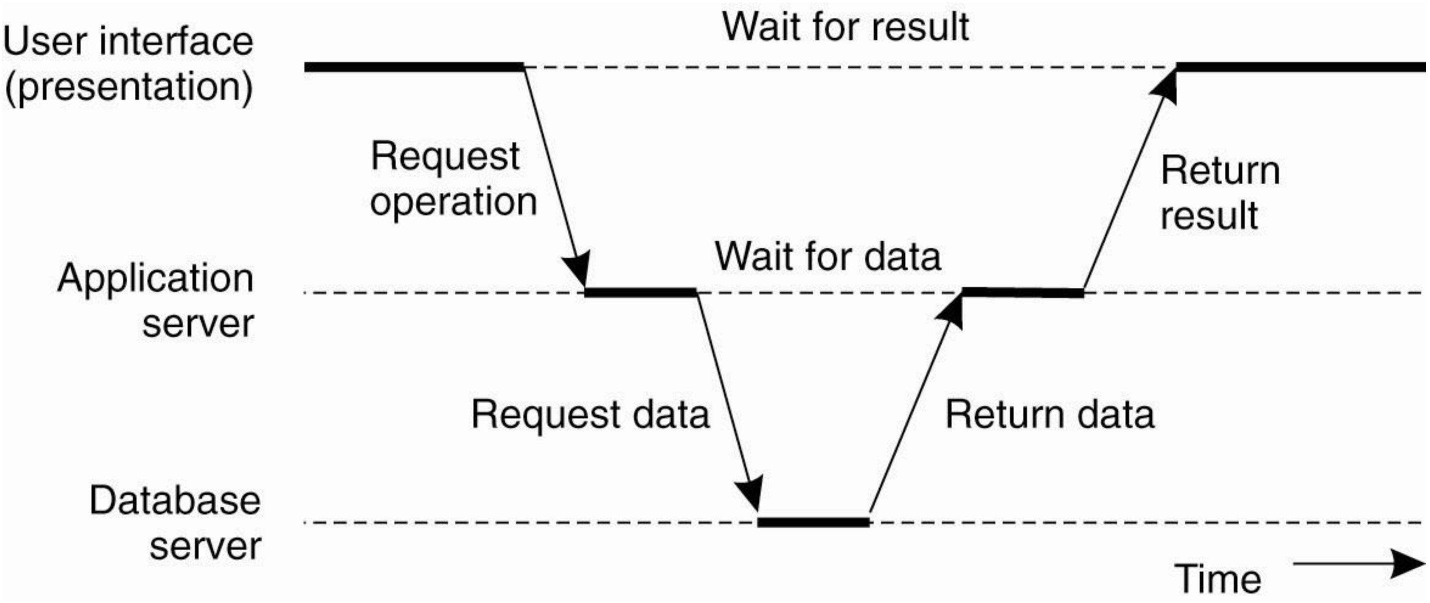


Always-up server
Client is transient

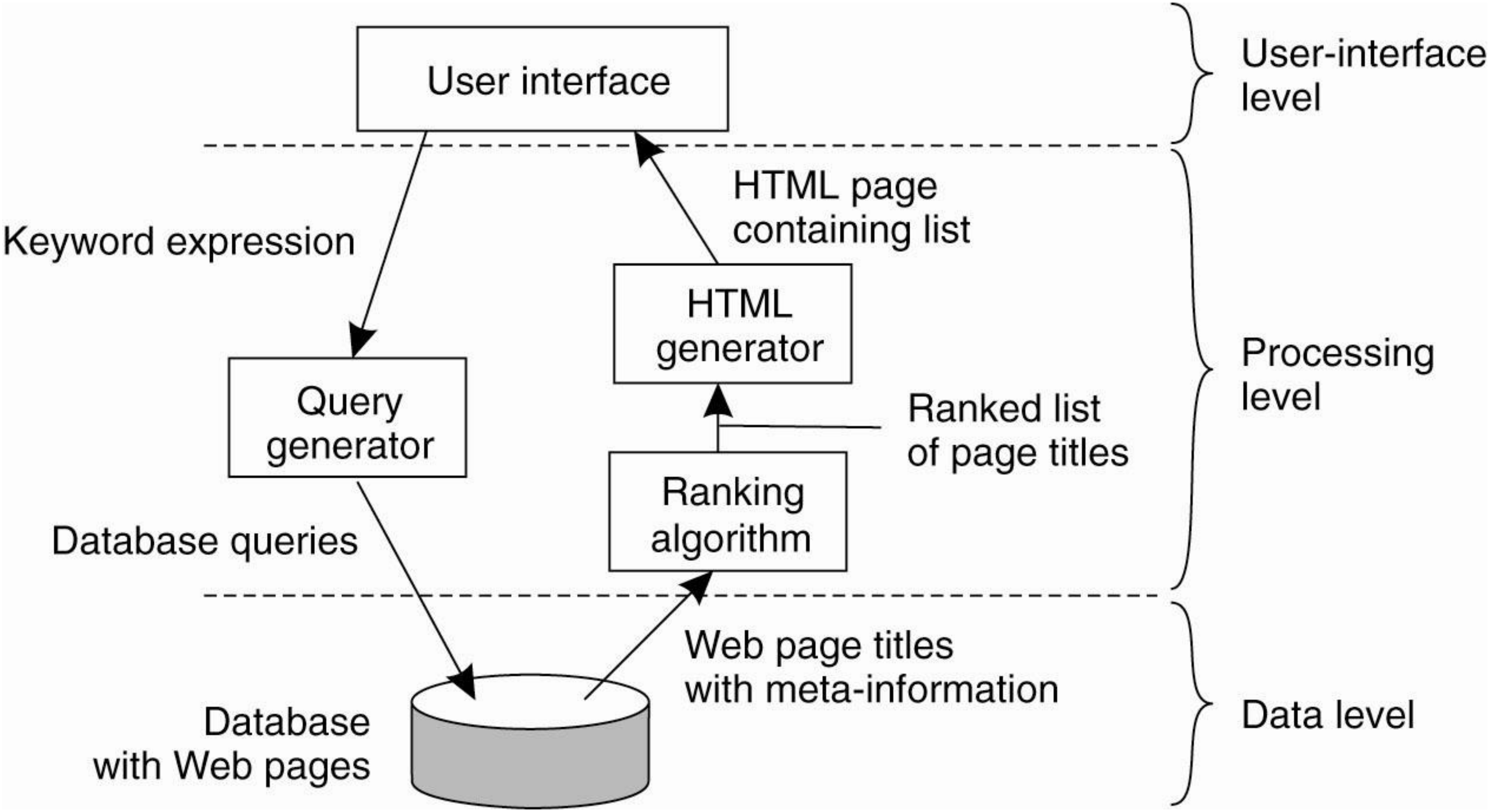
+/- ?

Multi-tier Client-Server

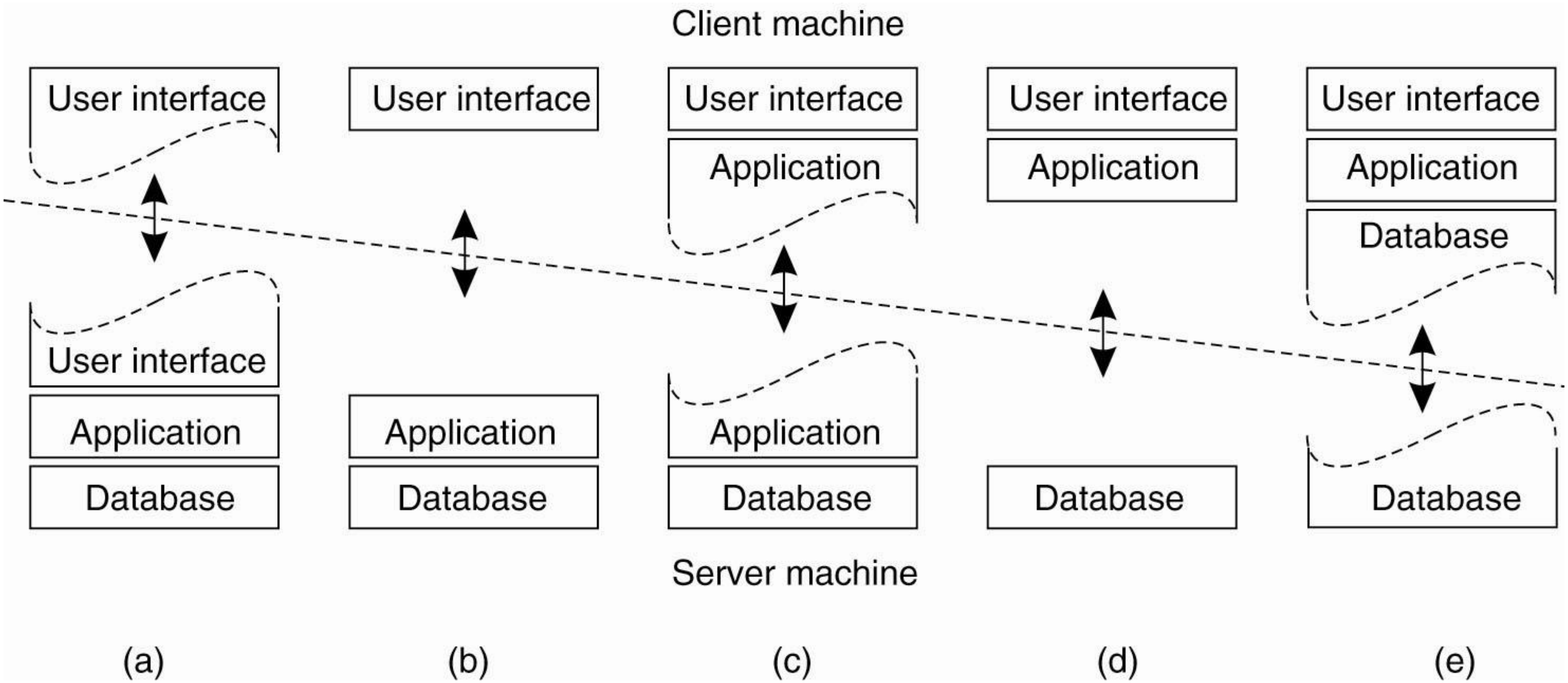
- The user-interface level
- The processing level
- The data level



Example: Web Search

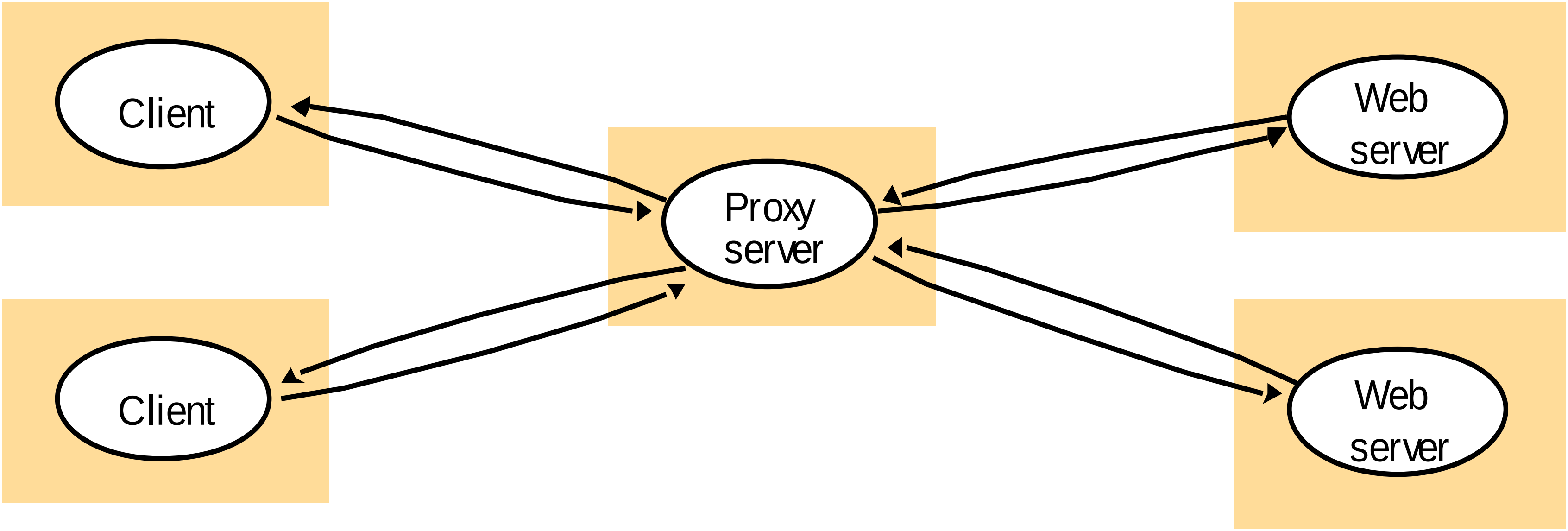


Where are the tiers? Rich array of options.



Very thin => Very thick client

Proxy Architecture



Peer-to-Peer Architecture

- No “always-on” server
- Role of clients and server are blurred

+/- ?

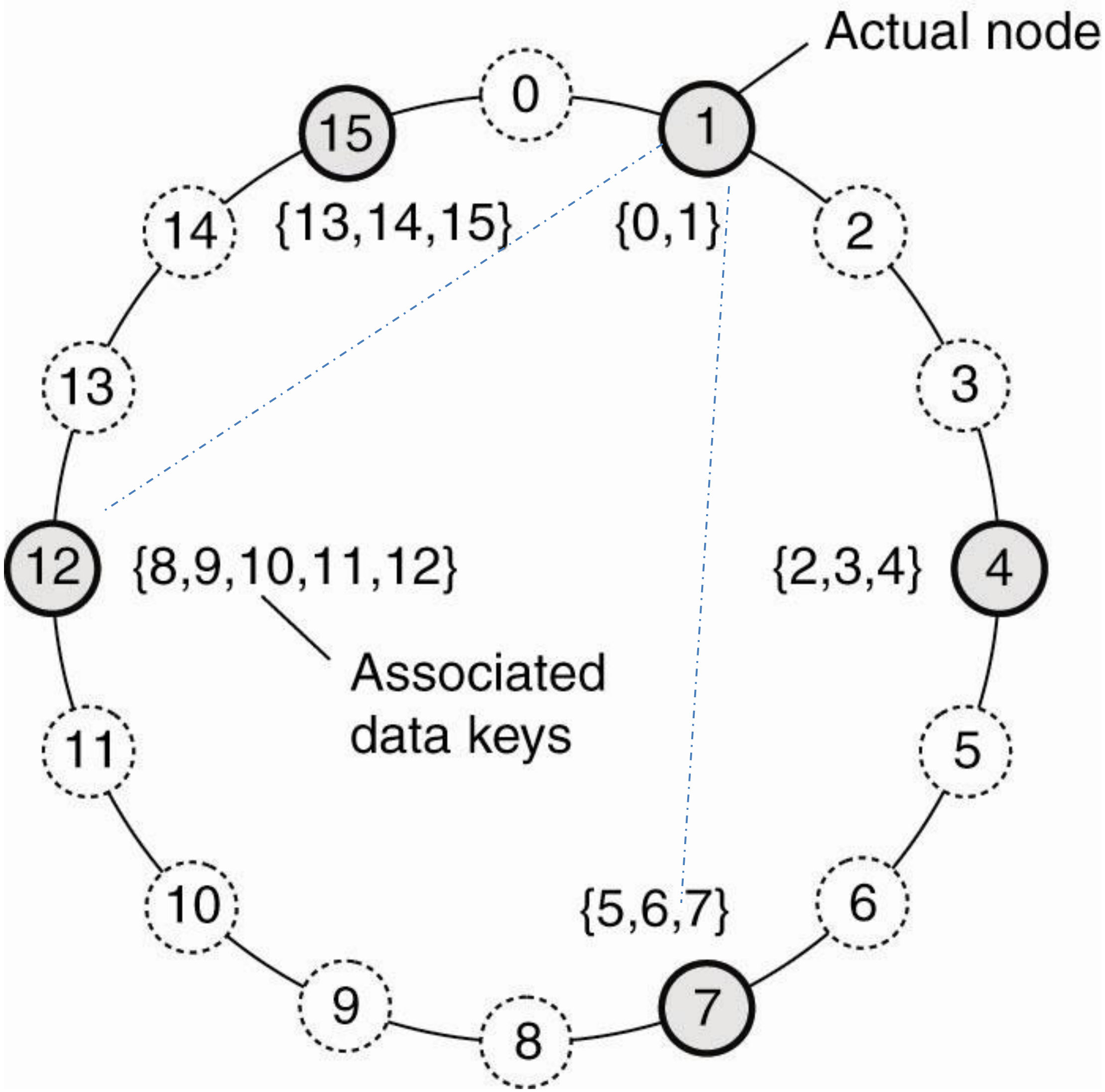
Structured Peer-to-Peer Architectures

Chord (MIT):

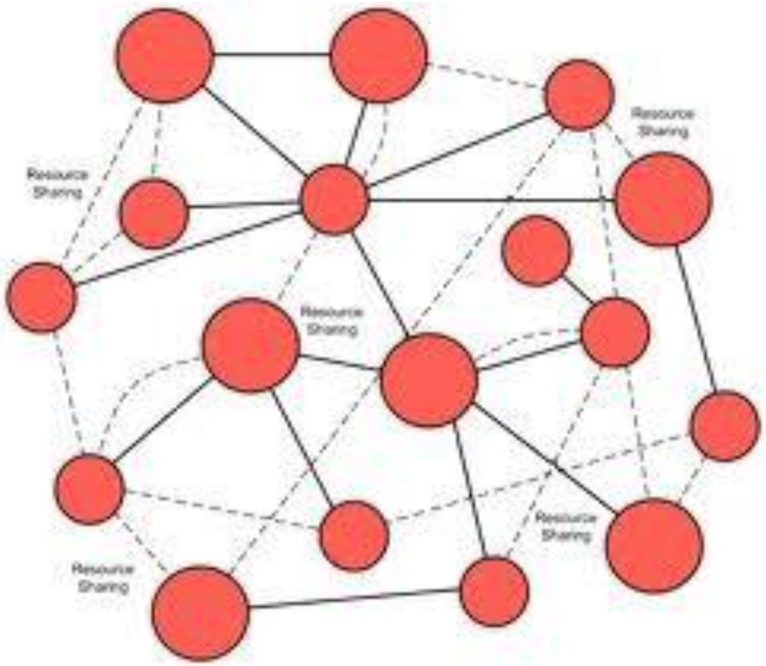
Hash (node),
Hash (item) -> ID

Log_2 N operations

Later: many others

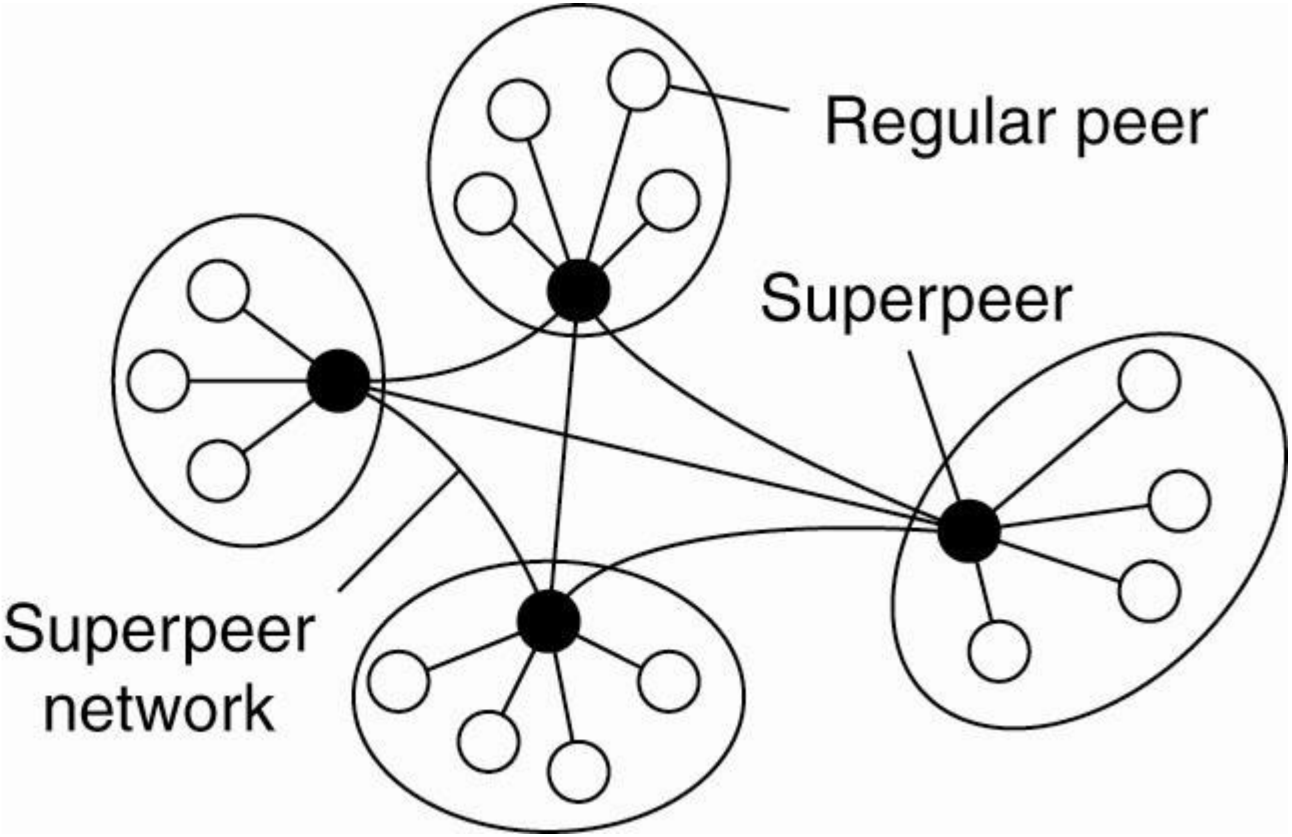


Unstructured Peer-to-Peer Architectures



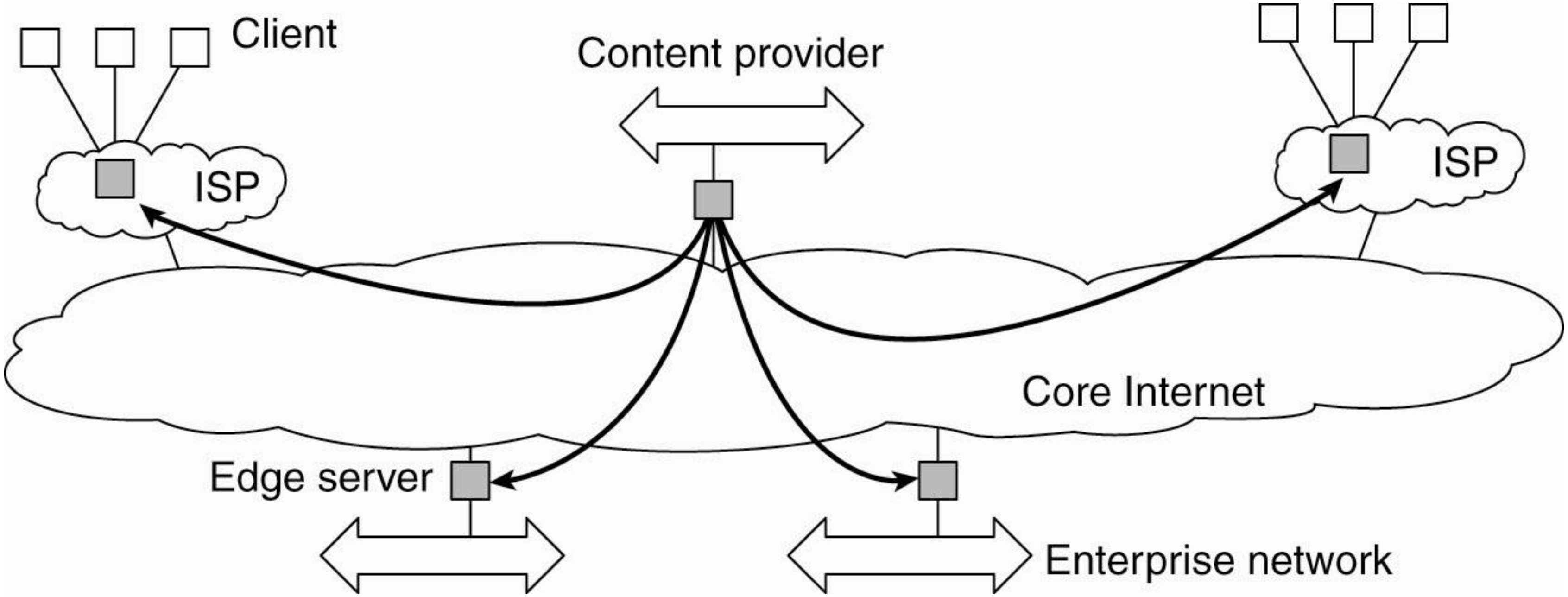
- Random graph
- Lookup: random walk or flooding

Hybrids: Adding Structure



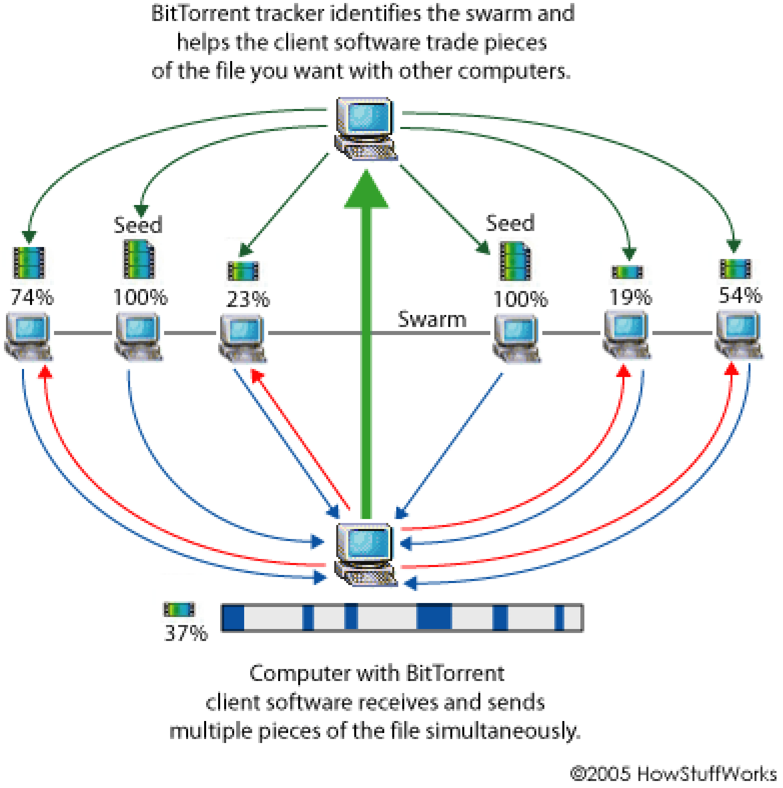
Super peer indexes content of its contained peers

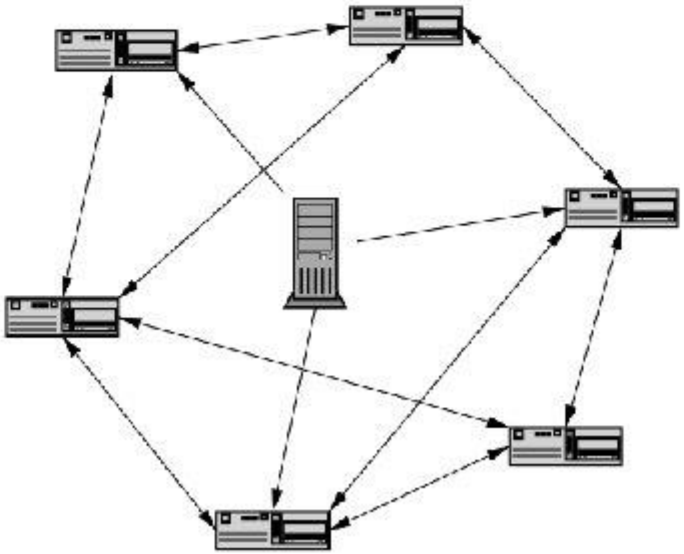
Edge-Server Systems: CDNs



Locality-Aware

Controlled Peer-to-Peer


BitTorrent tracker identifies the swarm and helps the client software trade pieces of the file you want with other computers.
Seed
Seed
74%
100%
23%
100%
19%
54%
Swarm
37%
Computer with BitTorrent client software receives and sends multiple pieces of the file simultaneously.
©2005 HowStuffWorks



BitTorrent: rate of download = rate of upload

Prevent freeloaders

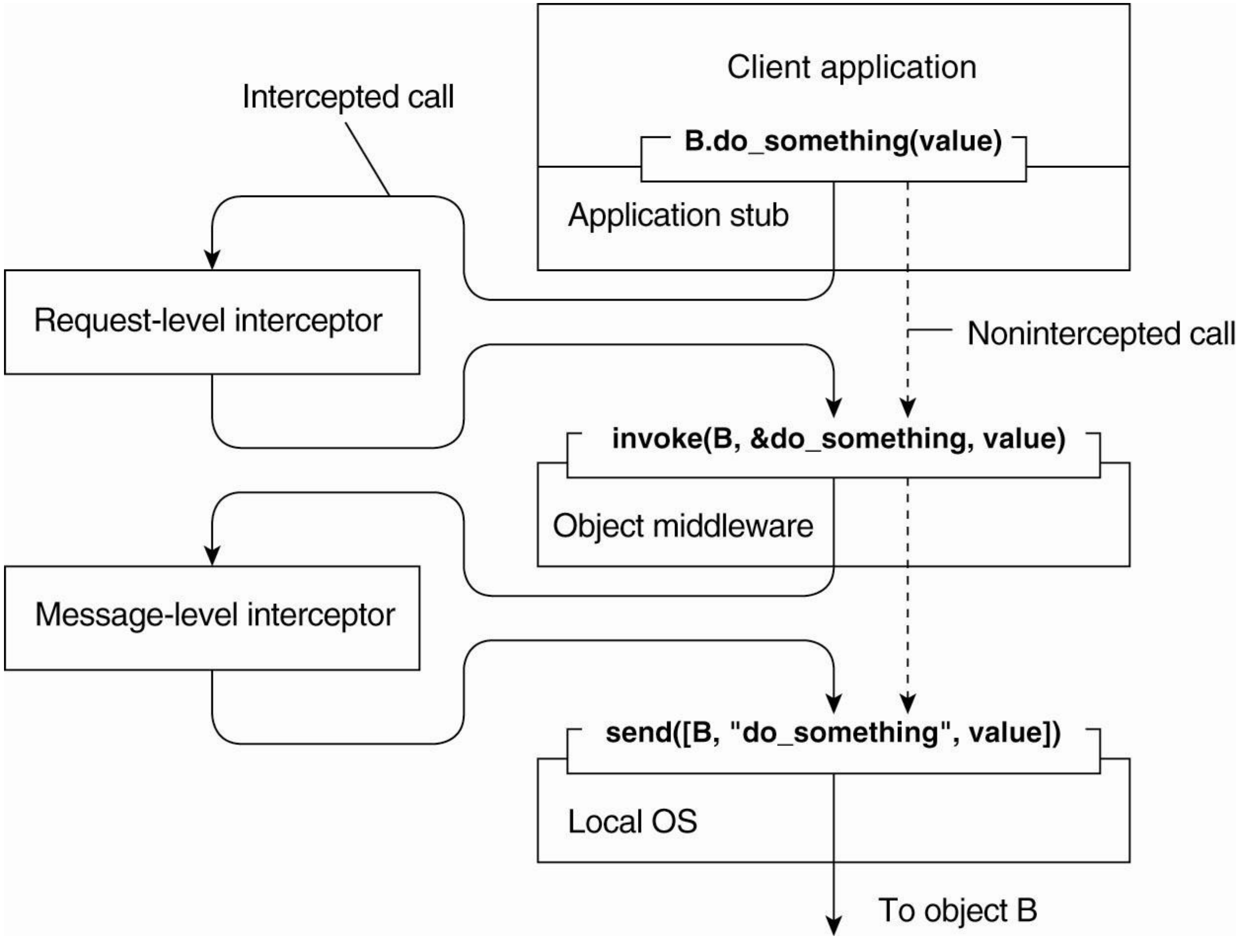
Optional Material

- Time permitting, we will discuss these

Adaptation Styles

- Distributed applications must be flexible and adaptive
  - Add functionality (even while up!)
  - React to changes in the environment

Interceptors



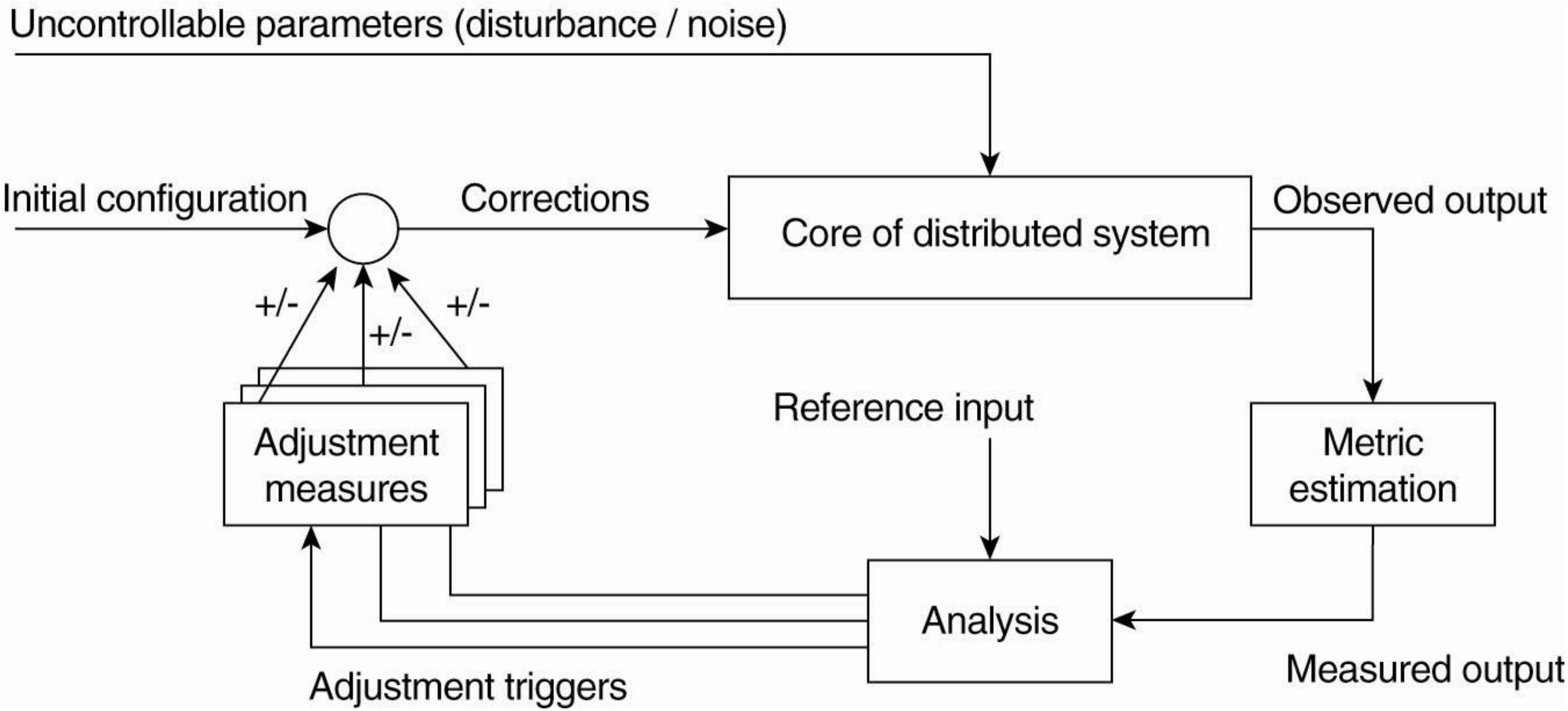
RPC is a type of interceptor: fairly static

Dynamic Component Replacement

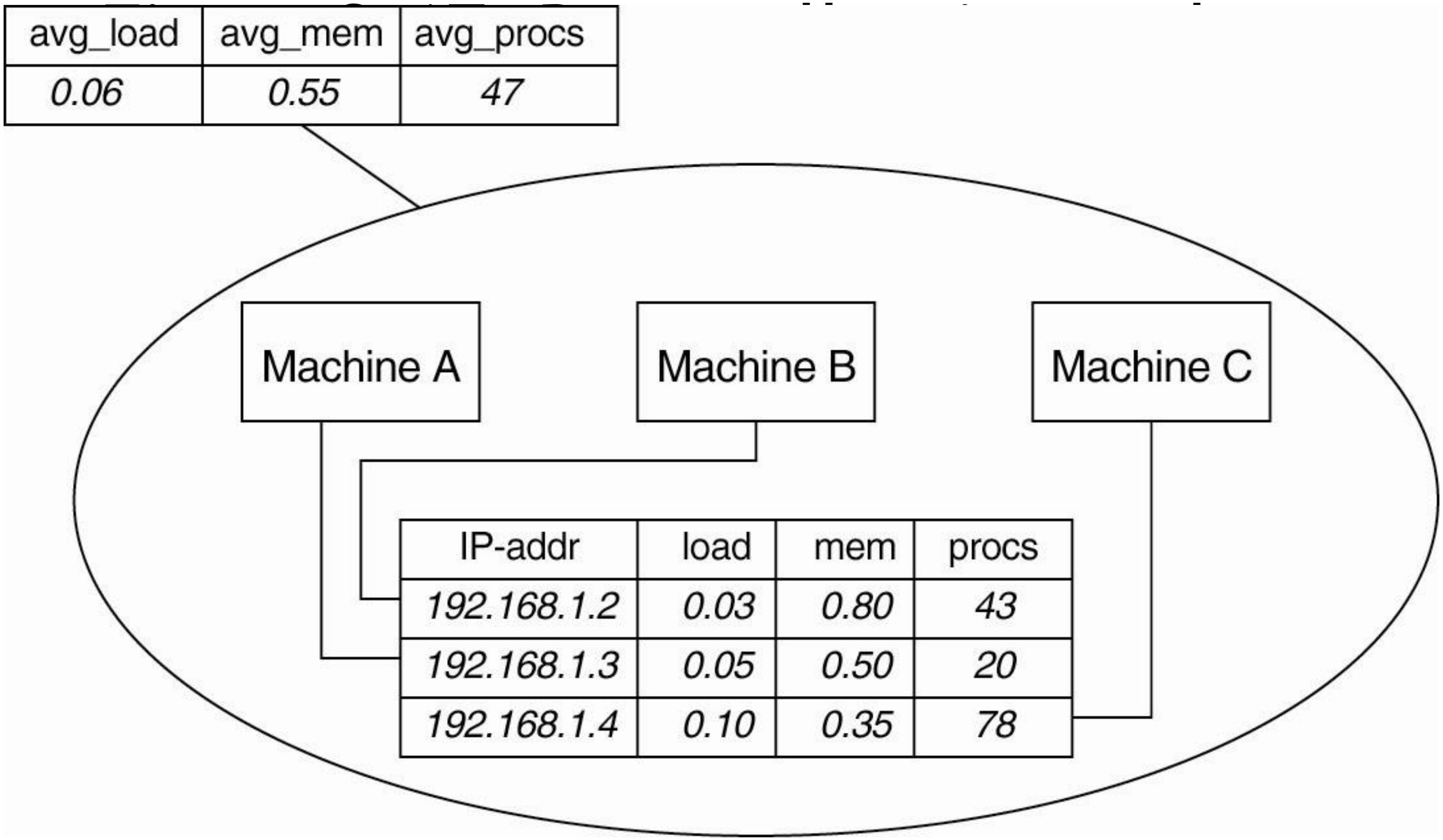
- Replace component on the fly
- Dynamic binding process
- A->B
- Runtime locates appropriate version of B
- Jade

Self-*: Feedback Control Model

Autonomic Computing



Self-*: Need Measurement Infrastructure



avg_load	avg_mem	avg_procs
0.06	*0.55*	*47*

IP-addr	load	mem	procs
192.168.1.2	*0.03*	*0.80*	*43*
192.168.1.3	*0.05*	*0.50*	*20*
192.168.1.4	*0.10*	*0.35*	*78*

Next Time

Next topic: Communication

Read Chapters 3.4.2-3.4, 4.1-4.3 TVS, LRPC paper