# 95-843 Service Oriented Architecture

## Review of Distributed Systems Architectural Models

From Coulouris, 5th Ed.

# Definitions, Goals and Concerns

Definition: The architecture of a system is its structure in terms of separately specified components and their interrelationships".

Goal: The structure will meet present and future demands.

Concerns: reliability, manageability, adaptability, cost-effectiveness, security

# Architectural Elements of a Distributed System

- Communicating entities
- Communication  paradigms
- Roles played by communicating entities
- Placement of communication entities

# Communicating Entities

- From a system level: Processes, threads or simply nodes are communicating.
- From a problem level: Objects, Components, Web Services are communicating.
- In asynchronous systems, the client makes a call and continues with other business. Perhaps it provides a means for a response.
- In synchronous systems, the client calls, blocks and waits for the response.

# Communication Paradigms

Interprocess communication (TCP Sockets, UDP Sockets, Multicast Sockets)

Remote invocation (Two way exchange with a remote operation, procedure or method) RPC, RMI, HTTP.

Coupled in time (both parties exist during interaction)

Coupled in space (parties likely know who they are interacting with)

Indirect communication (less tightly coupled and involving a third party)

Communicating to a group be sending a message to a group identifier

Publish-subscribe (AKA distributed event based systems) routes messages to interested parties. One-to-many style of communication.

Message queues (AKA channels) for point-to-point messaging.

Tuple spaces allows for the placement and withdrawal of structured sequences of data.

# Roles and Responsibilities

Entities interact to perform a useful activity.

One entity may act as a client and another as a server.

Each entity may act as a peer.

# Placement of Communicating Entities

- Entities may be placed on a single or multiple machines.
- Data may be cached and services replicated.

- Mobile code (e.g. applets and Java Script).

- Mobile agents or worms.

# Architectural Patterns (1)

## Layered architecture

the vertical organization of services into layers of abstraction:

applications, services layered on the top

middleware between the application and the operating system

operating system

computer and network hardware

# Architectural Patterns (2)

Tiered architecture:

- complimentary to layering
- usually applied to the applications and services layer

- a technique to organize the functionality of a given layer and place this functionality into appropriate servers and onto physical devices
- An application may be described in terms of presentation logic, business logic, and data logic
- Such an application might be built upon two tiers or three.
- This is separation of concerns

Note: presentation logic may present data to a non-human.

# Architectural Patterns (3)

In a two-tier solution, the business logic and user interface may reside on the client and the data logic layer may be placed on the server. This is the classic client server architecture.

Other organizations are possible:

In a three-tier solution, the logical description may correspond directly to the physical machines and processes.

An AJAX application such as Google Maps is an example of a responsive multi-tiered application. New Map tiles (256X256 pixel images) are fetched as as needed.

The thin client approach is a trend in distributed computing. Move complexity into internet based services. Cloud computing and Virtual Network Computing (remote desktop ) are examples.

# Two Commonly Occurring Architectural Patterns in Distributed Systems

- The proxy pattern: the client makes calls on a local object (the proxy) that has the same interface as a remote object. The proxy hides the communication details.
- The brokerage pattern consists of a trio of service provider, service requestor and service broker (typically with lookup and bind operations).