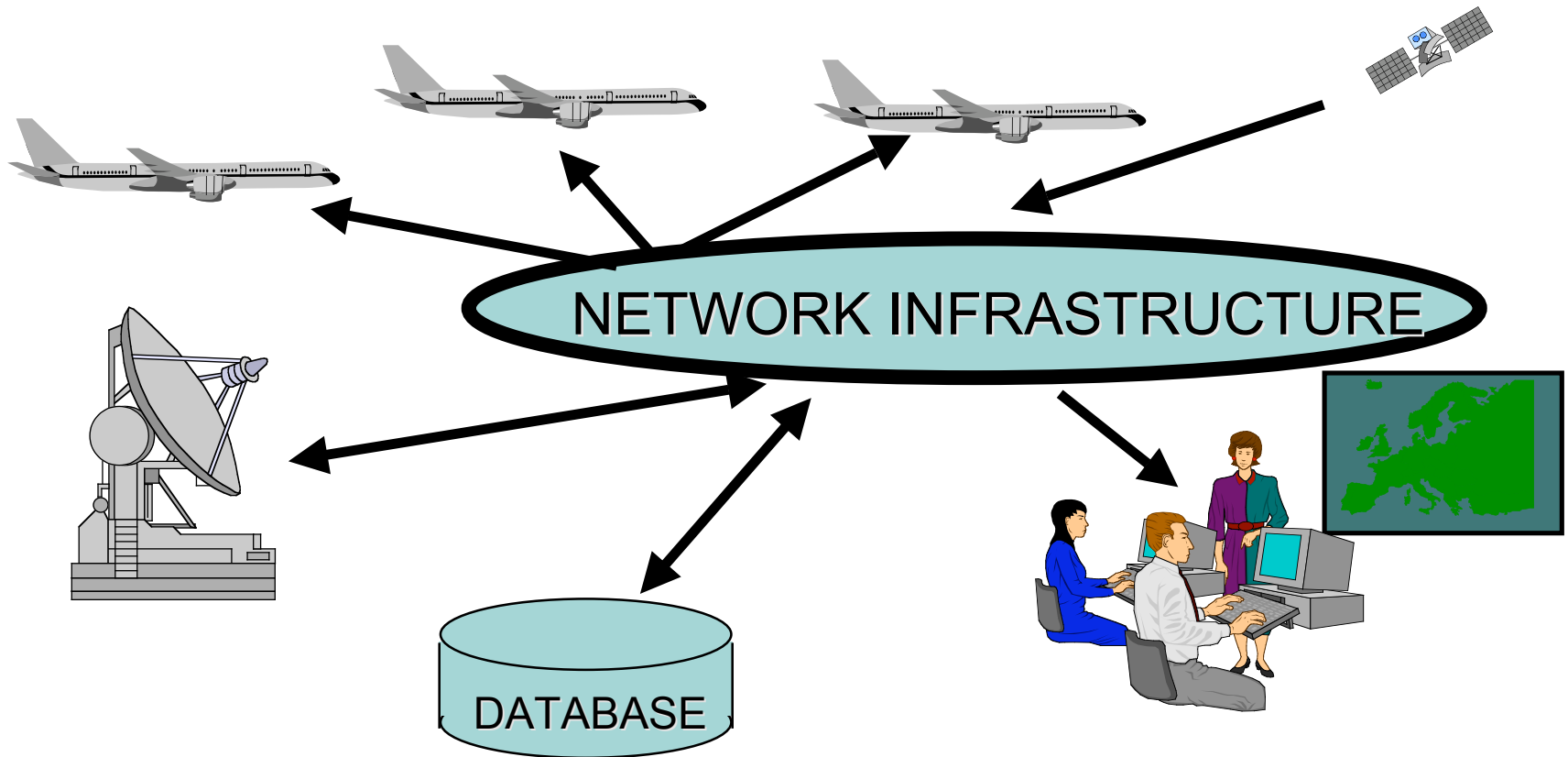


CS603: Distributed Systems

Lecture 2: Client-Server Architecture, RPC, Corba

ATC Architecture



HOW WOULD YOU START BUILDING SUCH A SYSTEM?

Outline

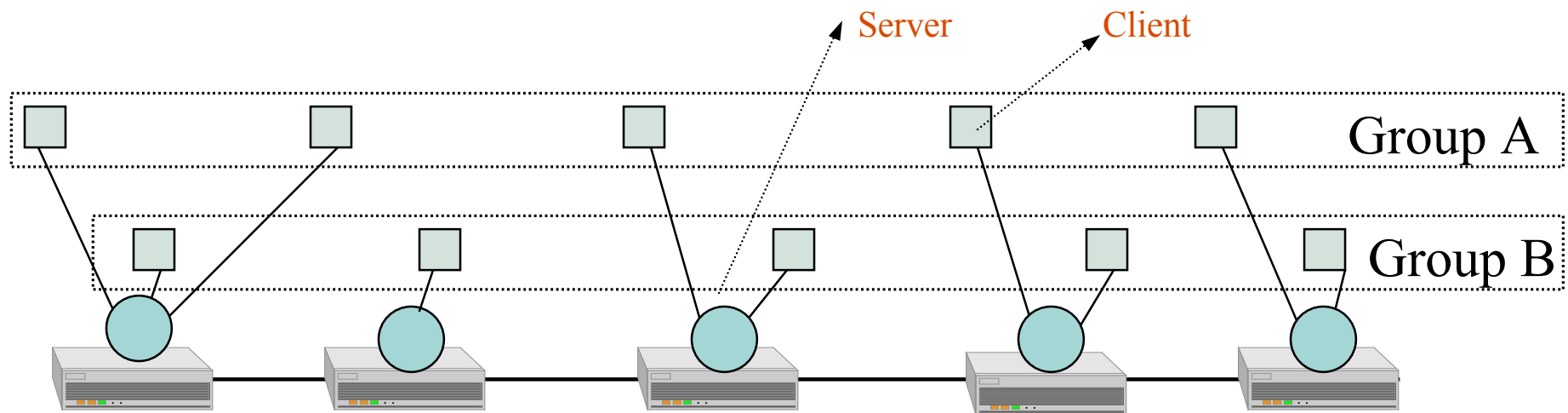
- Technologies/Protocols used in Designing Distributed Systems
 - Client/Server
 - RPC
 - Corba
 - J2EE
 - .NET
 - Web Services



Client/Server Architecture

- Functionality is partitioned in a set of services provided by a set of servers
- Clients (applications) interact with each through the servers
- Examples:
 - File servers
 - Database servers
 - Network name servers
 - Network time servers
 - Mail servers
 - Web servers

Example: Group Communication Systems

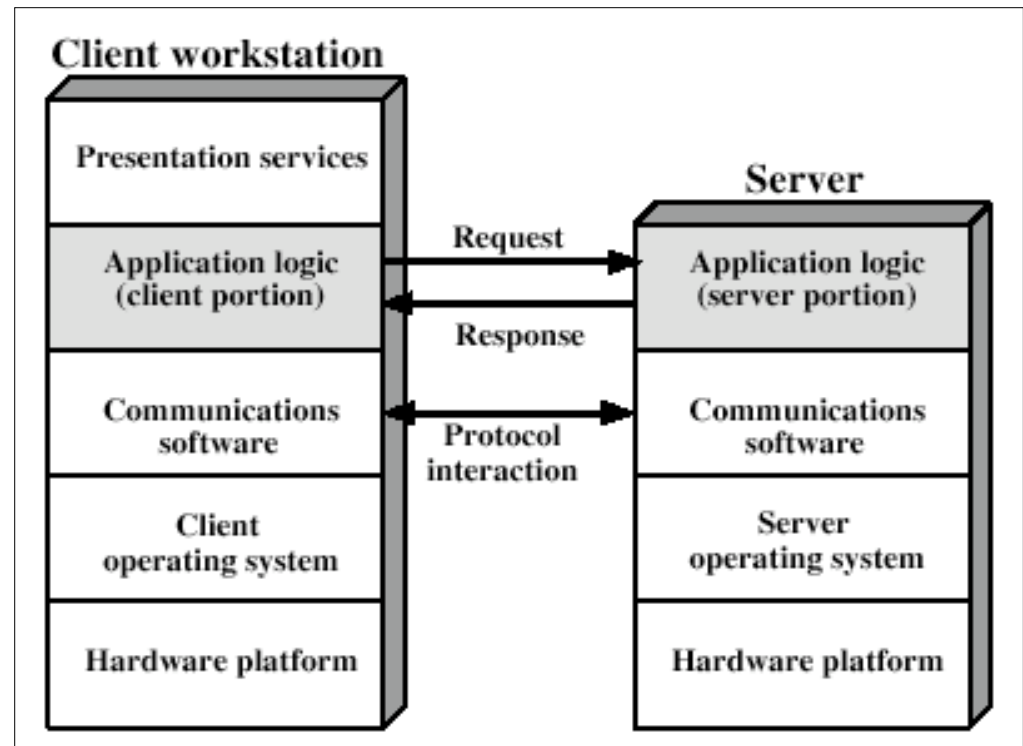


- Reliable and ordered message delivery
- Group membership service (list with connected members)

**Clients do not connect with each other,
they communicate using the GCS servers**

Client/Server General Architecture

- Client must 'bind' to a server
- Standard services run on well-known ports
- Clients discover services (directory of servers providing a desired service)



Styles of Client/Server

- **Stateless**: server does not keep any information between requests. There may be a shared state in the form of cache, but the correct function does not require the shared state to be accurate.
- **Stateful**: server remembers information between requests. Client may take local actions based on accuracy of information.
- Can you think about examples in each case?

