Distributed Systems Development

Paulo Gandra de Sousa

psousa@dei.isep.ipp.pt

MSc in Computer Engineering

DEI/ISEP

Programação de Sistemas Distribuidos

Paulo Gandra de Sousa

psousa@dei.isep.ipp.pt

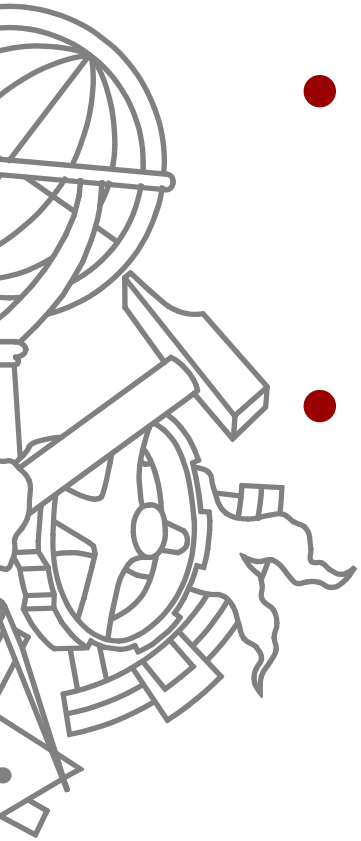
Mestrado em Engenharia Informática

DEI/ISEP

Disclaimer

- Parts of this presentation are from:
  - Tannembaum
  - Coulouris
  - Doug Terry (CS 294)

Today's lesson



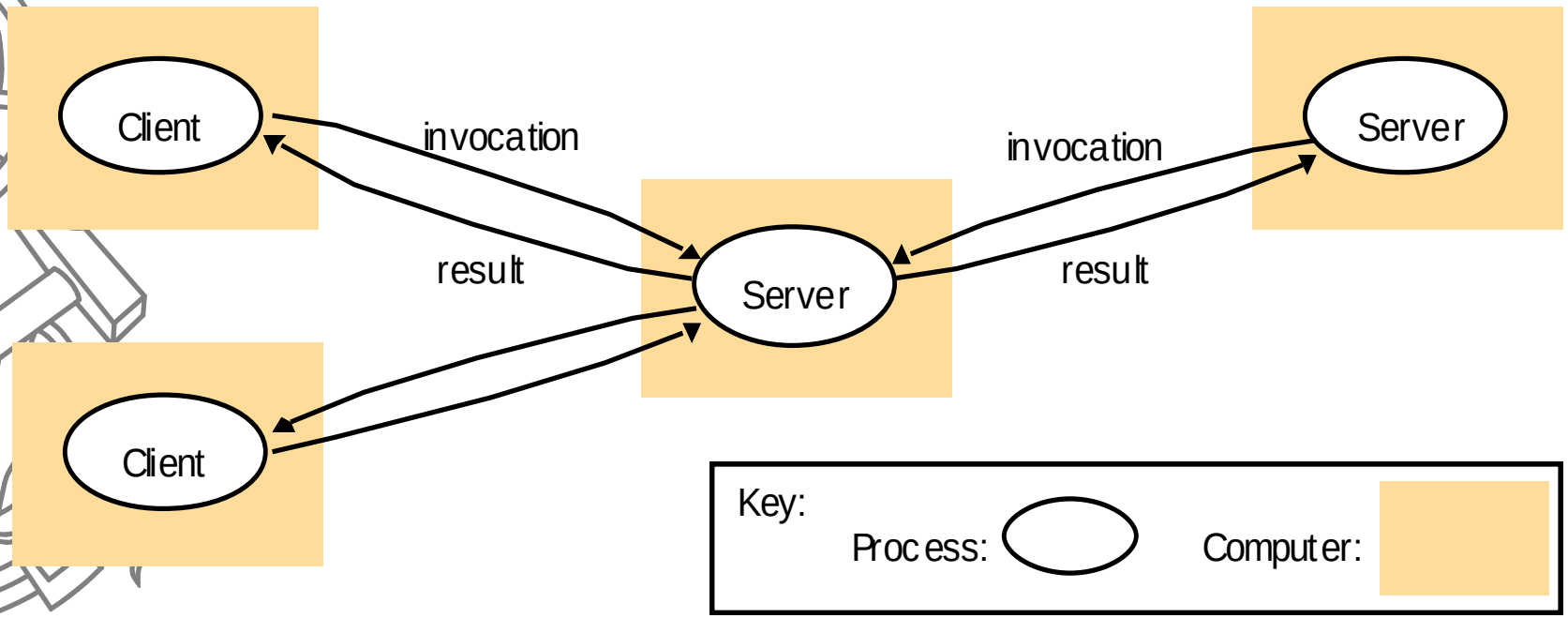
- Models of distributed computing
  - Architectural styles
  - Clients, servers and state
- Communication
  - Types of communication

MODELS OF DISTRIBUTED COMPUTING

Client/Server

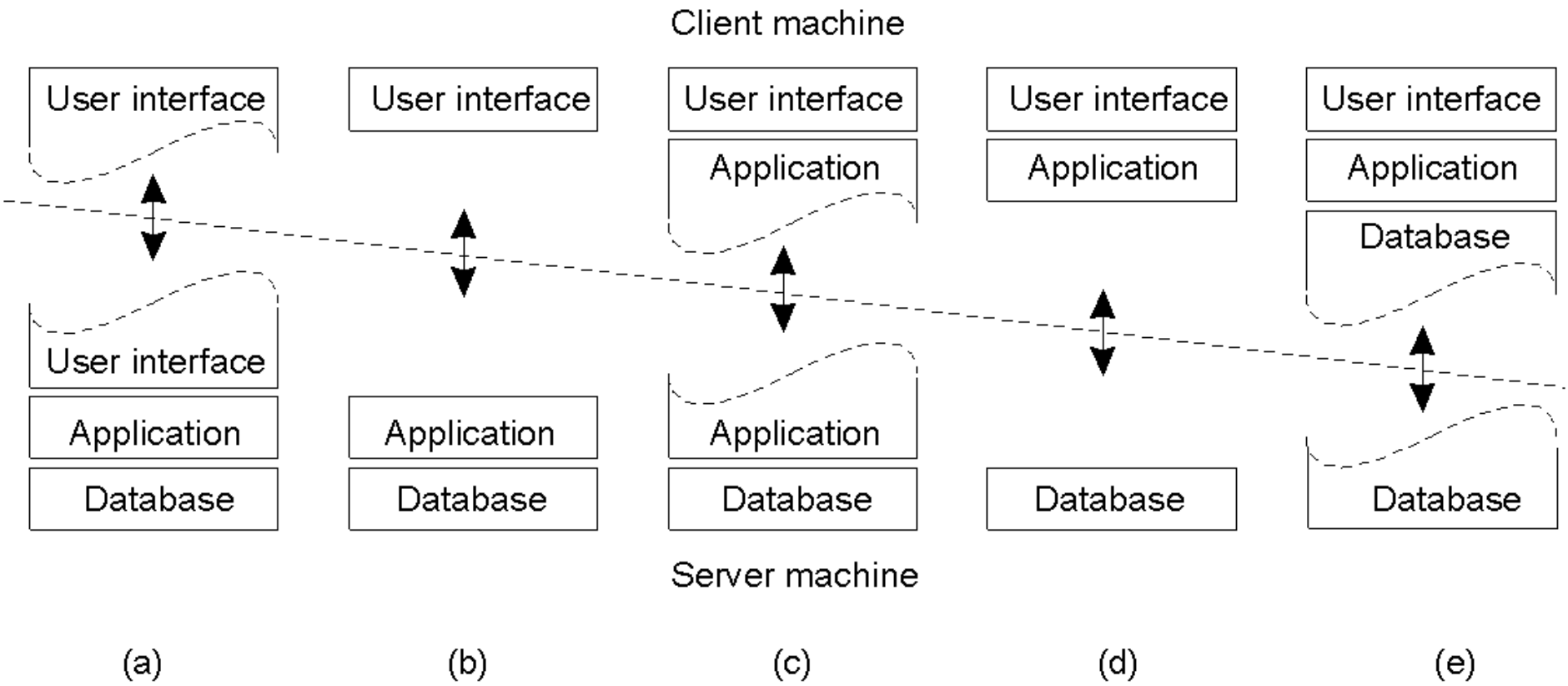
- functional decomposition
  - e.g. workstations that use a mail server to exchange messages

Clients invoke individual servers



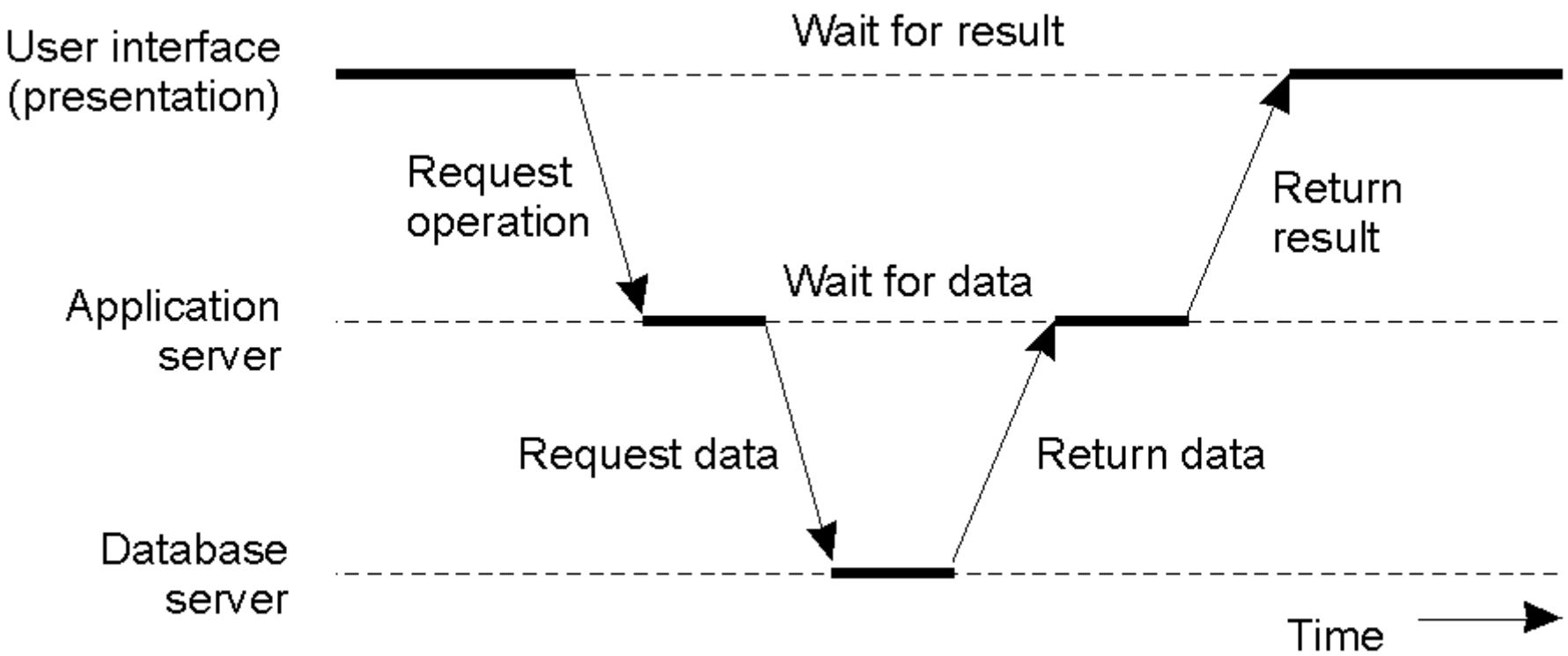
Multitiered Architectures (1)

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



Multitiered Architectures (2)

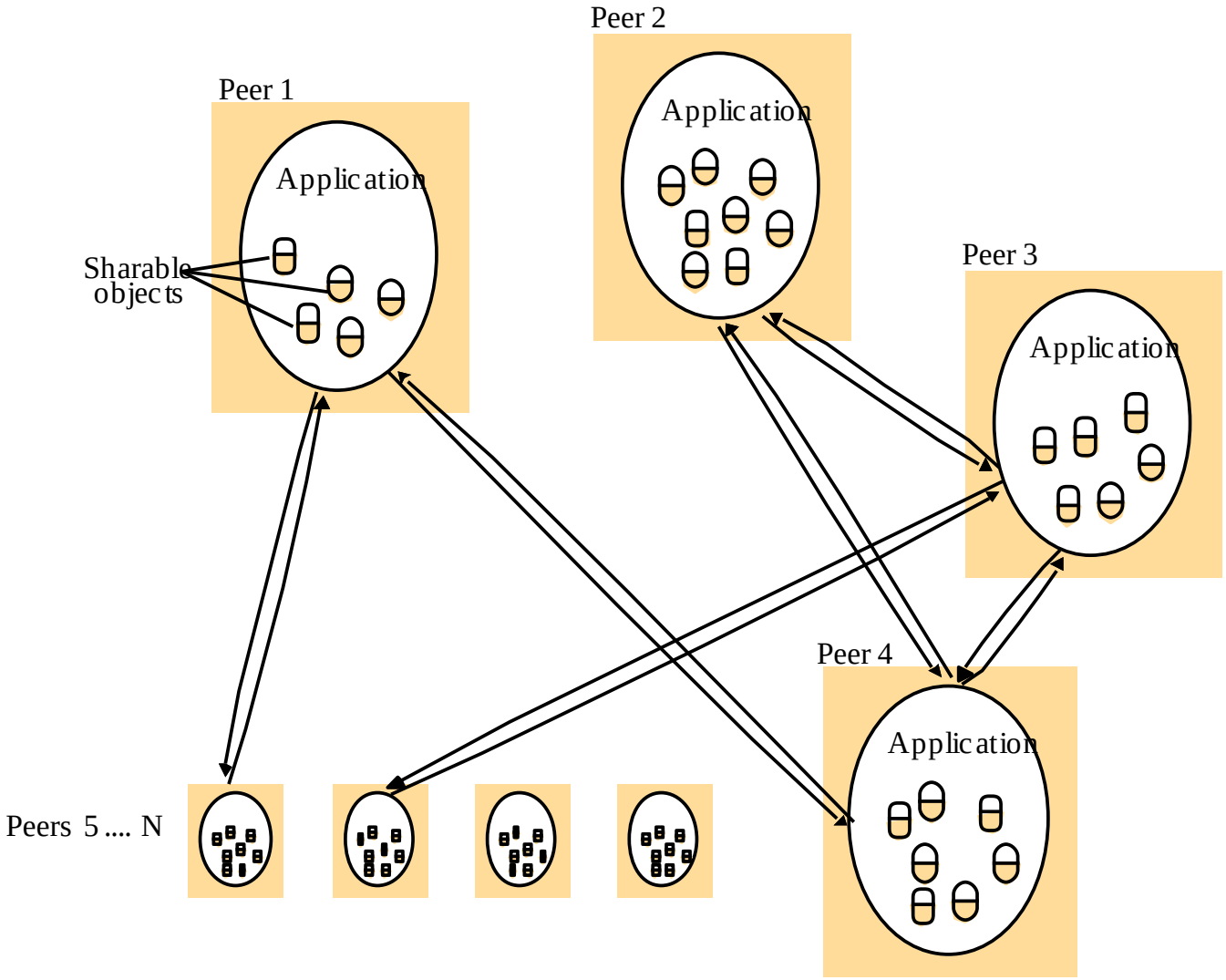
- An example of a server acting as a client.



Peer-to-peer

- physical decomposition into identical components
  - e.g. the exchange of mail messages among hosts

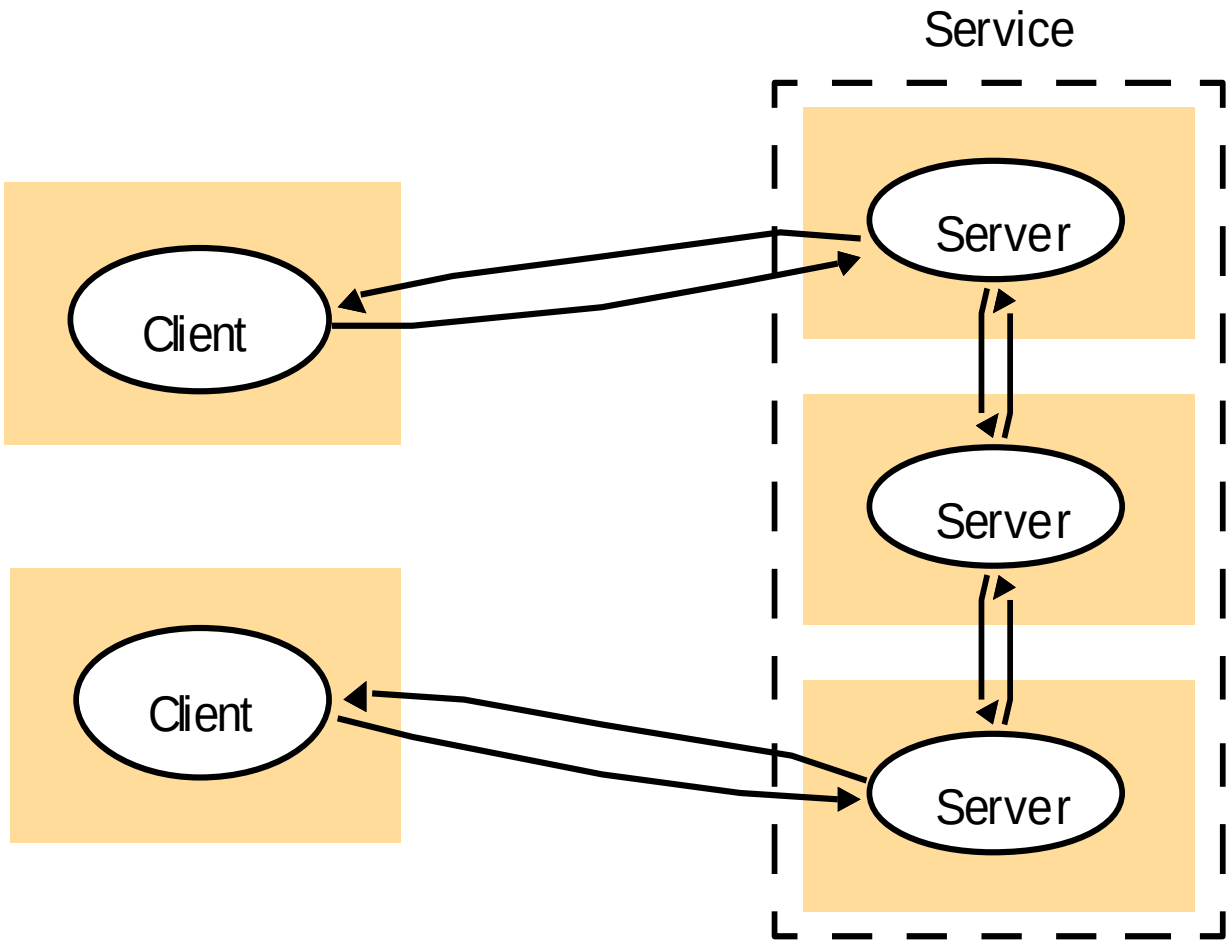
A distributed application based on peer processes



Hybrid

- physical and functional decomposition
  - e.g. a collection of servers provide a service for some collection of clients

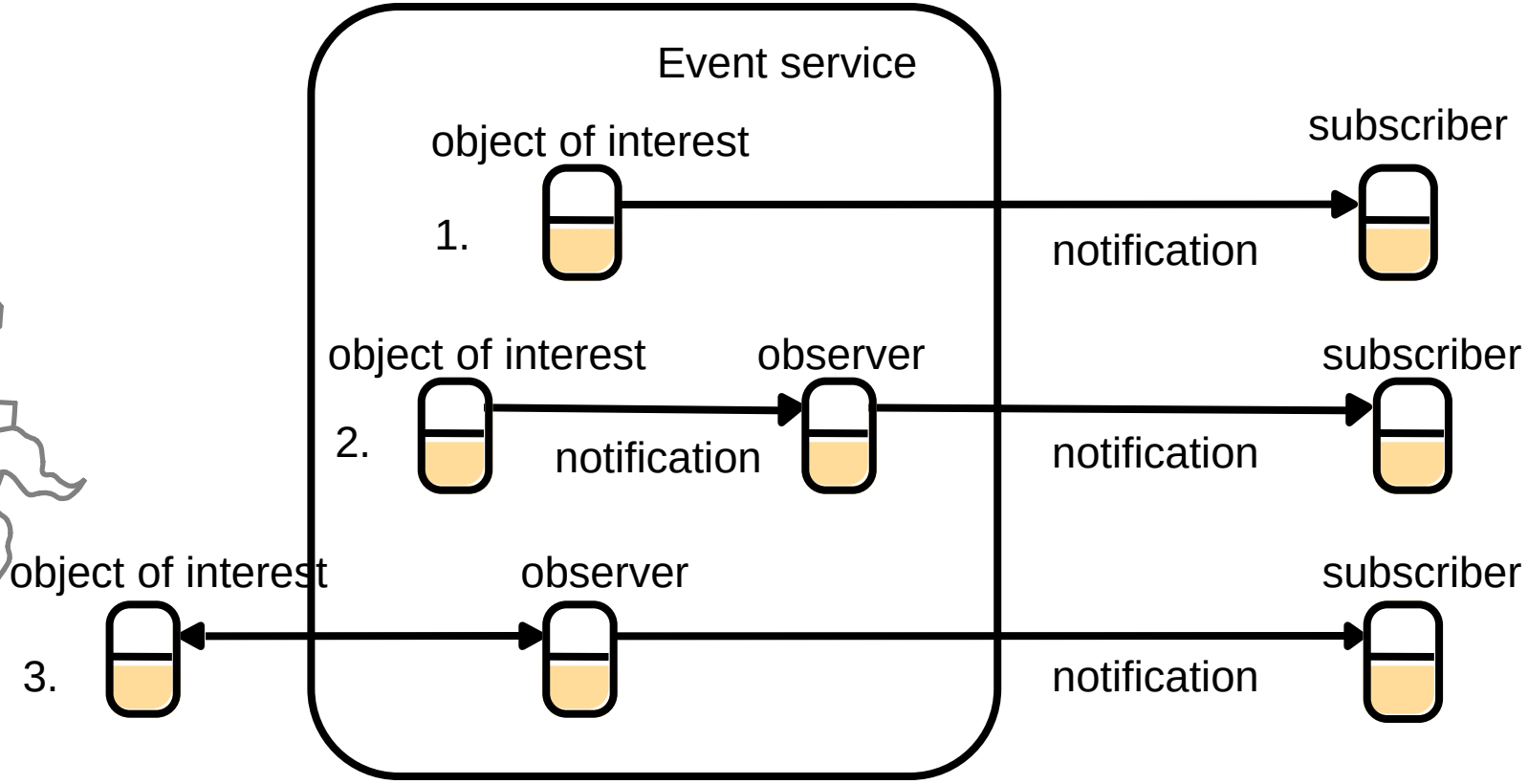
A service provided by multiple servers



Publish/subscribe

- Event based, topic-of-interest decomposition
  - e.g. news alert

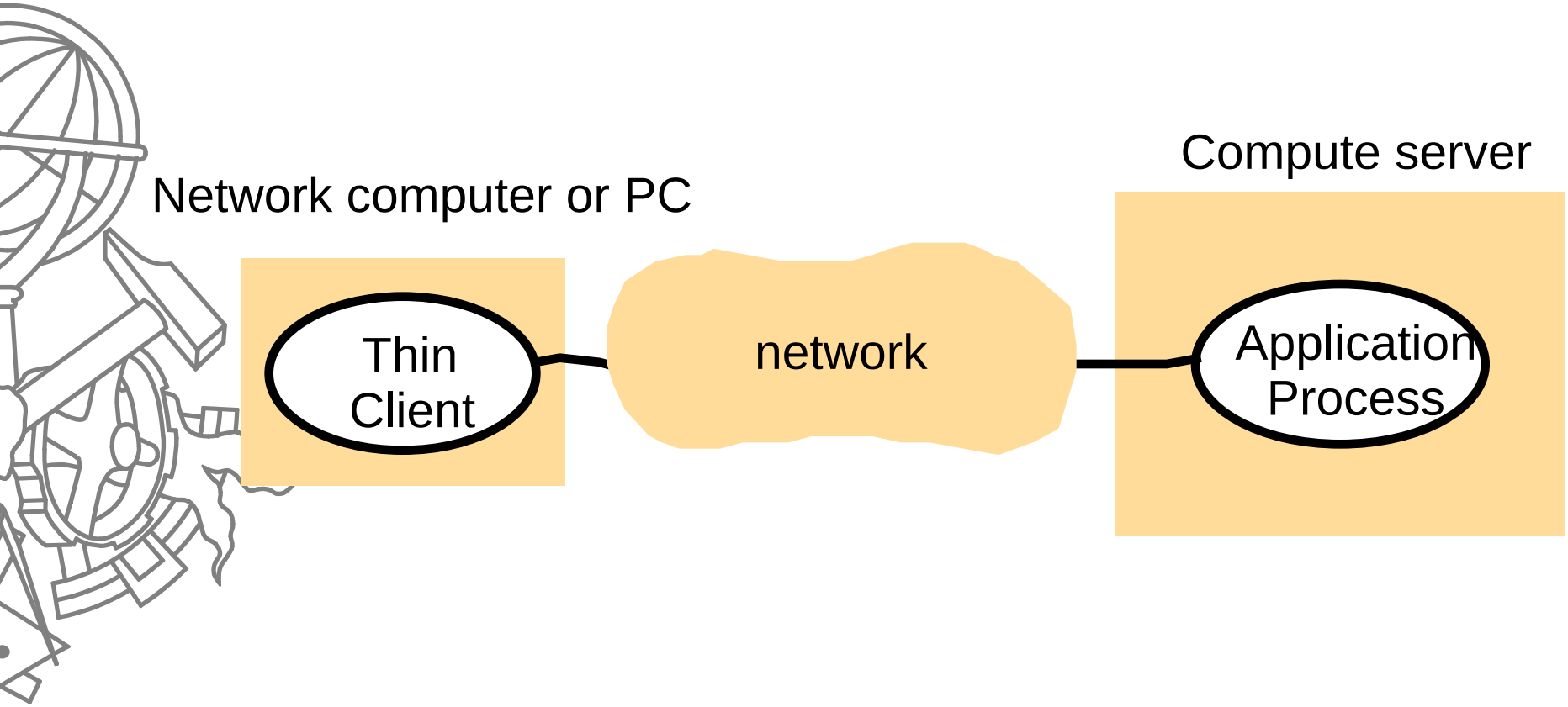
Architecture for distributed event notification



Of the client…

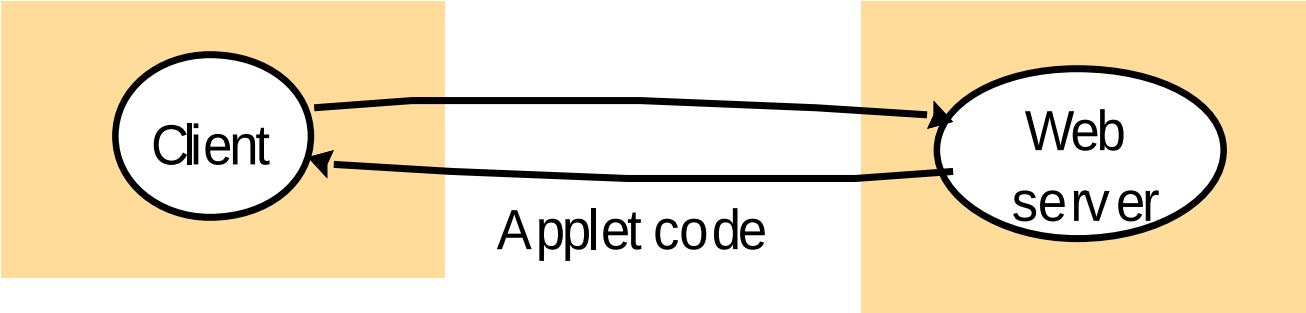
- Thin vs. Fat
  - Presentation only (e.g., DEI webmail) or full-fleged app with remote server (e.g., Outlook)
- Local vs. Downloaded
  - Locally installed software (e.g., Picassa) or downloaded from the server (e.g., applet)

Thin clients and compute servers



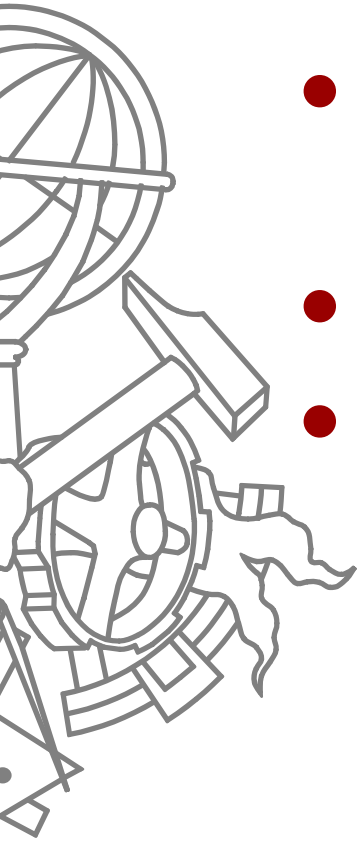
Web applets

a) client request results in the downloading of applet code



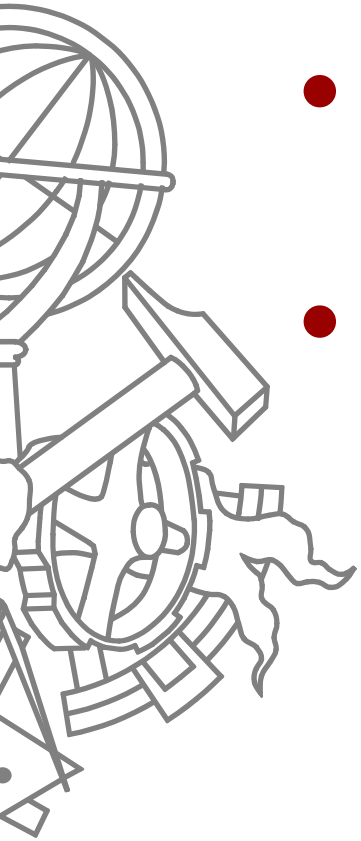
b) client interacts with the applet

Of the server…



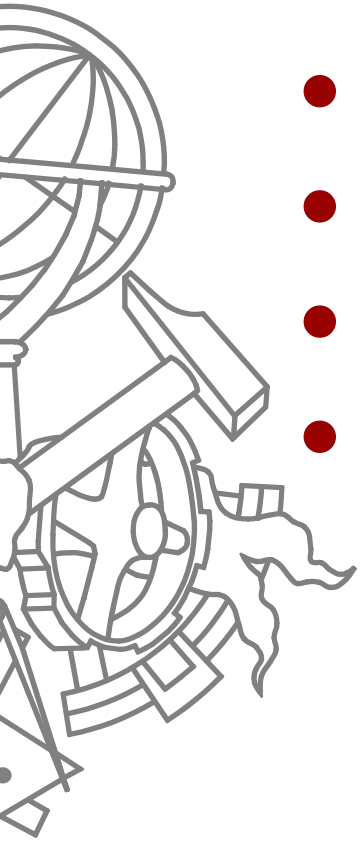
- Single vs. multi-instance
  - cluster
- Well known address vs. Lookup
- Stateless vs. Statefull

How to handle state?



- Server vs. Client state
- Online vs. Offline state
  - Session variables
  - Database/files

Of the nodes…



- Networks of computers
- Single vs. Multi-processor computer
- Homogenous vs. Heterogenous nodes
- Mobile/offline vs. Fixed/online nodes

Exercise

- Remember the example DS you provided in the last session.
- How does it relate to these models?

Exercise

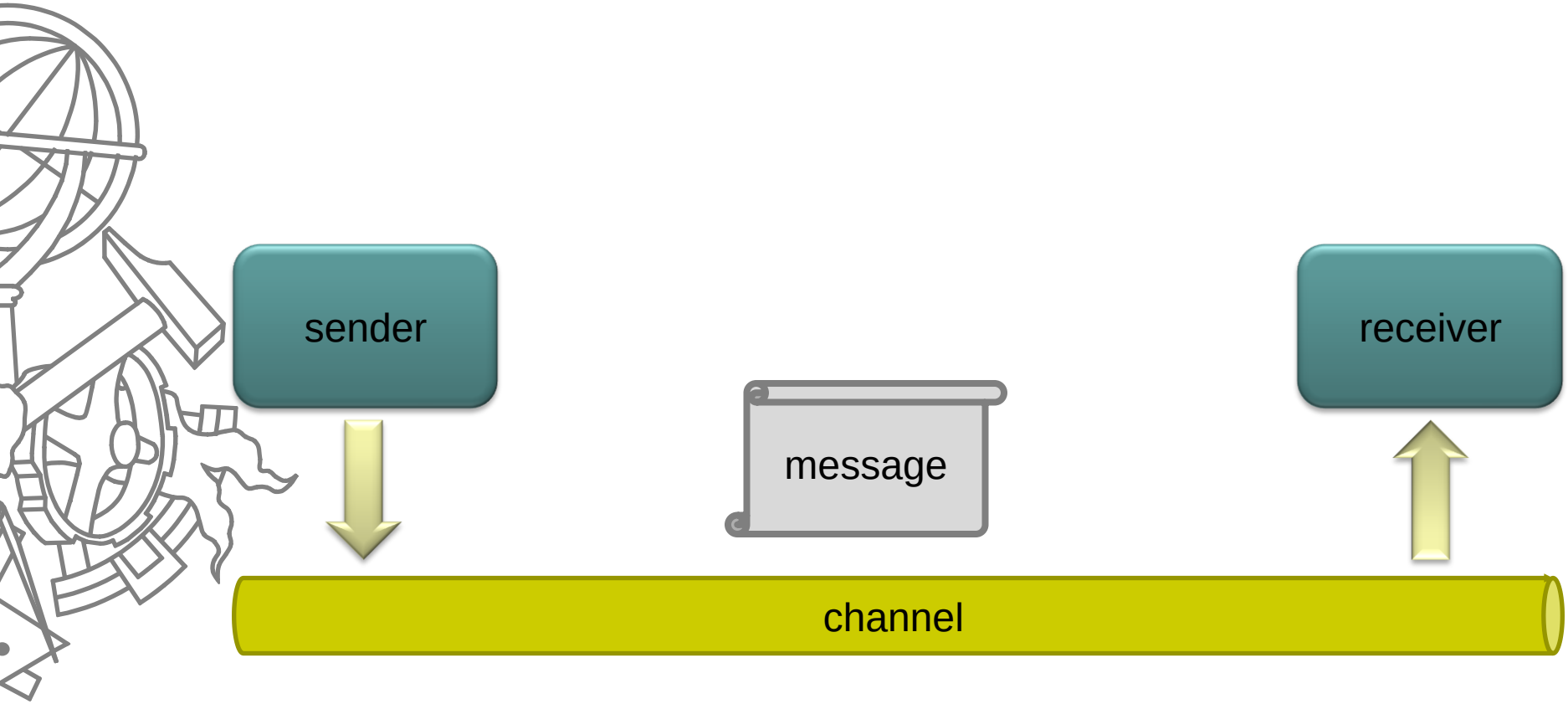
- If you haven't done yet, give examples of:
  - Publish-subscriber
  - Thin client
  - Fat client

COMMUNICATION

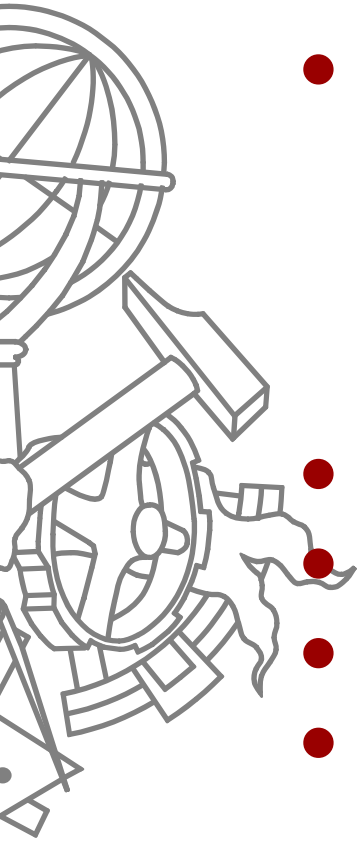
Characterization

- The parties
- The act of communicating
- Dialogues

The parties



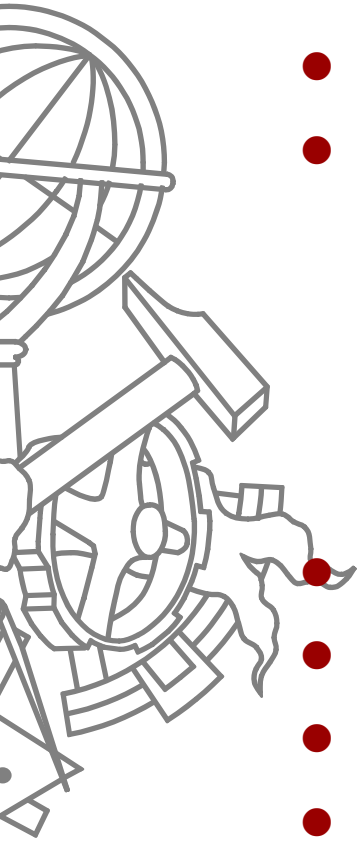
Chanels



- The medium thru where the message goes that connects the sender and the receiver
  - E.g., carrier pigeons
  - E.g., TCP

- Uni or bidirectional
- Secure or insecure
- Reliable or unreliable
- May provide aditional services
  - E.g., logging

The Message



- What you want to transmit
- Usually divided in two parts
  - Infra-structure's info (e.g., header)
  - Payload
    - Structured or unstructured
    - Attachments
- Ciphered or plain
- Signed or not
- Priority
- Importance
- Sensitivity

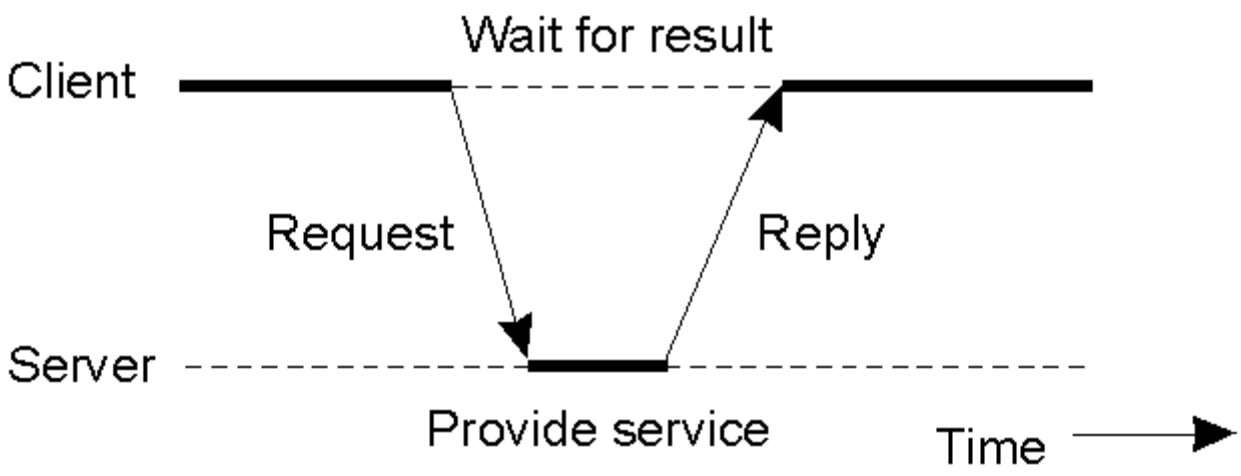
The Sender

- The (role of the) party that wants to transmit some information to another party

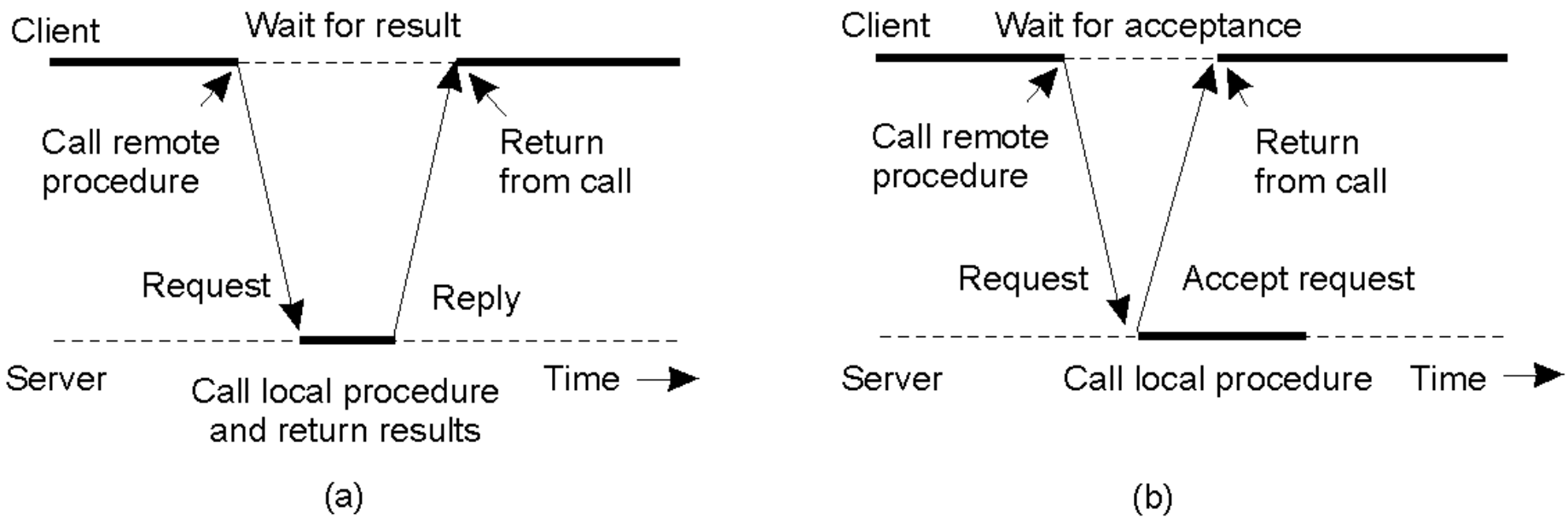
- Does it continue to work after sending the message, or blocks execution?
- Is it a peer, a report or a manager relative to the receiver?
- To whom is it talking to?

Synchronous sender

- General interaction between a client and a server.



Asynchronous sender (1)

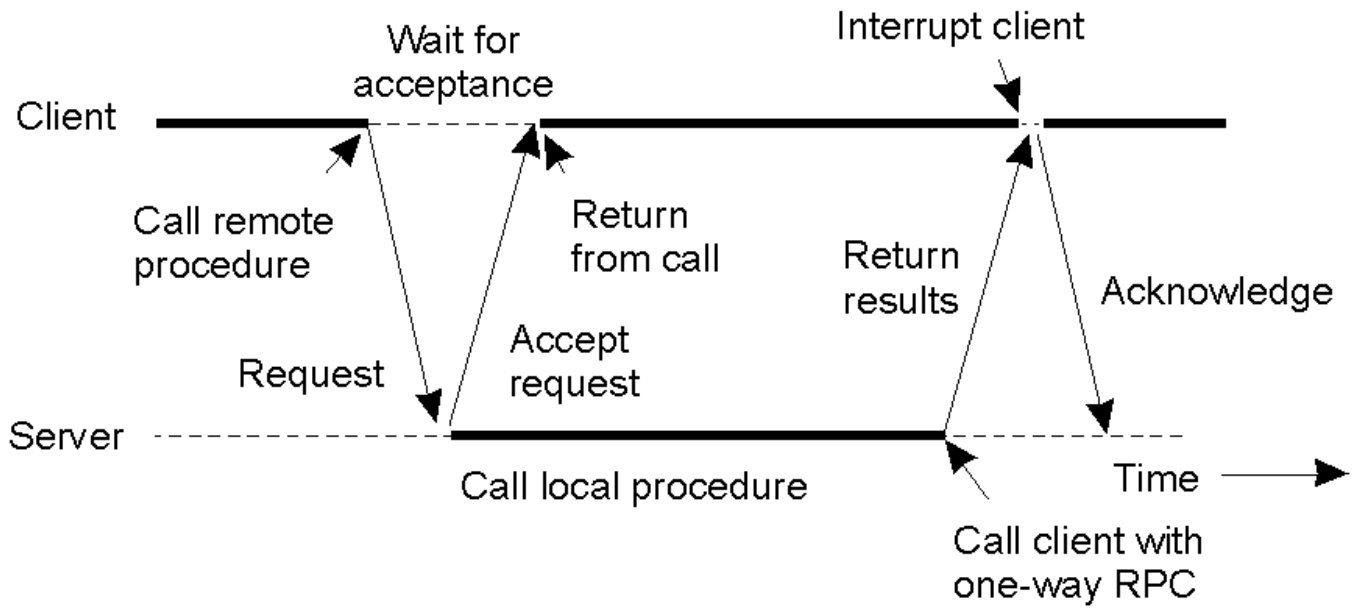


a) The interconnection between client and server in a traditional RPC

b) The interaction using asynchronous RPC

Asynchronous sender (2)

- A client and server interacting through two asynchronous RPCs



To whom it may concern

- Unicast
  - The message is intended for one, and only one, specific receiver
- Multicast
  - The message is intended for a list of designated receivers
- Broadcast
  - The message is placed in *ether*, anyone (or no-one) can read it, but the sender typically has no way of knowing it

The receiver

- The (role of the) party that receives a message

- Can the message be ignored?
- Can it be dealt later on or needs immediate attention?
- Is it a peer, a report or a manager relative to the sender?

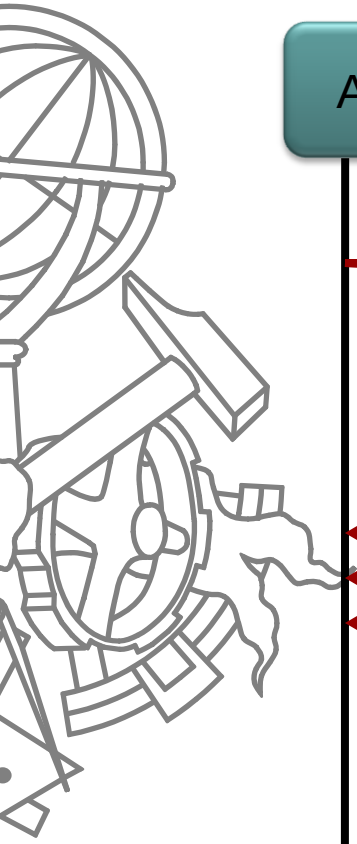
Basic Dialogues

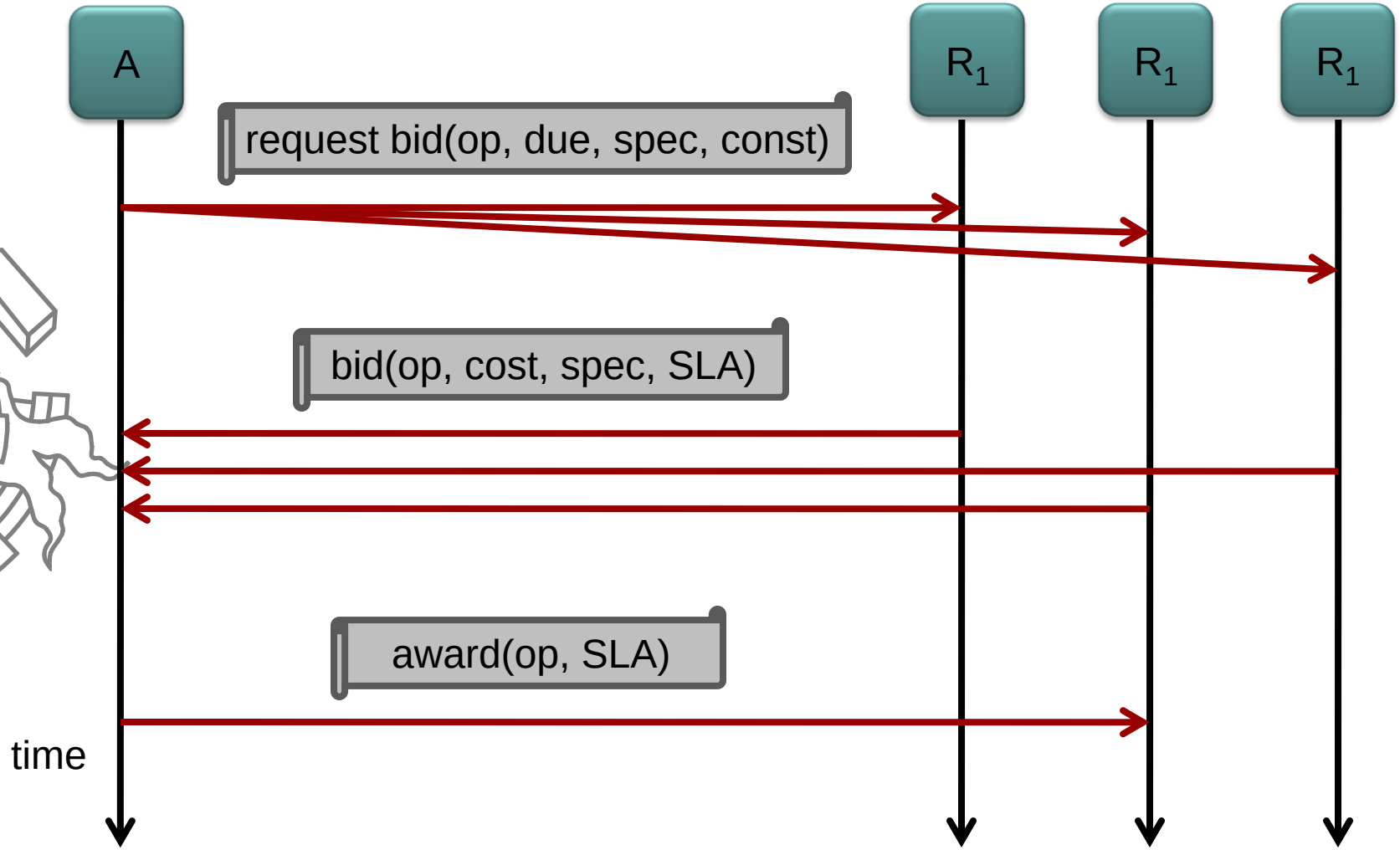
- Command/Do-Report
  - The sender party tells the receiver party what to do and the receiver is obliged to do it
- Inform
  - The sender party tell the receiver something; the receiver is free to do whatever it wants with it; the sender expects no callback
- Request/Process-Respond
  - The sender ask for the receiver to perform some task; the receiver may choose not to perform it and may or may not callback on the sender

Interaction protocols

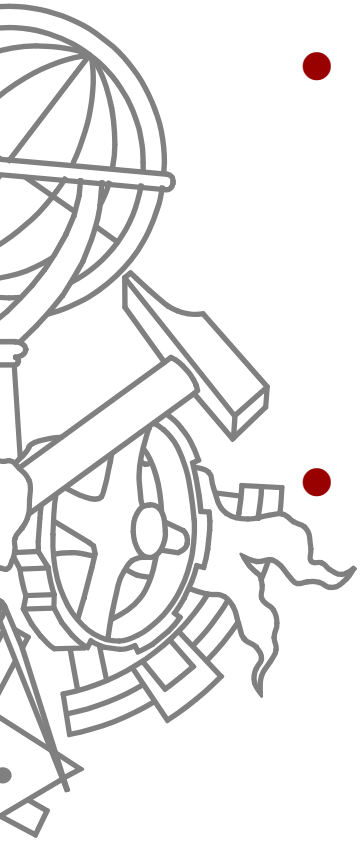
- Specific dialogues between two or more parties consisting of multiple steps (basic dialogues)
  - Example: Contract Net Protocol

Contract Net Protocol





Exercise



- Suggest an interaction protocol for an ATM machine's "Withdrawal" use case
- Identify sender, receiver, messages and channels and describe their characteristics for this scenario

Bibliography

- Chapter 2 Tanenbaum
- Chapter 2 & 4 Coulouris
- Smith, R., 1980, “The Contract Net Protocol”. *IEEE Transactions on Computers*, vol. **C-29**(12), pp.1104-1113.