

Sistemas Distribuídos

José Orlando Pereira

Departamento de Informática
Universidade do Minho

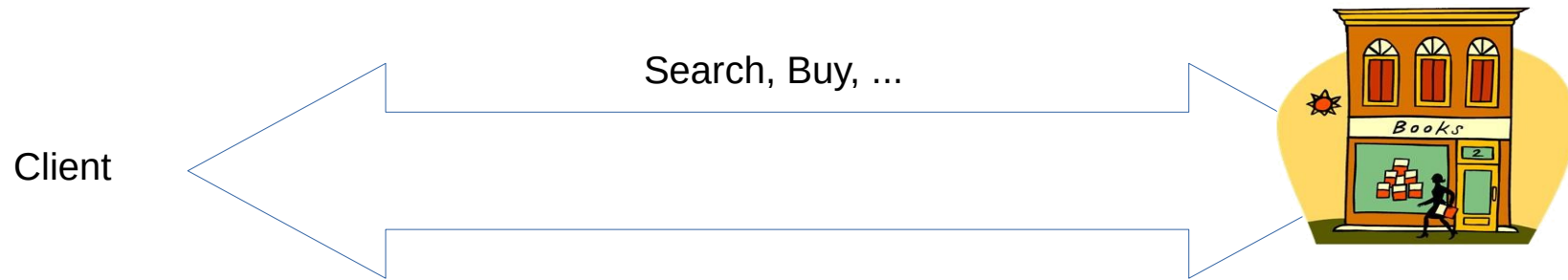


RPC / RMI

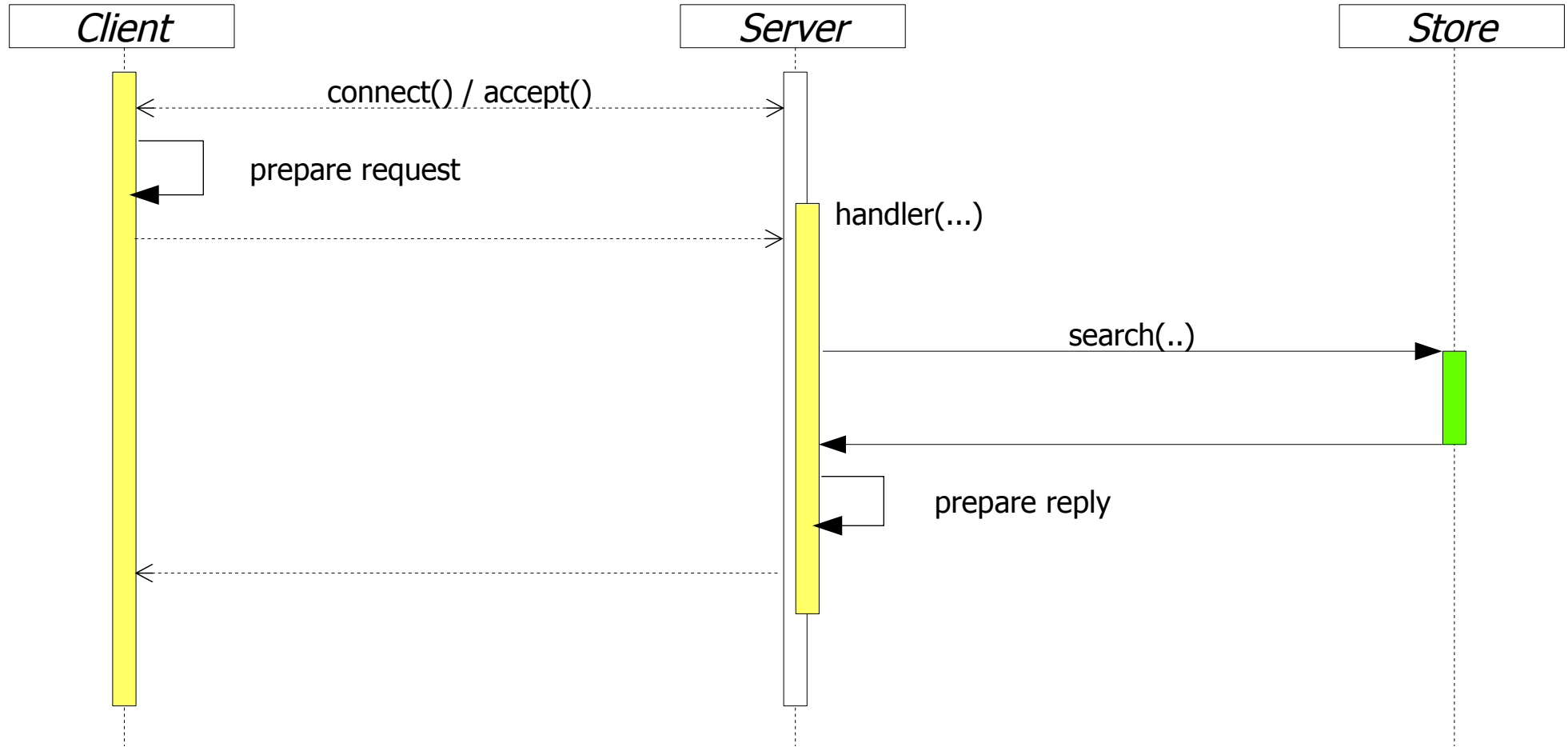
- Middleware that hides client/server interaction as a procedure (method) invocation
- Puts it all together:
 - Communication with sockets
 - Serialization
 - Threading strategies in clients and servers

Example: Book store

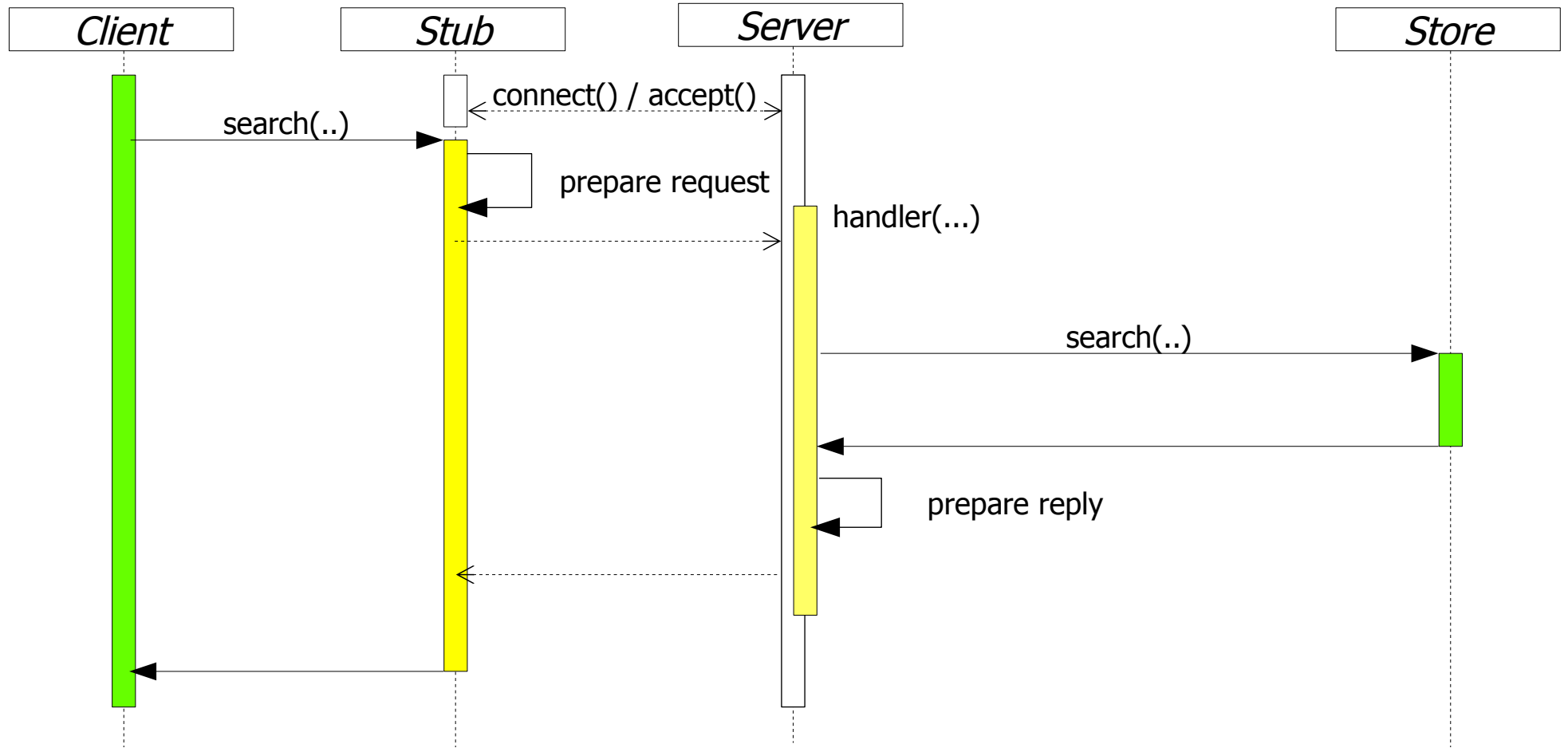
- Book store operations:
 - Search books
 - Book details
 - ...



Client/server with sockets

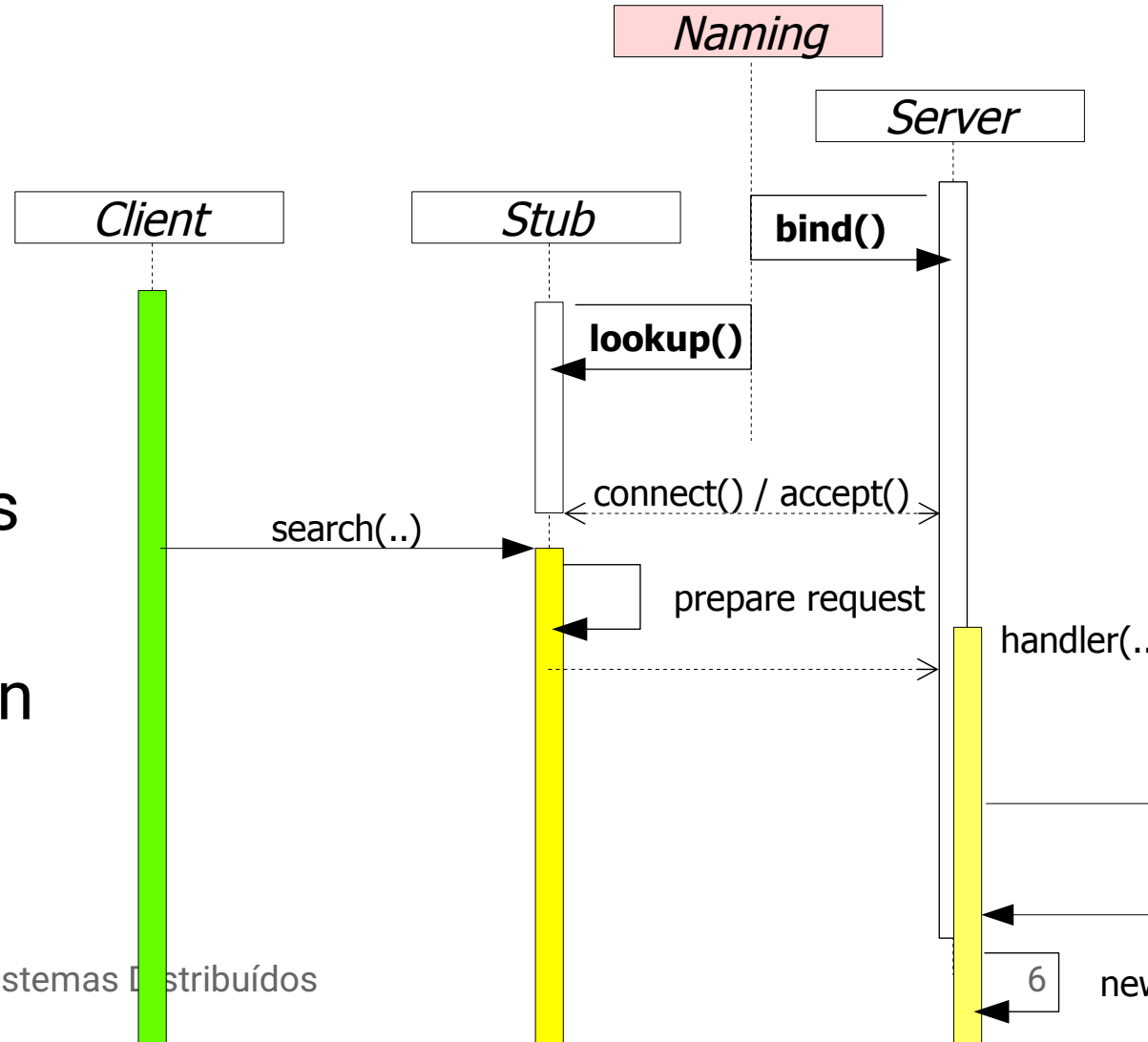


Client encapsulation with stubs



Naming service

- Provides location transparency
- Server registers reference in naming service
- Client looks up address using name
- Lookup encapsulated in client code



Code generation

- Stub and server code is mechanically determined by the interface protocol
- Can be generated from a description
- Code first:
 - Write code, generate stubs using reflection
- Protocol first:
 - Write abstract interface definition, generate stubs with a compiler
 - Uses an Interface Definition Language (IDL)

Parameter passing

- Parameters are copied from client to server
- In some cases, they can be copied back:
 - Pointer parameters in C
 - Objects in Java (not int, etc.)
- Parameters can be labeled as in, out or in-out

Error handling

- Connection problems / server not available cannot be hidden:
 - No corresponding situation in non-distributed system
- Possible semantics:
 - Exactly once: Ideal, hard in a real system
 - At least once: Try repeatedly until acknowledged
 - May block forever
 - Valid only for idempotent operations
 - At most once: Try once and throw exception

Example: gRPC

- Protocol first with Protobuf interface language
 - Multiple target languages, little effort in total transparency for any of them
 - Single-input, single-output parameters, described as Protobuf messages
 - Support for streaming input and output

