VSR://EDU/SSE

Software Service Engineering

WS 2025/2026 – 2. Tutorial

Valentin Siegert M.Sc.

Maheshika Walpola M.Sc.

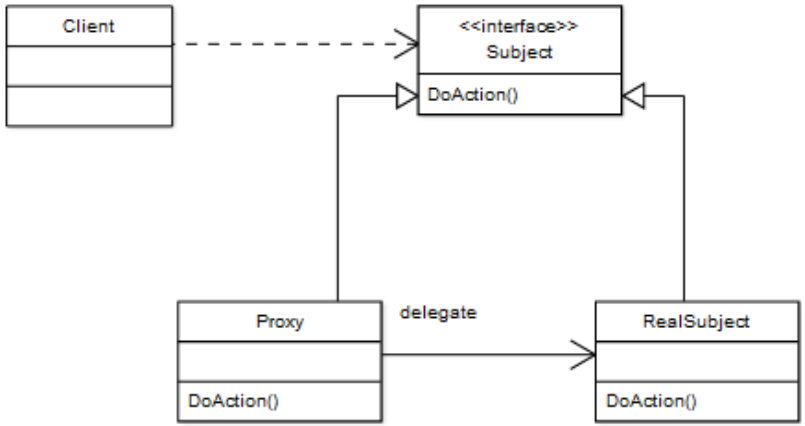
*VSR.Informatik.*TU-Chemnitz.*de*

1 Task 1

Get informed about the Proxy software design pattern.
Create a proxy for the Calculator class from the first tutorial.
The proxy should cache last 10 calculations and return the cached results when the new and the cached operation/operands pairs match.

The Task1-Template.zip gives you a good starting point if you have not finished the first tutorial.

- Problem: control access to objects:
  - Delay creation / initialization of real objects
  - Access mediation and control
  - Caching
- Solution: Substitution by a Proxy



Task 2

Answer the following questions:

1. What is Middleware?

Middleware

- The software layer that lies between the operating system and applications on each side of a distributed computing system in a network

Krakowiak, Sacha. "What's middleware?". ObjectWeb.org

Answer the following questions:

1. What is Middleware?
2. Which services are usually provided by Middleware?

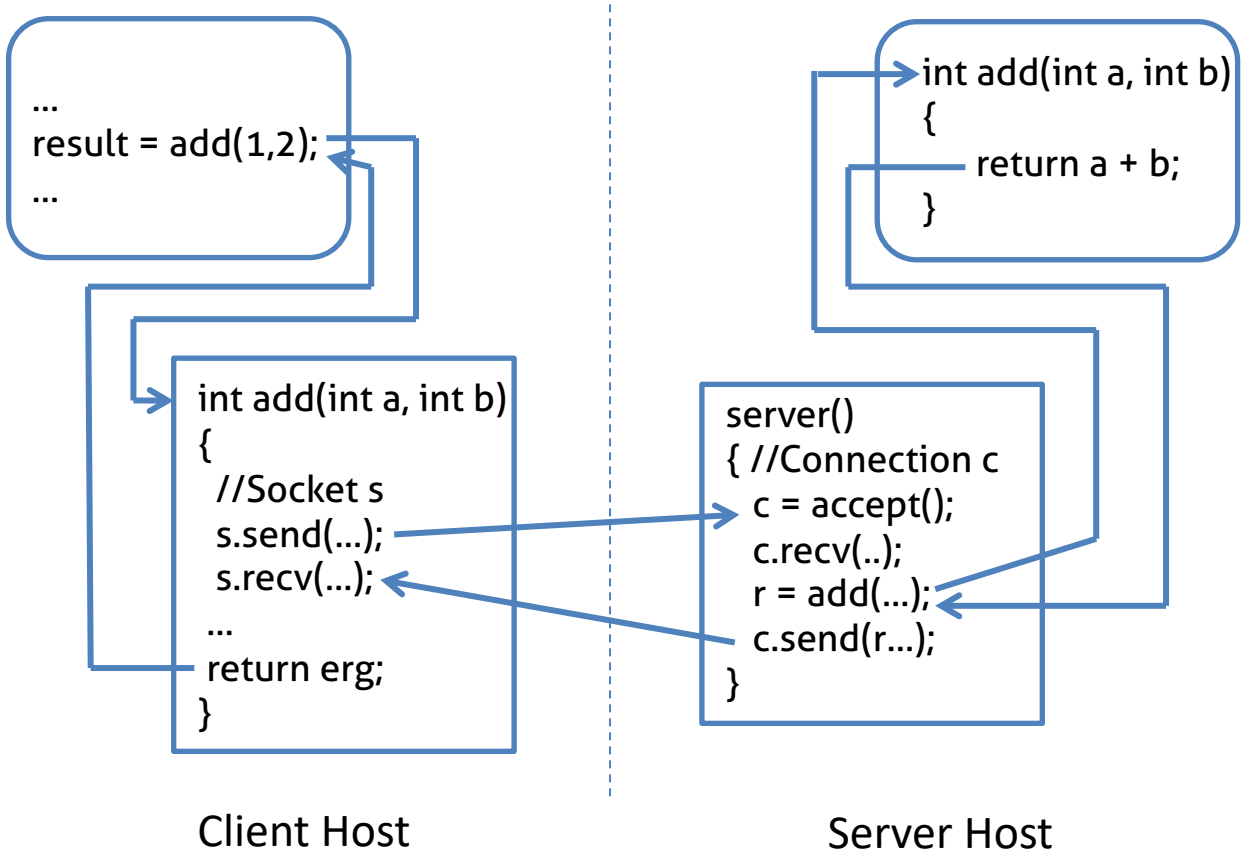
Middleware Services

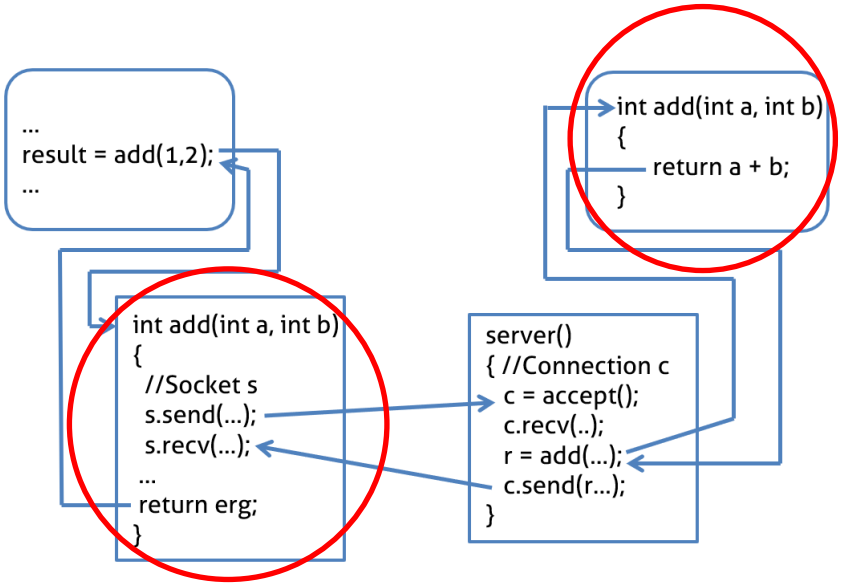
- Goal:
  - Connect heterogeneous network and software systems
- Toolbox:
  - Messaging facilities
  - Session management
  - Transaction management
  - Security services
  - Directory services

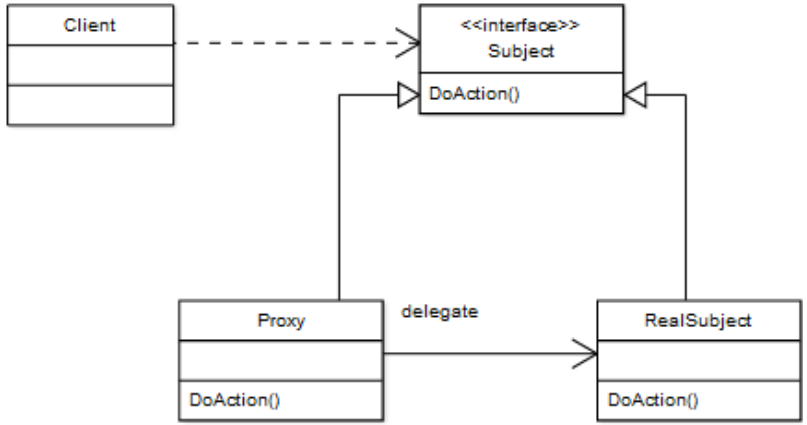
Answer the following questions:

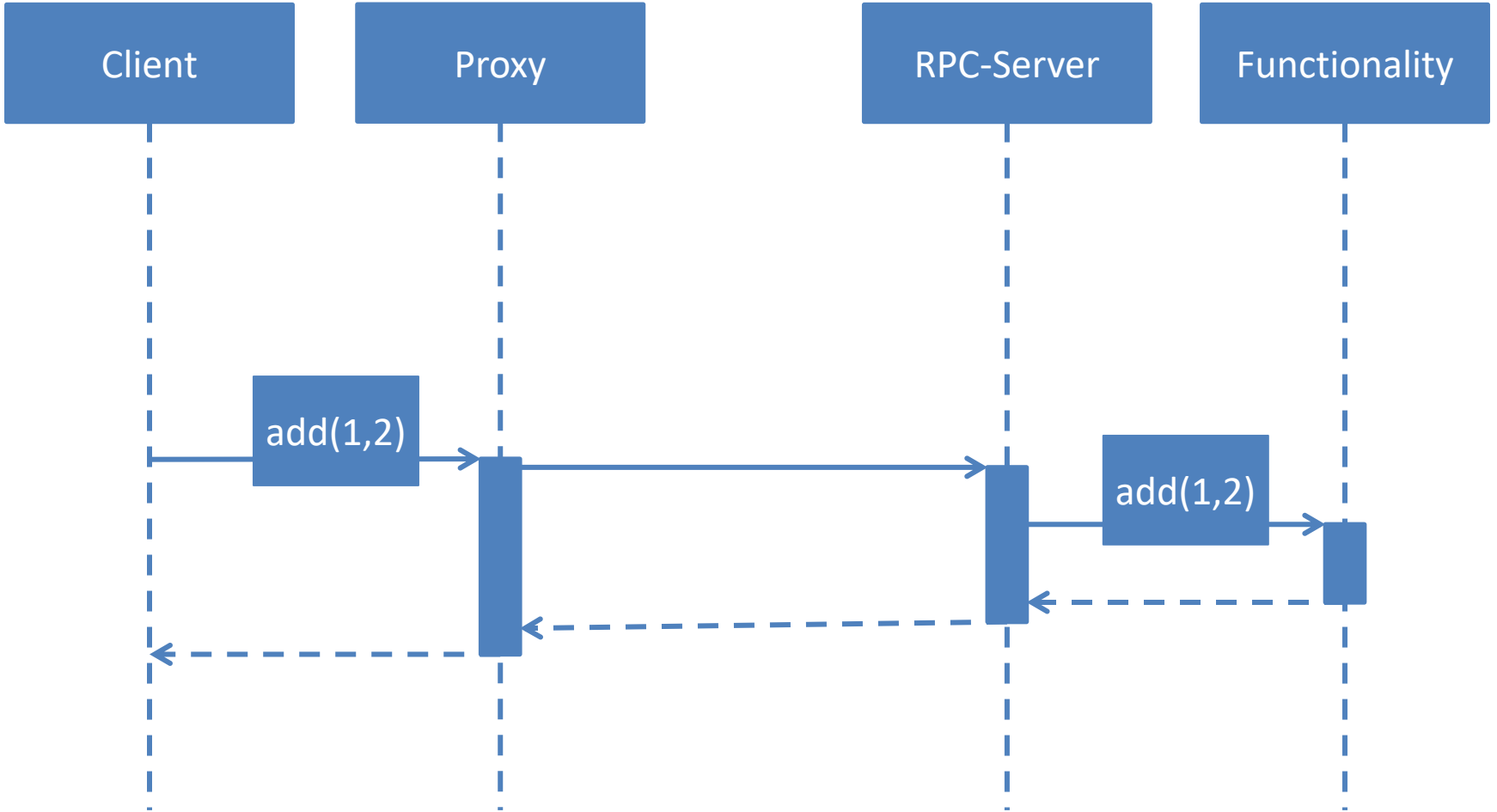
1. What is Middleware?
2. Which services are usually provided by Middleware?
3. What is Remote Procedure Call?

RPC


...
result = add(1,2);
...
int add(int a, int b)
{
//Socket s
s.send(...);
s.recv(...);
...
return erg;
}
int add(int a, int b)
{
return a + b;
}
server()
{ //Connection c
c = accept();
c.recv(..);
r = add(...);
c.send(r...);
}
Client Host
Server Host


...
result = add(1,2);
...
int add(int a, int b)
{
//Socket s
s.send(...);
s.recv(...);
...
return erg;
}
server()
{ //Connection c
c = accept();
c.recv(..);
r = add(...);
c.send(r...);
}
int add(int a, int b)
{
return a + b;
}


Client
<<interface>>
Subject
DoAction()
Proxy
DoAction()
delegate
RealSubject
DoAction()


Client
Proxy
RPC-Server
Functionality
add(1,2)
add(1,2)

Task 3

a) What is Marshalling?

Marshalling

- Passing signature of a function, parameters and return values to a different process (potentially on a different machine)
- Usually implemented by conversion of structured data into a dedicated format, which can be transferred to other processes or systems (*serialization / deserialization*)
- Reverse process: *demarshalling* (or *unmarshalling*)

b) The *Marshalling.zip* project implements a scenario, in which a client requests a server with printing functionality to format and print given data. The connection is established by middleware, which in addition takes care of transfer of typed objects.

- Read the code and learn how the connection gets established.
- Implement the methods *Marshall* and *Demarshall*
- Start the two projects *Client* and *Server* and test your implementation (right click on *Solution* in *Solution Explorer* and then *StartUp Projects*)

c) Beside the default printing functionality, the server provides two string operations:

- string concat(string arg1, string arg2, string arg3)
- string substring(string str, string positionIndex)

Assume, there are more of such operations, all only with string parameters. Extend your client towards RPC of these operations. Complete the placeholders marked with TODO. Define a message encoding scheme for RPC calls and use Reflection on the server side to invoke the methods.

VSR

Your feedback on today's session:

mytuc.org/ttbw

Questions?

vsr-sse@informatik.tu-chemnitz.de

VSR.*Informatik*.TU-Chemnitz.*de*