

Distributed Applications – Session 7

Lecturer Mouhsen Ibrahim

Contents

- Client Library
- Example 1
- Fault Tolerance
- Example 2
- Scaling
- Example 3

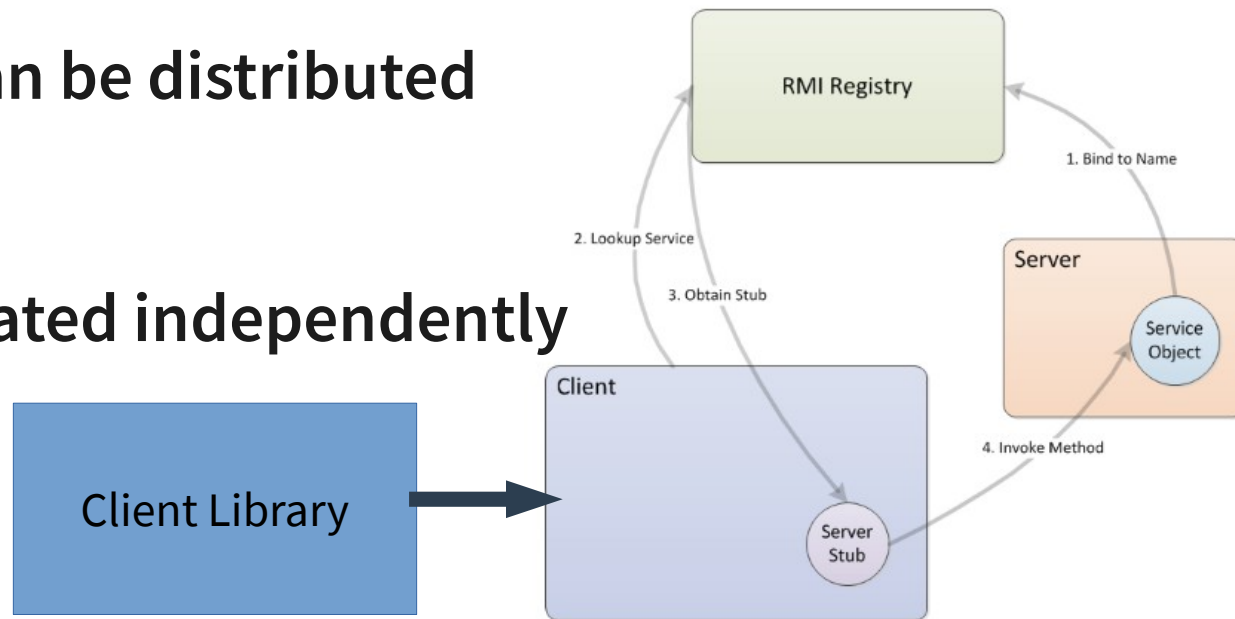
Client Library

A set of **classes** and **methods** to consume **RMI Services**

It **abstracts** remote invocation nature How Java RMI Works

The library can be distributed
as a **jar** file

It can be updated independently



Example 1

Create an RMI service that exposes these two methods

register(String username, String password)

Store username and password in files

login(String username, String password)

Return True or False if credentials are True

Develop a **client library** for these two methods

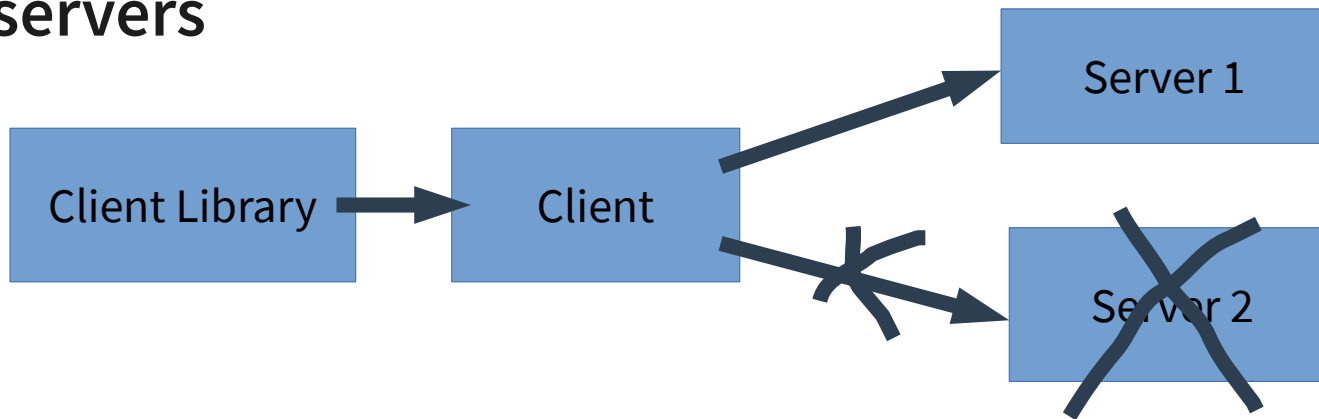
Fault Tolerance

A distributed system must not fail if one component fails

Handling failure can happen in **Client Side** or **Server Side**

In client we use **Client Library** to hide server failure

In server we use a **load balancer** to route requests to healthy servers



Example 2

Based on previous example create **two servers** to handle requests

Update the **client library** to hide **server problems** from the client

Simulate **one server error** then check if client still works

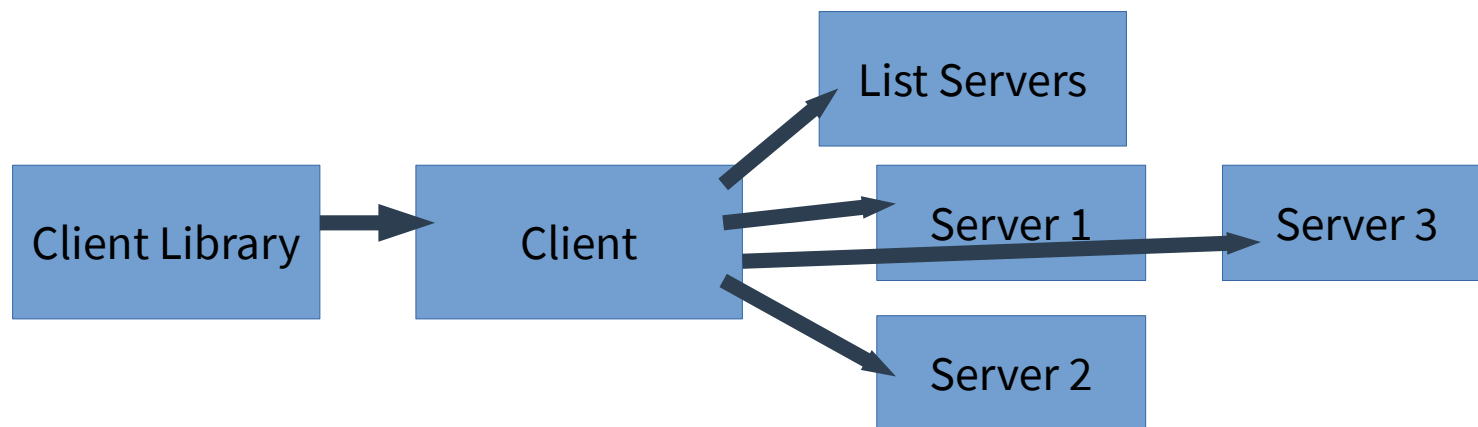
Scaling

Scaling is the process of **adding new servers dynamically** to the system

There must be a way to identify the new servers to clients

Load balancer → add new server to its configuration

Client library → create a service for listing available servers



Example 3

Create a **Server List** service to be used as a catalog for available servers

Add **new servers** to the previous example and **register** them in new service

Make the **client library** use new servers when needed

Good Luck