Socket Programming in Java

CS 640 Introduction to Computer Networks

Course Instructor - Ming Liu (*mgliu@cs.wisc.edu)*
Teaching Assistant - Partho Sarthi (*sarthi@wisc.edu)*

Office Hours: #3209

WF 1:00PM - 2:00PM

or by appointment

Agenda

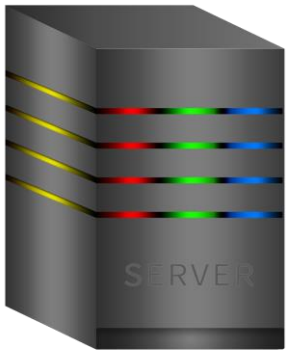
1. Introduction
2. What is a Socket?
3. Methods in Socket API
    1. Client Side
    2. Server Side
4. Further exploration

Introduction

How does one computer send information to another computer?

Request

Response



Client 1

Server

IP Address – Unique Id on the network

Port – Unique Id number on your computer linked to your program

Packet – Data sent from one computer to another

Introduction

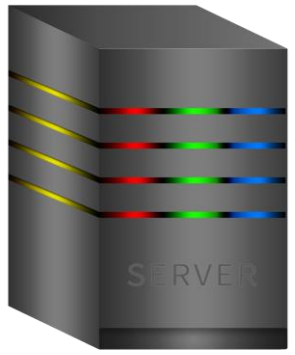
How does one computer send information to another computer?

IP: 192.168.10.4
Port: 25600

Client 1

Request

Response



IP: 192.168.10.1
Port: 25500

Server

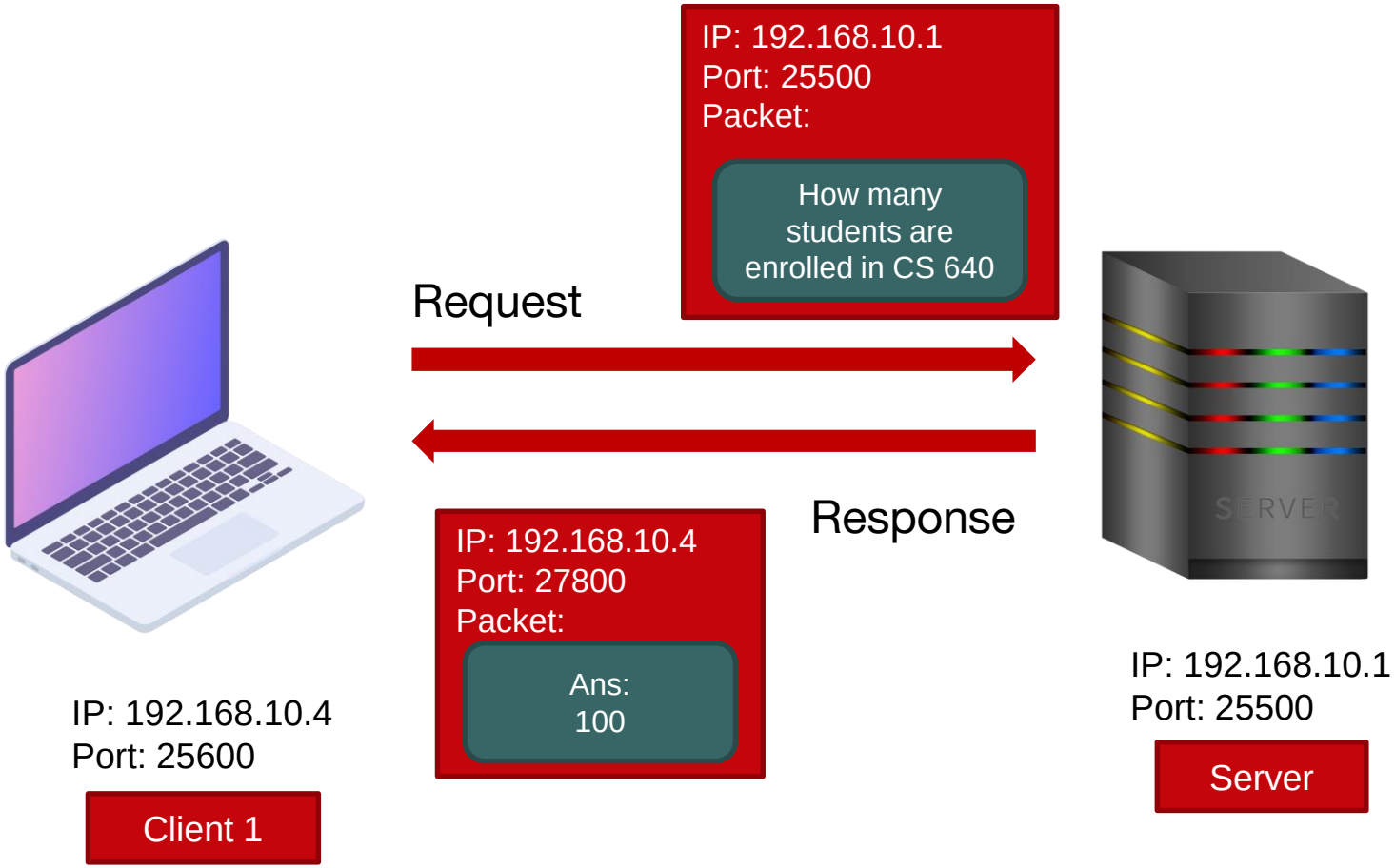
IP Address – Unique Id on the network

Port – Unique Id number on your computer linked to your program

Packet – Data sent from one computer to another

Introduction

How does one computer send information to another computer?



IP Address – Unique Id on the network

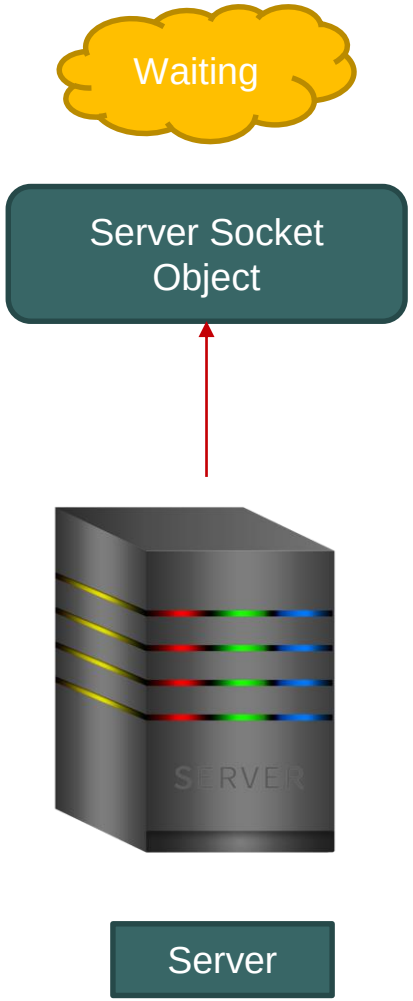
Port – Unique Id number on your computer linked to your program

Packet – Data sent from one computer to another

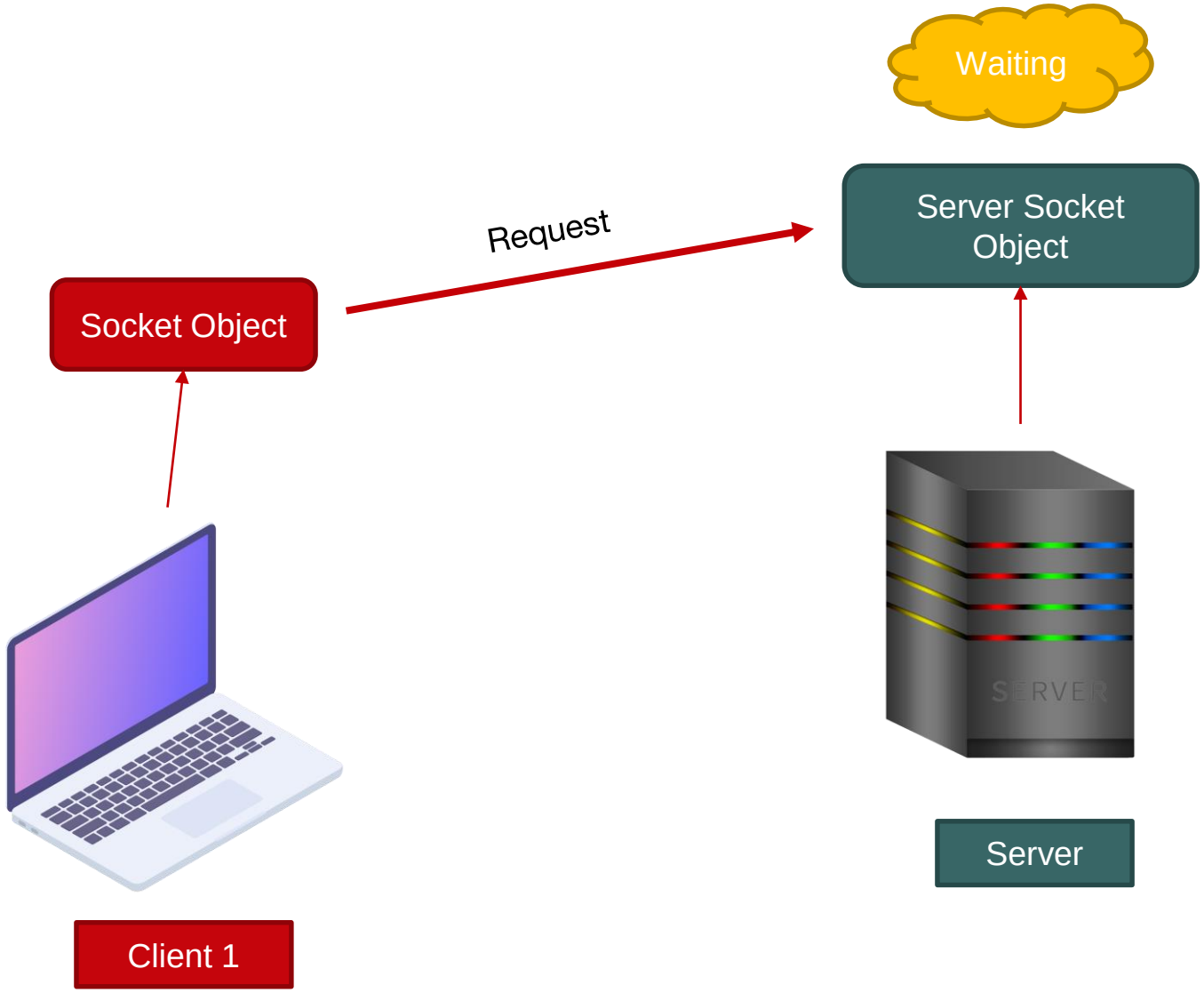
What is a Socket?

- Sockets allow communication between two different processes on the same or different machines.
- A socket is bound to a port number so that the transport layer can identify the application that data is destined to be sent to.

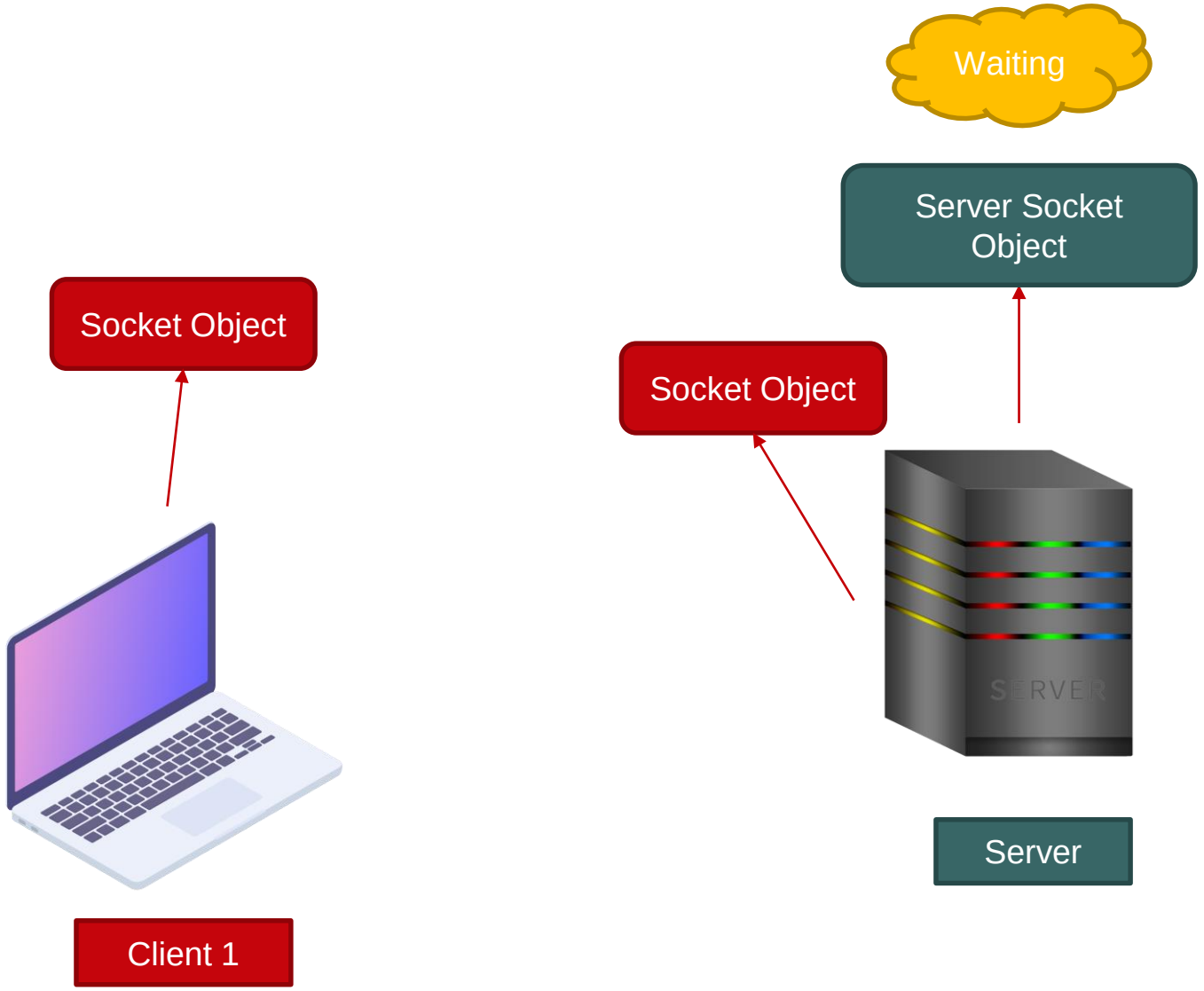
What is a Socket?



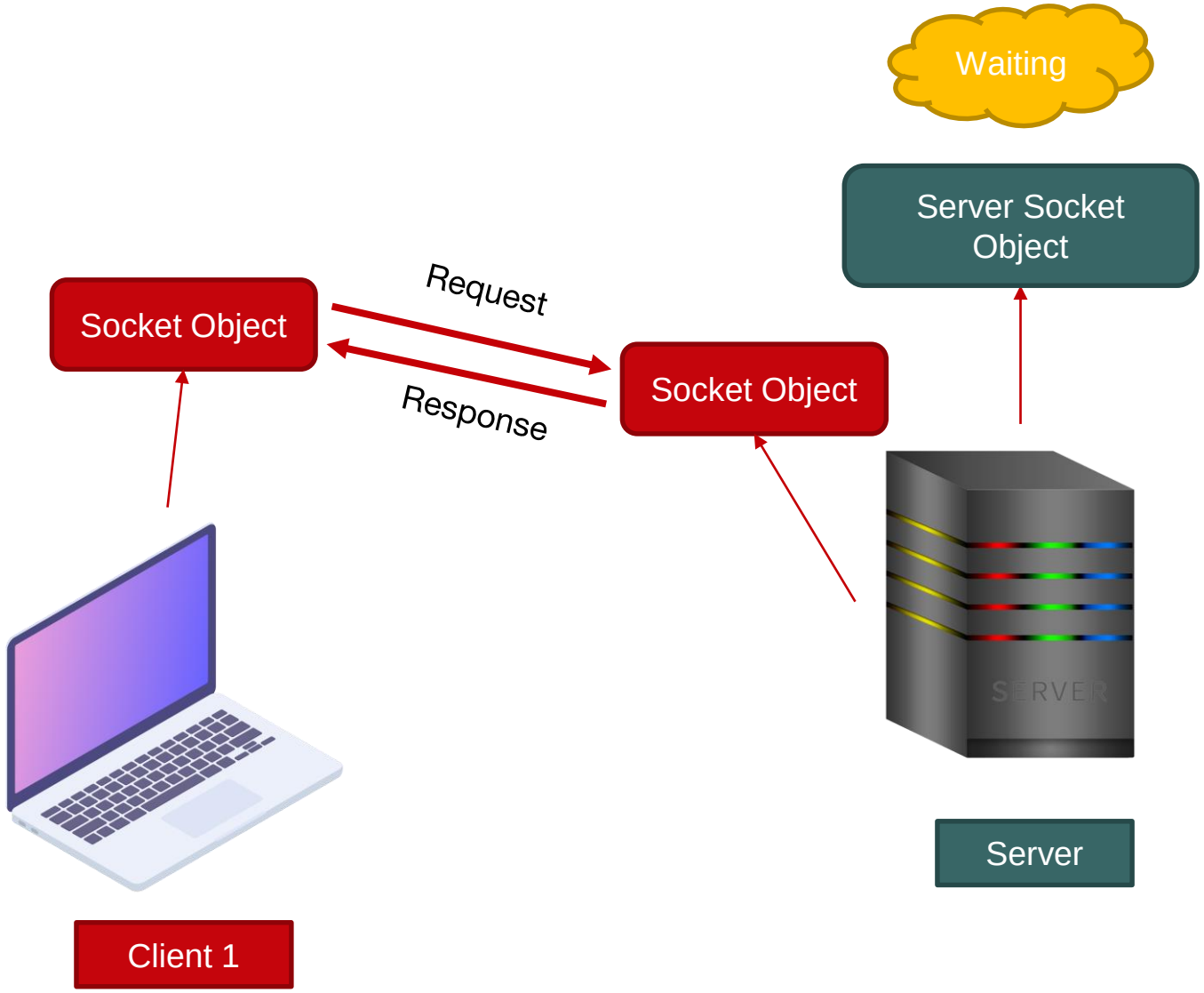
What is a Socket?



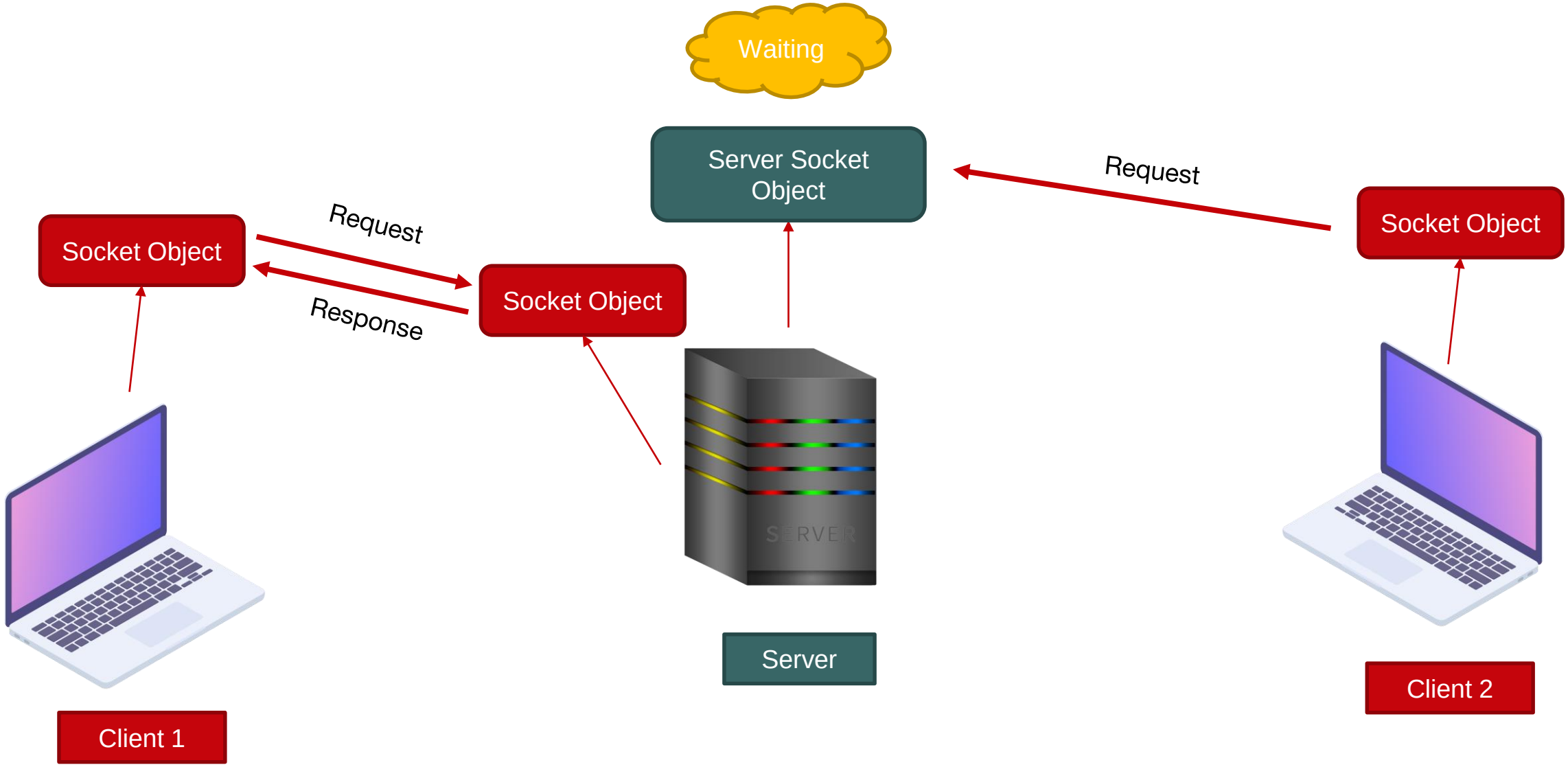
What is a Socket?



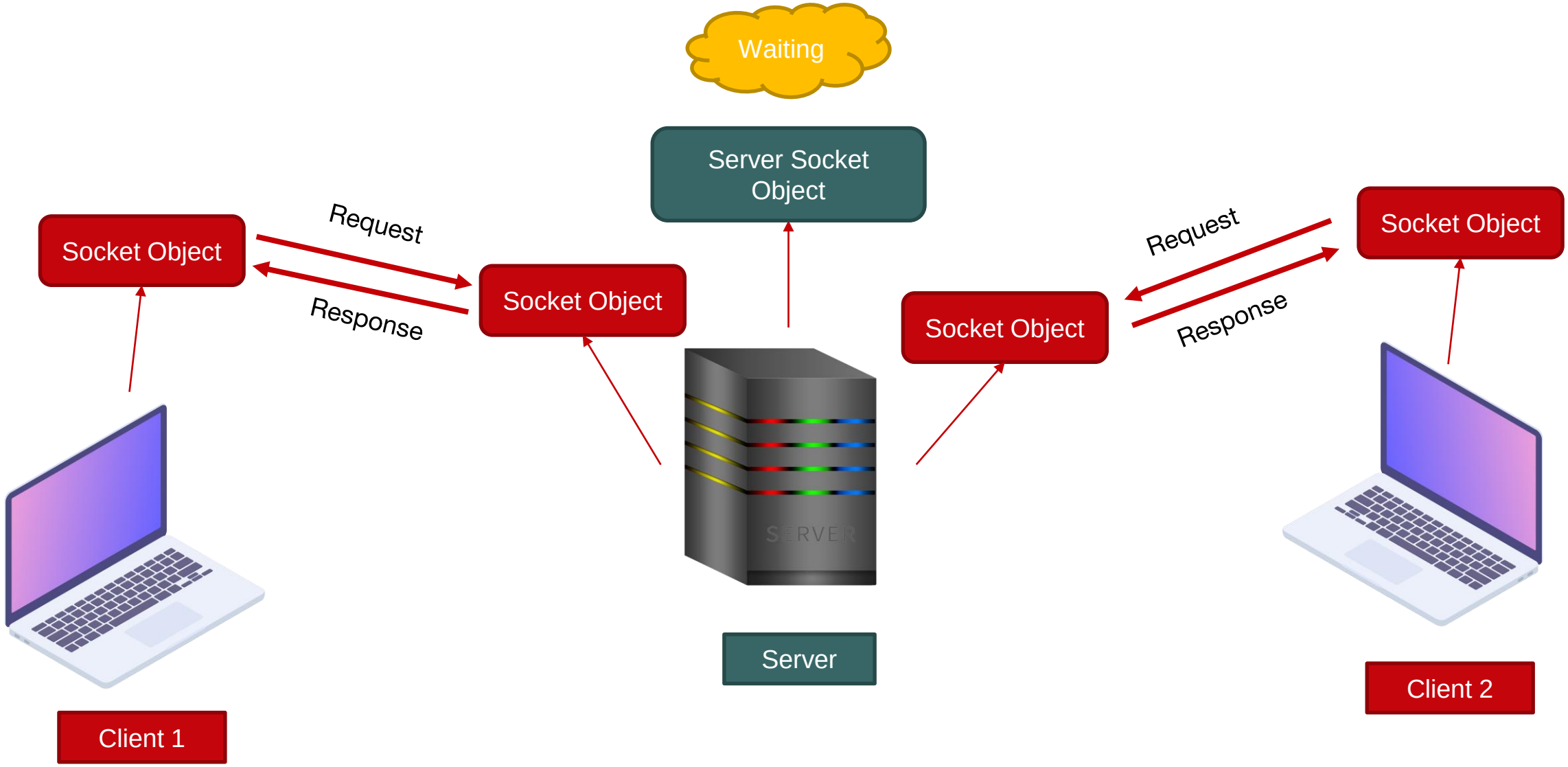
What is a Socket?



What is a Socket?



What is a Socket?



Methods in Socket API

Client Side –

1. Create a Socket object:

```
Socket socket = new Socket(address, port);
```

2. Create an Output Writer:

```
PrintWriter out = new PrintWriter(socket.getOutputStream(), true);
```

3. Write the message:

```
out.println(msg);
```

4. Close the connection:

```
out.close();
socket.close();
```

Methods in Socket API

Server Side –

1. Create a Server Socket object:

```
ServerSocket server = new ServerSocket(port);
```

2. Wait for Client and then create a Socket object:

```
Socket socket = server.accept();
```

3. Create an Input Reader:

```
in = new BufferedReader(InputStreamReader(socket.getInputStream()));
```

3. Read and Display the message:

```
line = in.readLine();
```

4. Close the connection:

```
in.close();
socket.close();
server.close();
```

Watch the Walkthrough Video

Further exploration

- Keep the server running till a “quit” message is received.
- Multiple clients – synchronous and asynchronous communication.

(Hint – see Threads in Java)

Reading :

- Lesson: All About Sockets (The Java™ Tutorials) (oracle.com)
- Socket Programming in Java

Thank You

Use Piazza for any doubts