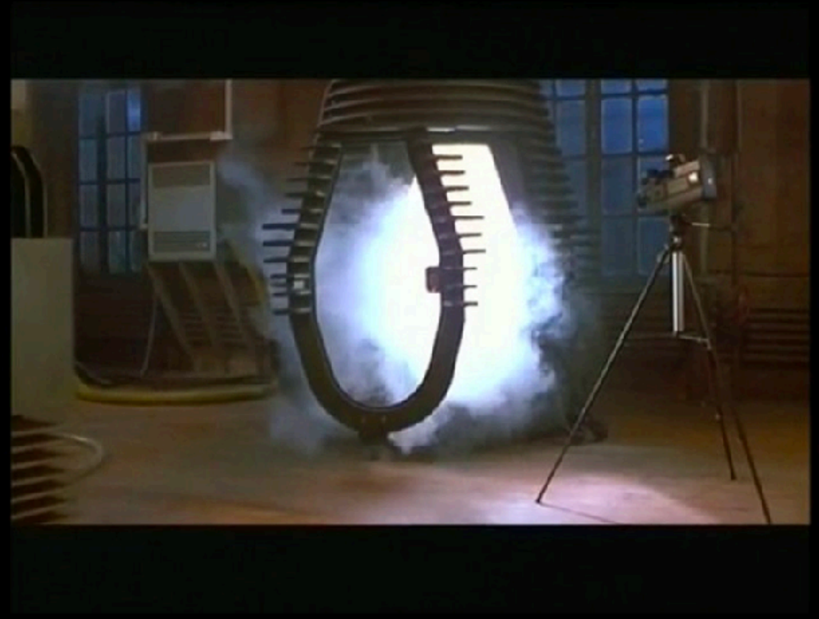
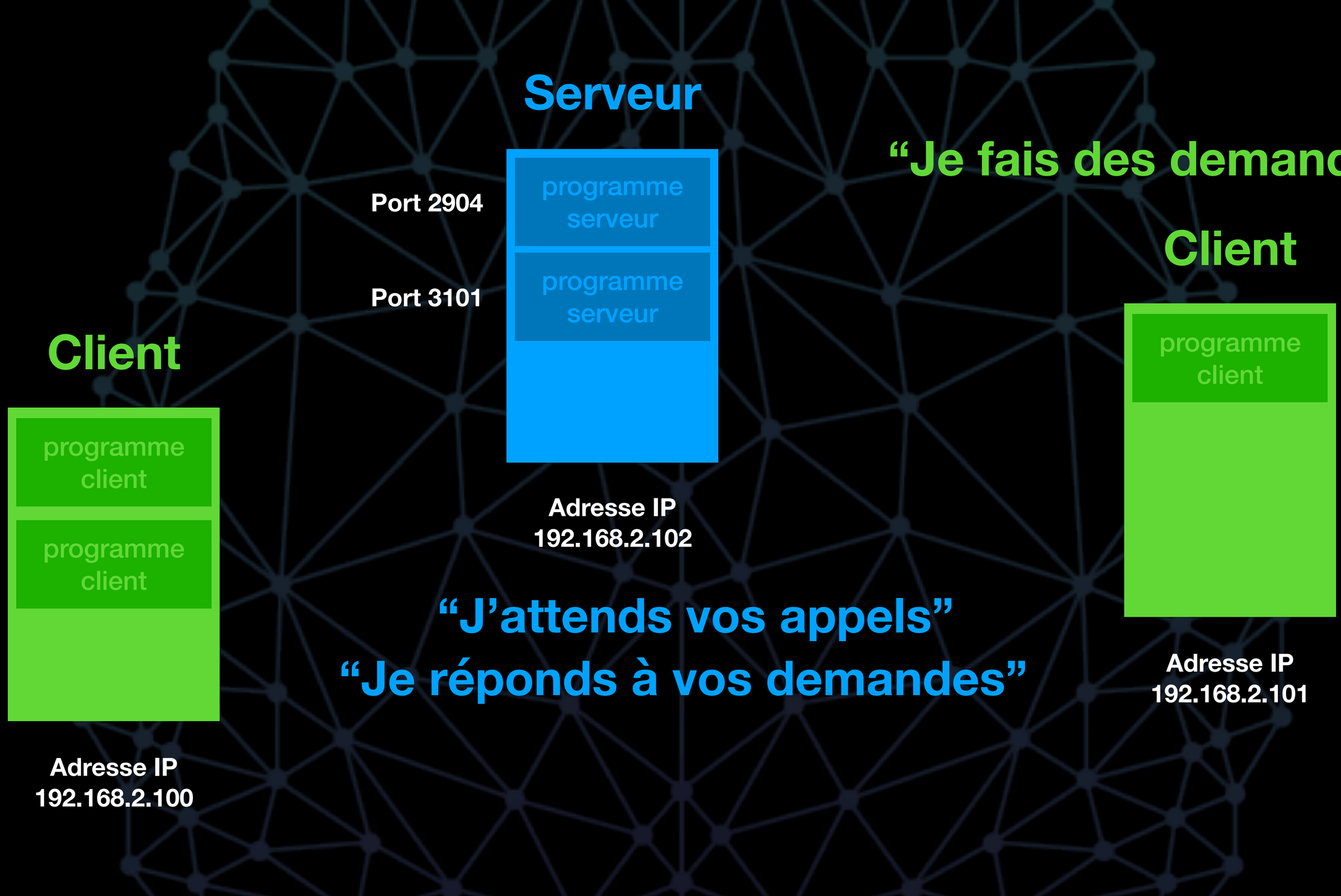


Expérience de Téléportation


Serveur
Port 2904
programme serveur
Port 3101
programme serveur
Adresse IP
192.168.2.102
“Je fais des demandes”
Client
programme client
Adresse IP
192.168.2.101
Client
programme client
programme client
Adresse IP
192.168.2.100
“J’attends vos appels”
“Je réponds à vos demandes”

Client

programme client

```
nc 172.16.29.111 3101
```

Serveur

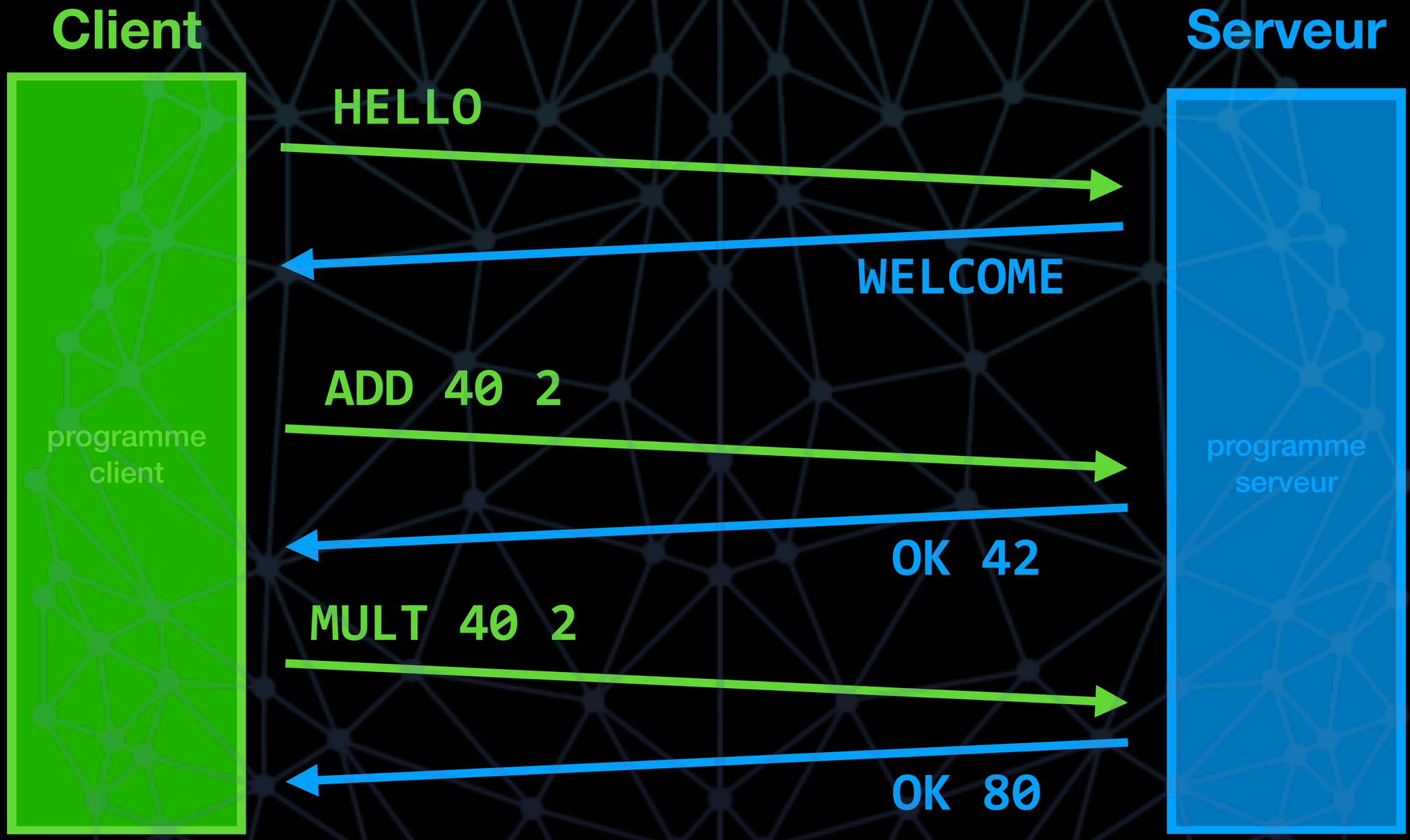
programme serveur

```
nc -l 3101
```

“l pour listen”

Quelle est l’adresse IP du serveur?

```
ipconfig getifaddr en0
> 172.16.29.111
```


Client
Serveur
HELLO
WELCOME
ADD 40 2
OK 42
MULT 40 2
OK 80
programme client
programme serveur

```java
package ch.heigvd;
import java.io.*;
import java.net.*;
public class Server {

    public void waitForIncomingClient() throws IOException {

        ServerSocket receptionistSocket = new ServerSocket(2205);
        Socket workerSocket = receptionistSocket.accept();
        BufferedReader in = new BufferedReader(new InputStreamReader(workerSocket.getInputStream()));
        PrintWriter out = new PrintWriter(new OutputStreamWriter(workerSocket.getOutputStream()));

        String message = in.readLine();
        while (message != null) {
            String[] tokens = message.split(" ");
            String operation = tokens[0];
            int operand1 = Integer.parseInt(tokens[1]);
            int operand2 = Integer.parseInt(tokens[2]);
            if (operation.equals("ADD")) {
                out.println("OK " + (operand1 + operand2));
            } else if (operation.equals("MULT")) {
                out.println("OK " + (operand1 * operand2));
            }
            out.flush();
            message = in.readLine();
        }
        in.close();
        out.close();
        workerSocket.close();
        receptionistSocket.close();
    }
    public static void main( String[] args ) throws IOException
    {
        Server server = new Server();
        while (true) {
            server.waitForIncomingClient();
        }
    }
}
```

```
package ch.heigvd;
import java.io.*;
import java.net.*;
public class Server {

    public void waitForIncomingClient() throws IOException {

        ServerSocket receptionistSocket = new ServerSocket(2205);
        Socket workerSocket = receptionistSocket.accept();
        BufferedReader in = new BufferedReader(new InputStreamReader(workerSocket.getInputStream()));
        PrintWriter out = new PrintWriter(new OutputStreamWriter(workerSocket.getOutputStream()));

        String message = in.readLine();
        while (message != null) {
            String[] tokens = message.split(" ");
            String operation = tokens[0];
            int operand1 = Integer.parseInt(tokens[1]);
            int operand2 = Integer.parseInt(tokens[2]);
            if (operation.equals("ADD")) {
                out.println("OK " + (operand1 + operand2));
            } else if (operation.equals("MULT")) {
                out.println("OK " + (operand1 * operand2));
            }
            out.flush();
            message = in.readLine();
        }
        in.close();
        out.close();
        workerSocket.close();
        receptionistSocket.close();
    }
    public static void main( String[] args ) throws IOException
    {
        Server server = new Server();
        while (true) {
            server.waitForIncomingClient();
        }
    }
}
```

“Tant que des clients arrivent, je m’occupe d’eux”

```
package ch.heigvd;
import java.io.*;
import java.net.*;
public class Server {

    public void waitForIncomingClient() throws IOException {

        ServerSocket receptionistSocket = new ServerSocket(2205);
        Socket workerSocket = receptionistSocket.accept();
        BufferedReader in = new BufferedReader(new InputStreamReader(workerSocket.getInputStream()));
        PrintWriter out = new PrintWriter(new OutputStreamWriter(workerSocket.getOutputStream()));

        String message = in.readLine();
        while (message != null) {
            String[] tokens = message.split(" ");
            String operation = tokens[0];
            int operand1 = Integer.parseInt(tokens[1]);
            int operand2 = Integer.parseInt(tokens[2]);
            if (operation.equals("ADD")) {
                out.println("OK " + (operand1 + operand2));
            } else if (operation.equals("MULT")) {
                out.println("OK " + (operand1 * operand2));
            }
            out.flush();
            message = in.readLine();
        }
        in.close();
        out.close();
        workerSocket.close();
        receptionistSocket.close();
    }
    public static void main( String[] args ) throws IOException
    {
        Server server = new Server();
        while (true) {
            server.waitForIncomingClient();
        }
    }
}
```

“J’écoute sur le port 2205 et je m’arrête jusqu’à ce qu’un client appelle.”

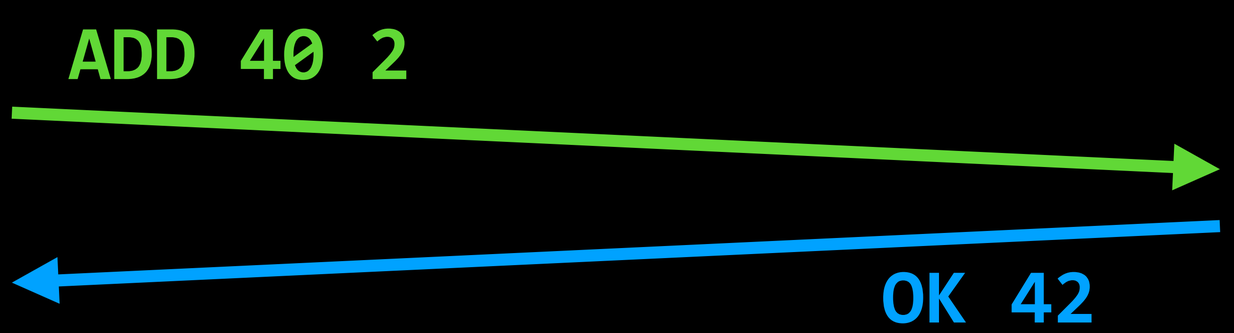
```java
package ch.heigvd;
import java.io.*;
import java.net.*;
public class Server {

    public void waitForIncomingClient() throws IOException {

        ServerSocket receptionistSocket = new ServerSocket(2205);
        Socket workerSocket = receptionistSocket.accept();
        BufferedReader in = new BufferedReader(new InputStreamReader(workerSocket.getInputStream()));
        PrintWriter out = new PrintWriter(new OutputStreamWriter(workerSocket.getOutputStream()));

        String message = in.readLine();
        while (message != null) {
            String[] tokens = message.split(" ");
            String operation = tokens[0];
            int operand1 = Integer.parseInt(tokens[1]);
            int operand2 = Integer.parseInt(tokens[2]);
            if (operation.equals("ADD")) {
                out.println("OK " + (operand1 + operand2));
            } else if (operation.equals("MULT")) {
                out.println("OK " + (operand1 * operand2));
            }
            out.flush();
            message = in.readLine();
        }
        in.close();
        out.close();
        workerSocket.close();
        receptionistSocket.close();
    }
    public static void main( String[] args ) throws IOException
    {
        Server server = new Server();
        while (true) {
            server.waitForIncomingClient();
        }
    }
}
```

“Un ‘tuyau virtuel’ me connecte au client. Avec ‘out’, je peux lui envoyer des caractères. Avec ‘in’, je peux lire ce qu’il m’envoie.

```java
package ch.heigvd;
import java.io.*;
import java.net.*;
public class Server {

    public void waitForIncomingClient() throws IOException {

        ServerSocket receptionistSocket = new ServerSocket(2205);
        Socket workerSocket = receptionistSocket.accept();
        BufferedReader in = new BufferedReader(new InputStreamReader(workerSocket.getInputStream()));
        PrintWriter out = new PrintWriter(new OutputStreamWriter(workerSocket.getOutputStream()));

        String message = in.readLine();
        while (message != null) {
            String[] tokens = message.split(" ");
            String operation = tokens[0];
            int operand1 = Integer.parseInt(tokens[1]);
            int operand2 = Integer.parseInt(tokens[2]);
            if (operation.equals("ADD")) {
                out.println("OK " + (operand1 + operand2));
            } else if (operation.equals("MULT")) {
                out.println("OK " + (operand1 * operand2));
            }
            out.flush();
            message = in.readLine();
        }
        in.close();
        out.close();
        workerSocket.close();
        receptionistSocket.close();
    }
    public static void main( String[] args ) throws IOException
    {
        Server server = new Server();
        while (true) {
            server.waitForIncomingClient();
        }
    }
}
```

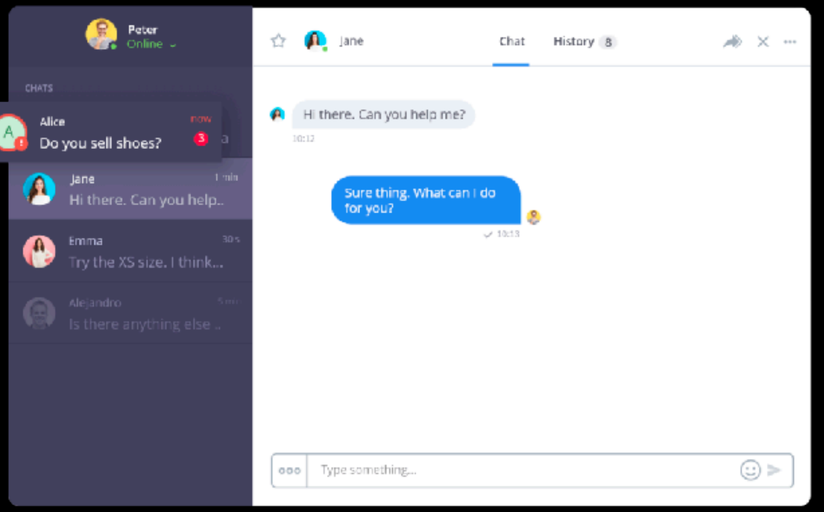
“Je lis ce que m’envoie le client, ligne par ligne. Jusqu’à ce qu’il ait raccroché.”

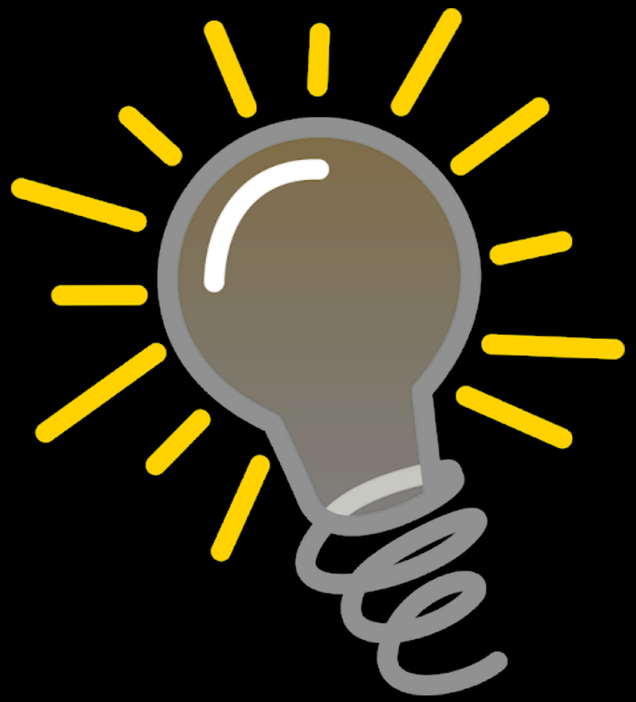
```java
package ch.heigvd;
import java.io.*;
import java.net.*;
public class Server {

    public void waitForIncomingClient() throws IOException {

        ServerSocket receptionistSocket = new ServerSocket(2205);
        Socket workerSocket = receptionistSocket.accept();
        BufferedReader in = new BufferedReader(new InputStreamReader(workerSocket.getInputStream()));
        PrintWriter out = new PrintWriter(new OutputStreamWriter(workerSocket.getOutputStream()));

        String message = in.readLine();
        while (message != null) {
            String[] tokens = message.split(" ");
            String operation = tokens[0];
            int operand1 = Integer.parseInt(tokens[1]);
            int operand2 = Integer.parseInt(tokens[2]);
            if (operation.equals("ADD")) {
                out.println("OK " + (operand1 + operand2));
            } else if (operation.equals("MULT")) {
                out.println("OK " + (operand1 * operand2));
            }
            out.flush();
            message = in.readLine();
        }
        in.close();
        out.close();
        workerSocket.close();
        receptionistSocket.close();
    }
    public static void main( String[] args ) throws IOException
    {
        Server server = new Server();
        while (true) {
            server.waitForIncomingClient();
        }
    }
}
```



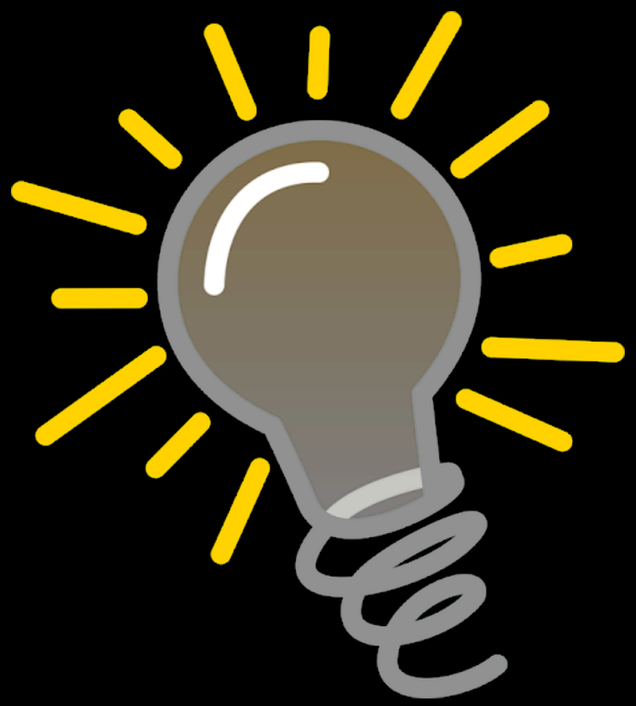
“Voici le coeur du protocole ‘calculatrice’:

- Je découpe chaque ligne en 3 morceaux.
- Si le premier morceau est ADD, alors j’additionne le 2ème et le 3ème morceau
- Si le premier morceau est MULT, alors je multiplie les 2ème et 3ème morceau
- Dans tout les cas, j’envoie une ligne au client: ‘OK’ suivi d’un espace, suivi du résultat.


Peter
Online
CHATS
Alice
Do you sell shoes?
Jane
Hi there. Can you help..
Emma
Try the XS size. I think...
Alejandro
Is there anything else ..
Jane
Chat
History 8
Hi there. Can you help me?
Sure thing. What can I do for you?
Type something...




Product Room, New York
Boardroom, London



Une **application** est un **système**:

un **ensemble de composants** qui interagissent les uns avec les autres en **échangeant des messages sur le Réseau**, en respectant un protocole de communication.