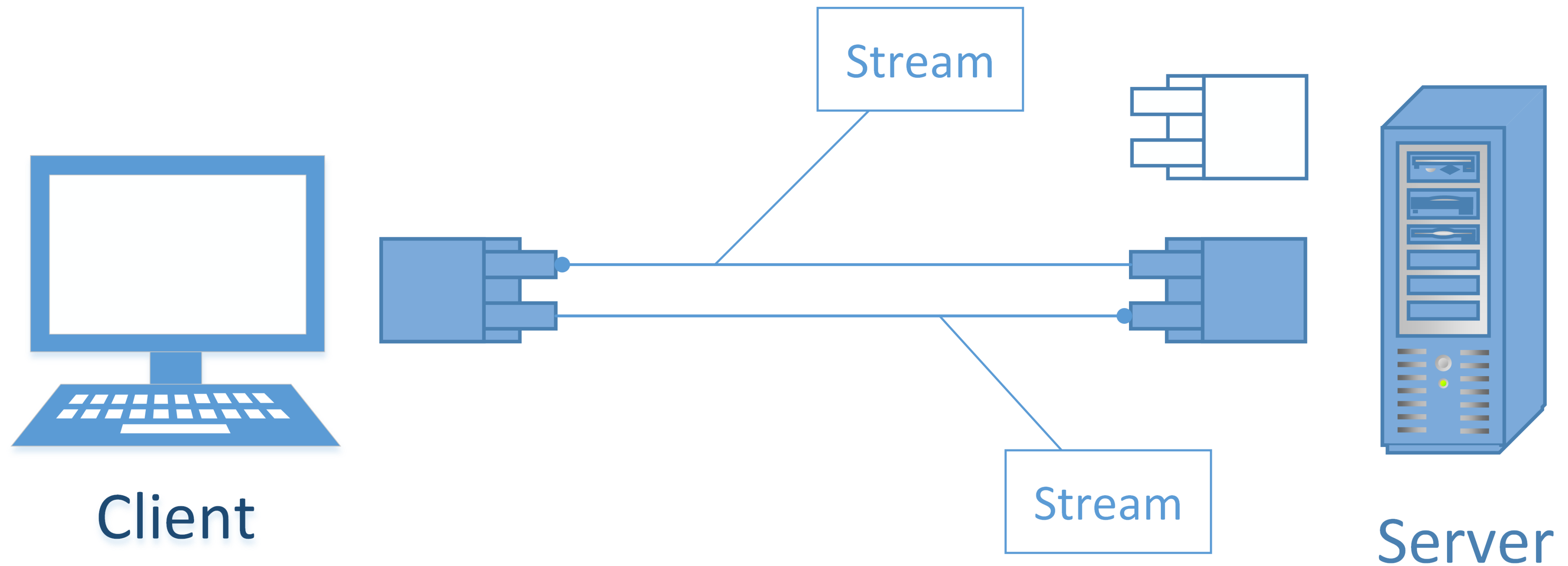


Programming Using Java

Session 11: Socket-Based Development



Sockets

“...allows communication between programs...”

Server Socket

create server socket and
wait for connection

```
public class MyServer {
```

```
    public static void main(String[] args) {  
        try  
        (  
            
```

```
                ServerSocket server = new ServerSocket(port);  
                Socket client = server.accept();
```

```
                PrintWriter out =  
                    new PrintWriter(client.getOutputStream(), true);  
  
                BufferedReader in =  
                    new BufferedReader(new InputStreamReader(  
                        client.getInputStream()));
```

```
            ) {... process client message ...} }}
```

access streams connected
to sockets

1. create a ServerSocket
2. wait for a connection
3. get socket i/o streams
4. custom processing
5. close the connection

Socket

```
public class MyClient {  
    public static void main(String[] args) {  
        try  
        (  
            Socket s = new Socket("host", port);  
  
            PrintWriter out =  
                new PrintWriter(s.getOutputStream(), true);  
  
            BufferedReader in =  
                new BufferedReader(new InputStreamReader(  
                    s.getInputStream() ));  
  
        ) {... process server message ...} }}
```

establish connection with
host server waiting on the
specified port

access streams connected
to sockets

1. create a socket
2. get socket i/o streams
3. custom processing
4. close the connection

```
public static void main(String[] args) {  
    ...  
    try (ServerSocket serverSocket = new ServerSocket(portNumber)) {  
        while (true) {  
            // spawn off thread to process work  
            new ServerThread(serverSocket.accept()).start();  
        }  
    } catch (IOException e) { handle exception }  
}  
  
class ServerThread extends Thread {  
    private Socket socket = null; ServerThread(Socket s) {...}  
    private Socket socket = null;  
    public void run() { ... process request ... }  
}
```

new thread is created, request is processed
in thread

client request processing
is abstracted into
a different
class

Multi-Threaded Server

“...a thread is spawned for each request...”