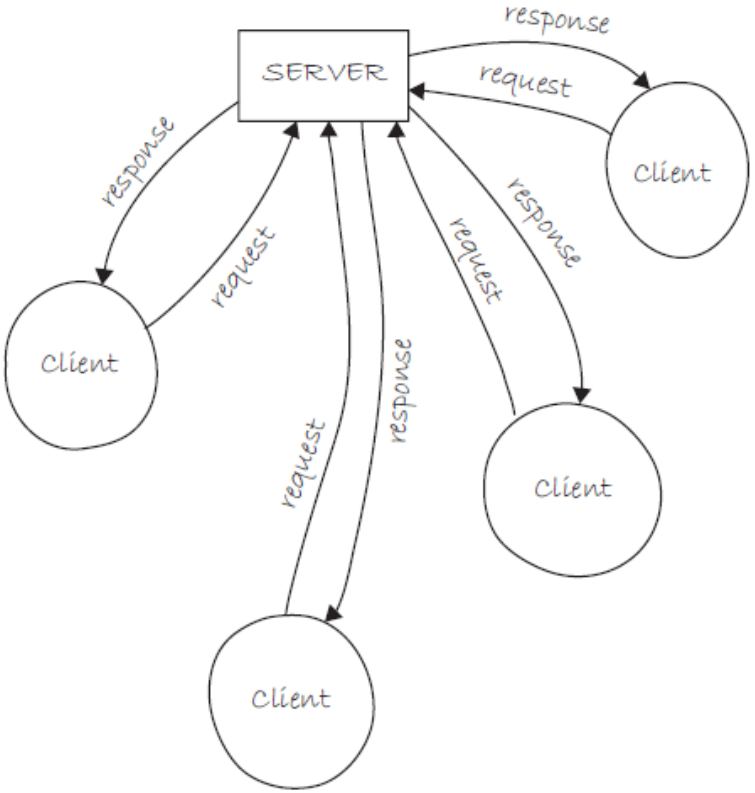
Network

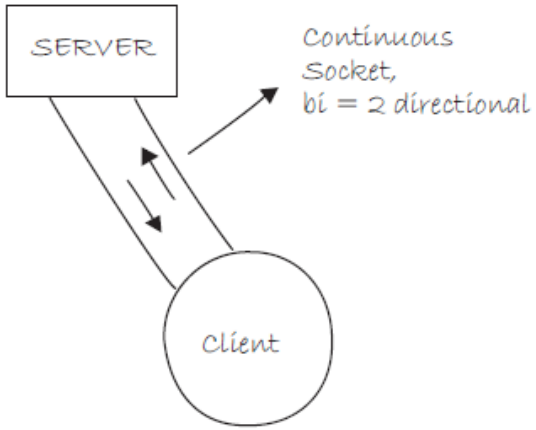
Technische Universiteit
Eindhoven
University of Technology

Where innovation starts

Asynchronous vs. synchronous

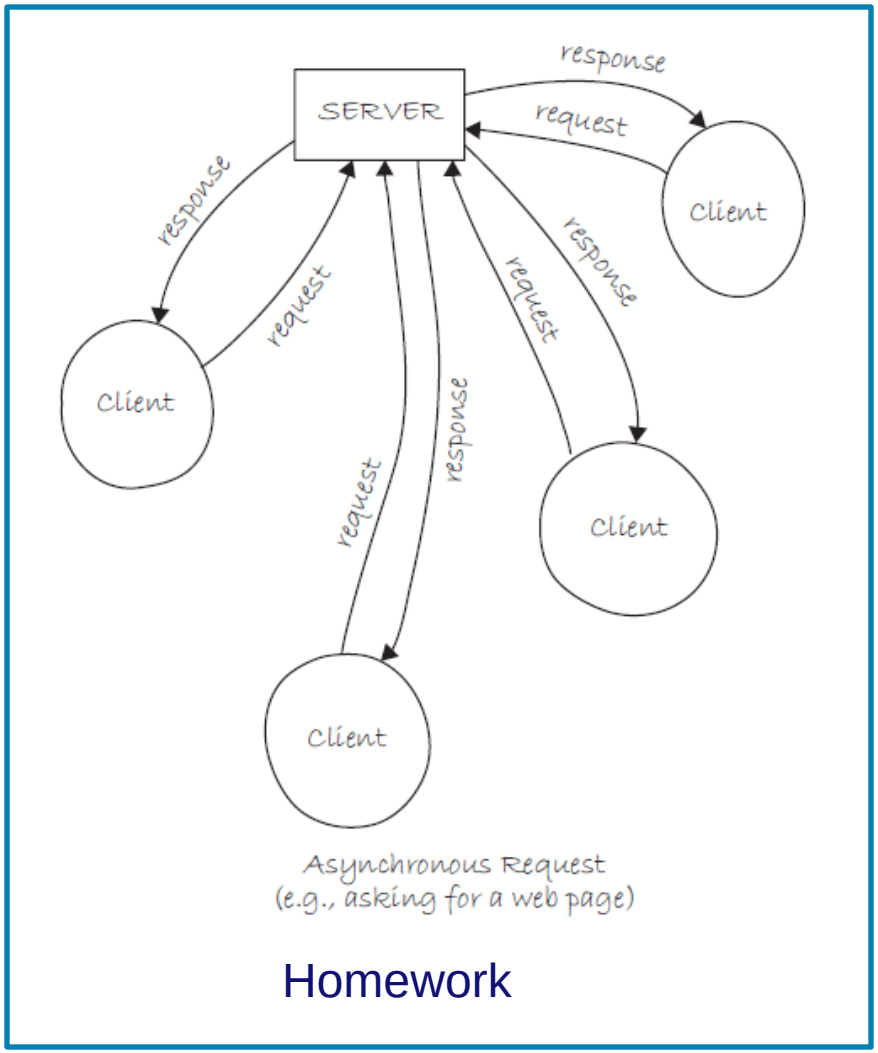


Asynchronous Request
(e.g., asking for a web page)

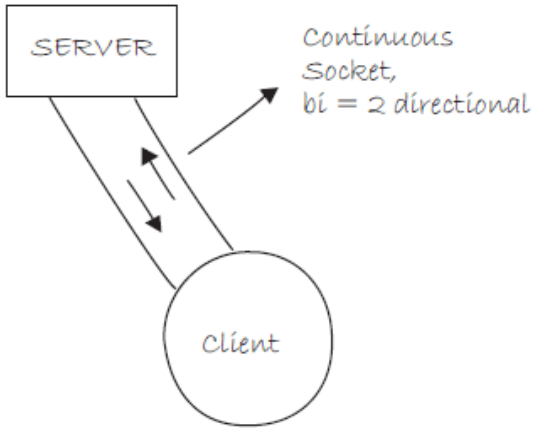


Socket Connection
(e.g., chat)

Asynchronous vs. synchronous



Homework

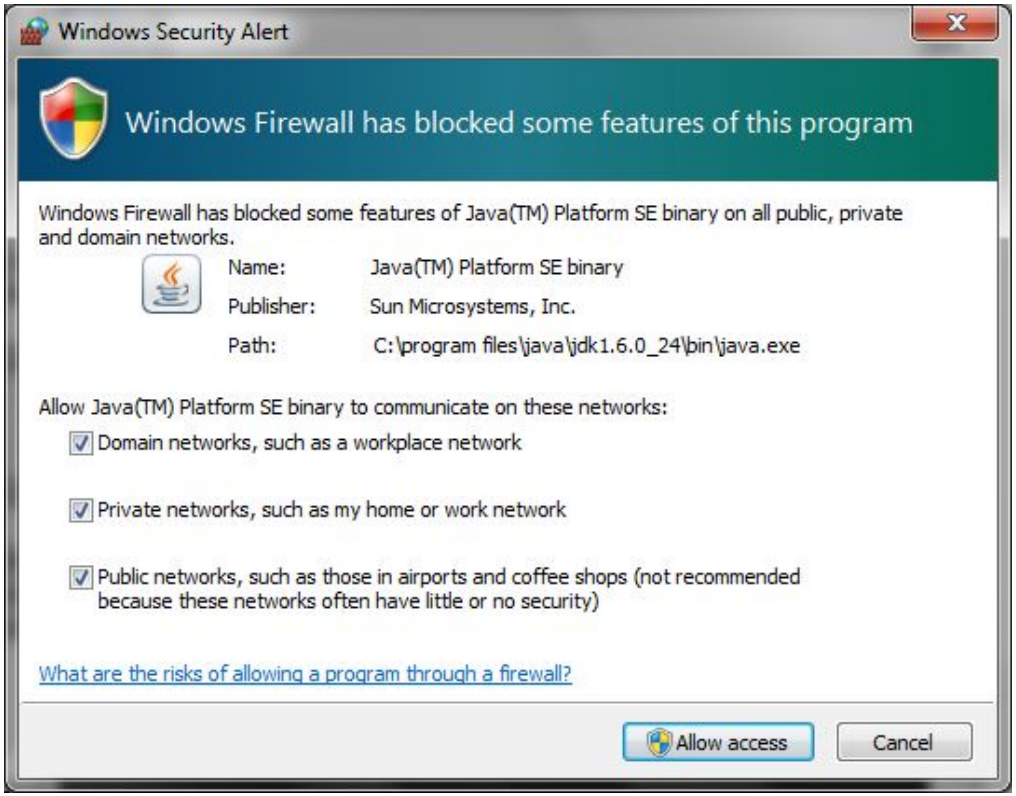


Socket Connection
(e.g., chat)

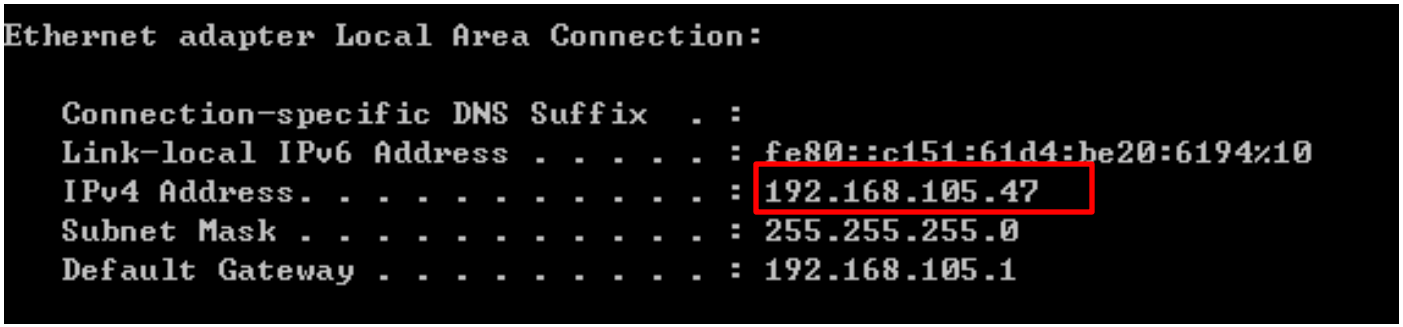
Client/Server

- **Help>Reference: Libraries : Network : Server**
- **Help>Reference: Libraries : Network : Client**

- **Before we continue…**



- **Find your IP**
- **On windows:**
  - **WIN+R, cmd, ipconfig**
- **On Mac:**
  - **Applications menu>Utilities>Terminal, ifconfig**



Client/Server

```
import processing.net.*;
Server myServer;
int val = 0;

void setup() {
  size(200, 200);
  // Starts a myServer on port 5204
  myServer = new Server(this, 5204);
}

void draw() {
  val = (val + 1)%255;
  background(val);
  myServer.write(val);
}
```

```
import processing.net.*;
Client myClient;
int dataIn;

void setup() {
  size(200, 200);
  // Connect to the local machine at port 5204.
  // This example will not run if you haven't
  // previously started a server on this port
  myClient = new Client(this, "127.0.0.1", 5204);
}

void draw() {
  if (myClient.available() > 0) {
    dataIn = myClient.read();
  }
  background(dataIn);
}
```

Client/Server

- **Now try to change the server code:**
  - **Reacts to mouse clicks**
  - **Position of the mouse changes the background color**
  - **Send the background color to clients**

Client/Server

```
import processing.net.*;
Server myServer;
int val = 0;

void setup() {
  size(200, 200);
  // Starts a myServer on port 5204
  myServer = new Server(this, 5204);
}

void draw() {
  background(val);
}

void mousePressed(){
  val = mouseY;
  myServer.write(val);
}
```

```
import processing.net.*;
Client myClient;
int dataIn;

void setup() {
  size(200, 200);
  // Connect to the local machine at port 5204.
  // This example will not run if you haven't
  // previously started a server on this port
  myClient = new Client(this, "127.0.0.1", 5204);
}

void draw() {
  if (myClient.available() > 0) {
    dataIn = myClient.read();
  }
  background(dataIn);
}
```

Client/Server

- Now try out with your neighbor Ms/r Nice:
  - Nice runs the server.
  - You replace “127.0.0.1” in your client with the IP address of Ms/r Nice’s computer
  - You run the client.
  - Try the opposite.
  - Later you can always try the same.

Client/Server

- Now let's try the opposite
- Now try to change the client code:
  - Reacts to mouse clicks
  - Position of the mouse changes the background color
  - Send the background color to the server

Client/Server

```
import processing.net.*;
Server myServer;
Client c;
int val = 0;

void setup() {
  size(200, 200);
  // Starts a myServer on port 5204
  myServer = new Server(this, 5204);
}

void draw() {
  c = myServer.available();
  if(c != null){
     val = c.read();
     background(val);
  }
}
```

```
import processing.net.*;
Client myClient;
int val;

void setup() {
  size(200, 200);
  // Connect to the local machine at port 5204.
  // This example will not run if you haven't
  // previously started a server on this port
  myClient = new Client(this, "127.0.0.1", 5204);
}

void draw() {
  background(val);
}

void mousePressed(){
  val = mouseY;
  myClient.write(val);
}
```

Client/Server

- **Now let's synchronize the background of all clients and the server.**

Client/Server

```
import processing.net.*;
Server myServer;
Client c;
int val = 0;

void setup() {
  size(200, 200);
  // Starts a myServer on port 5204
  myServer = new Server(this, 5204);
}

void draw() {
  c = myServer.available();
  if(c != null){
     val = c.read();
     background(val);
     myServer.write(val);
  }
}

void mousePressed(){
  val = mouseY;
  background(val);
  myServer.write(val);
}
```

```
import processing.net.*;
Client myClient;
int val;

void setup() {
  size(200, 200);
  // Connect to the local machine at port 5204.
  // This example will not run if you haven't
  // previously started a server on this port
  myClient = new Client(this, "127.0.0.1", 5204);
}

void draw() {
  if(myClient.available()>0){
    val = myClient.read();
    background(val);
  }
}

void mousePressed(){
  val = mouseY;
  myClient.write(val);
}
```

Client/Server

- A fancier example:
- File>Examples>Libraries>Network:
  - SharedCanvasClient
  - SharedCanvasServer

SharedCanvasServer

```
import processing.net.*;

Server s;
Client c;
String input;
int data[];

void setup()
{
  size(450, 255);
  background(204);
  stroke(0);
  frameRate(5); // Slow it down a little
  s = new Server(this, 12345); // Start a simple server on a port
}
```

SharedCanvasServer

```
void draw()
{
  if (mousePressed == true) {
    // Draw our line
    stroke(255);
    line(pmouseX, pmouseY, mouseX, mouseY);
    // Send mouse coords to other person
    s.write(pmouseX + " " + pmouseY + " " + mouseX + " " + mouseY + "\n");
  }
  // Receive data from client
  c = s.available();
  if (c != null) {
    input = c.readString();
    input = input.substring(0, input.indexOf("\n")); // Only up to the newline
    data = int(split(input, ' ')); // Split values into an array
    // Draw line using received coords
    stroke(0);
    line(data[0], data[1], data[2], data[3]);
  }
}
```

SharedCanvasClient

```
import processing.net.*;

Client c;
String input;
int data[];

void setup()
{
  size(450, 255);
  background(204);
  stroke(0);
  frameRate(5); // Slow it down a little
  // Connect to the server's IP address and port
  c = new Client(this, "127.0.0.1", 12345); // Replace with your server's IP and port
}
```

SharedCanvasClient

```
void draw()
{
  if (mousePressed == true) {
    // Draw our line
    stroke(255);
    line(pmouseX, pmouseY, mouseX, mouseY);
    // Send mouse coords to other person
    c.write(pmouseX + " " + pmouseY + " " + mouseX + " " + mouseY + "\n");
  }
  // Receive data from server
  if (c.available() > 0) {
    input = c.readString();
    input = input.substring(0, input.indexOf("\n")); // Only up to the newline
    data = int(split(input, ' ')); // Split values into an array
    // Draw line using received coords
    stroke(0);
    line(data[0], data[1], data[2], data[3]);
  }
}
```

SharedCanvasServer
SharedCanvasClient

- **Let's try to improve the code**

```
input = c.readString();
input = input.substring(0, input.indexOf("\n")); // Only up to the newline
```

SharedCanvasServer
SharedCanvasClient

```
void draw()
{
  if (mousePressed == true) {
    // Draw our line
    stroke(255);
    line(pmouseX, pmouseY, mouseX, mouseY);
    // Send mouse coords to other person
    s.write(pmouseX + " " + pmouseY + " " + mouseX + " " + mouseY + "\n");
  }
  // Receive data from client
  c = s.available();
  if (c != null) {
    input = c.readStringUntil('\n').trim();
    data = int(split(input, ' ')); // Split values into an array
    // Draw line using received coords
    stroke(0);
    line(data[0], data[1], data[2], data[3]);
  }
}
```

```
void draw()
{
  if (mousePressed == true) {
    // Draw our line
    stroke(255);
    line(pmouseX, pmouseY, mouseX, mouseY);
    // Send mouse coords to other person
    c.write(pmouseX + " " + pmouseY + " " + mouseX + " " + mouseY + "\n");
  }
  // Receive data from server
  if (c.available() > 0) {
    input = c.readStringUntil('\n').trim();
    data = int(split(input, ' ')); // Split values into an array
    // Draw line using received coords
    stroke(0);
    line(data[0], data[1], data[2], data[3]);
  }
}
```

```
input = c.readString();
input = input.substring(0, input.indexOf("\n")); // Only up to the newline
```

```
input = c.readStringUntil('\n').trim();
```

SharedCanvasServer
SharedCanvasClient

- **Let's try to Synchronize the server and all clients.**

SharedCanvasServer

```
void draw()
{
  if (mousePressed == true) {
    // Draw our line
    stroke(255);
    line(pmouseX, pmouseY, mouseX, mouseY);
    // Send mouse coords to other person
    s.write(pmouseX + " " + pmouseY + " " + mouseX + " " + mouseY + "\n");
  }
  // Receive data from client
  c = s.available();
  if (c != null) {
    input = c.readStringUntil('\n').trim();
    data = int(split(input, ' ')); // Split values into an array
    // Draw line using received coords
    stroke(0);
    line(data[0], data[1], data[2], data[3]);
    s.write(data[0] + " " + data[1] + " " + data[2] + " " + data[3] + "\n");
  }
}
```

SharedCanvasClient

```
void draw()
{
  if (mousePressed == true) {
    // Draw our line
    stroke(255);
    //line(pmouseX, pmouseY, mouseX, mouseY);
    // Send mouse coords to other person
    c.write(pmouseX + " " + pmouseY + " " + mouseX + " " + mouseY + "\n");
  }
  // Receive data from server
  if (c.available() > 0) {
    input = c.readStringUntil('\n').trim();
    data = int(split(input, ' ')); // Split values into an array
    // Draw line using received coords
    stroke(0);
    line(data[0], data[1], data[2], data[3]);
  }
}
```

SharedCanvasServer
SharedCanvasClient

- Do you see the problem?
- How could we improve it?

SharedCanvasClient

```
void draw()
{
  if (mousePressed == true) {
    // Draw our line
    stroke(255);
    line(pmouseX, pmouseY, mouseX, mouseY);
    // Send mouse coords to other person
    c.write(pmouseX + " " + pmouseY + " " + mouseX + " " + mouseY + "\n");
  }
  // Receive data from server
  if (c.available() > 0) {
    input = c.readStringUntil('\n').trim();
    data = int(split(input, ' ')); // Split values into an array
    // Draw line using received coords
    stroke(0);
    line(data[0], data[1], data[2], data[3]);
  }
}
```

SharedCanvasServer

```
import processing.net.*;

Server s;
Client c;
String input;
int data[];
ArrayList clients;

void setup()
{
  size(450, 255);
  background(204);
  stroke(0);
  frameRate(5); // Slow it down a little
  s = new Server(this, 12345); // Start a simple server on a port
  clients = new ArrayList();
}
```

SharedCanvasServer

```
void serverEvent(Server someServer, Client someClient) {
  clients.add(someClient);
}

void disconnectEvent(Client someClient) {
 clients.remove(someClient);
}
```

SharedCanvasServer

```
void draw()
{
  if (mousePressed == true) {
    // Draw our line
    stroke(255);
    line(pmouseX, pmouseY, mouseX, mouseY);
    // Send mouse coords to other person
    s.write(pmouseX + " " + pmouseY + " " + mouseX + " " + mouseY + "\n");
  }
  // Receive data from client
  c = s.available();
  if (c != null) {
    input = c.readStringUntil('\n').trim();
    data = int(split(input, ' ')); // Split values into an array
    // Draw line using received coords
    stroke(0);
    line(data[0], data[1], data[2], data[3]);
    for (int i = 0; i < clients.size(); i++) {
      Client tmp = (Client) clients.get(i);
      if(tmp != c)
        tmp.write(data[0] + " " + data[1] + " " + data[2] + " " + data[3] + "\n");
    }
  }
}
```