

The background is a dark grey field filled with a dense network of white and light blue icons. These icons represent various digital concepts: a smartphone, a laptop, a magnifying glass, a person with a plus sign, a Twitter bird, a shopping cart, a play button, a home icon, a Wi-Fi symbol, a speech bubble, a music note, a link, a power button, a globe, a heart, a group of people, a mail icon, a search icon, a share icon, a refresh icon, a lock, a padlock, a speech bubble with a plus sign, a speech bubble with a minus sign, a speech bubble with a checkmark, a speech bubble with an 'x', a speech bubble with a question mark, a speech bubble with an exclamation mark, a speech bubble with a smiley face, a speech bubble with a sad face, a speech bubble with a neutral face, a speech bubble with a surprised face, a speech bubble with a angry face, a speech bubble with a love face, a speech bubble with a hate face, a speech bubble with a neutral face, a speech bubble with a surprised face, a speech bubble with an angry face, a speech bubble with a love face, a speech bubble with a hate face. A large, semi-transparent white rectangle with a blue border is centered in the lower half of the image. Inside this rectangle, the text 'The Internet' is written in a large, bold, black font. Below it, 'Nick Troccoli' and 'CSBridge CTU 2019' are written in a smaller, black font. At the bottom of the image, there is a large, faint, white outline of a smartphone.

The Internet

Nick Troccoli
CSBridge CTU 2019

Learning Goals

1. Write a program that can make internet requests
2. Write a program that can respond to internet requests



Plan For Today

- How do internet programs work?
- Sending Data Over The Internet
- Writing a client and server
- *Demo:* echo
- *Demo:* Chat

Plan For Today

- **How do internet programs work?**
- Sending Data Over The Internet
- Writing a client and server
- *Demo: echo*
- *Demo: Chat*

Programs and the Internet

How does your
phone
communicate with
Facebook?

Programs and the Internet

The Java program
on your phone talks
to the Java program
at Facebook.

Facebook

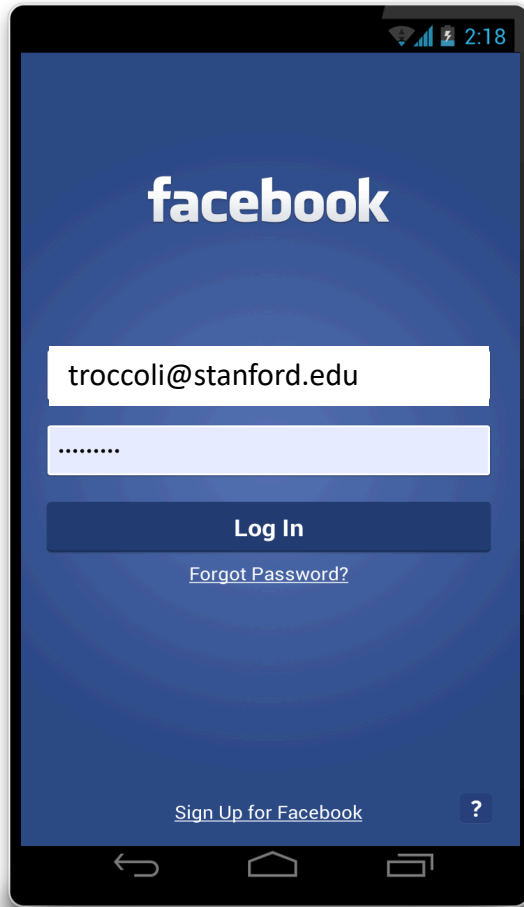


Facebook Server



* Android phones run Java. So do Facebook servers!

Facebook



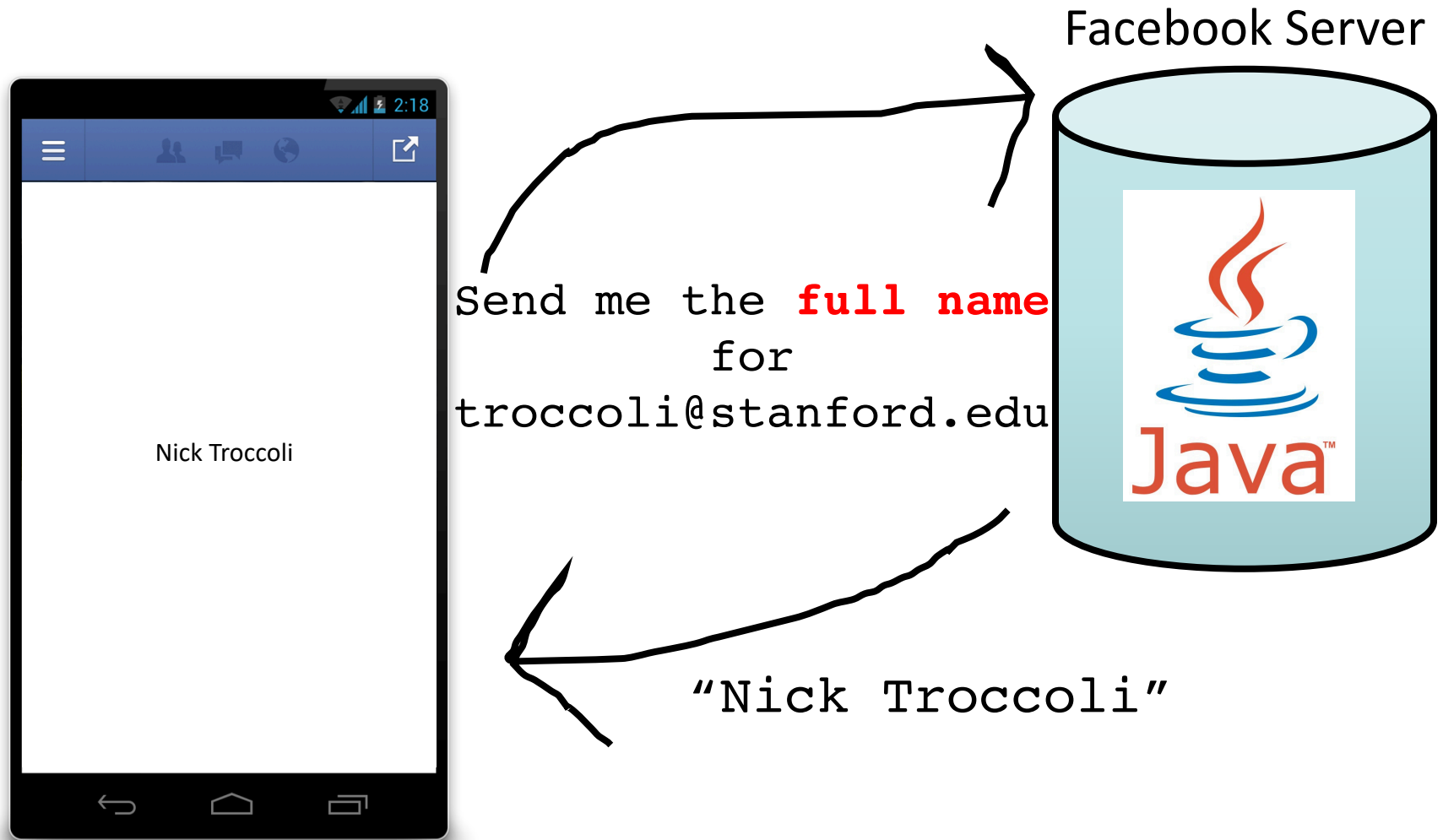
Is this login ok?

Facebook Server

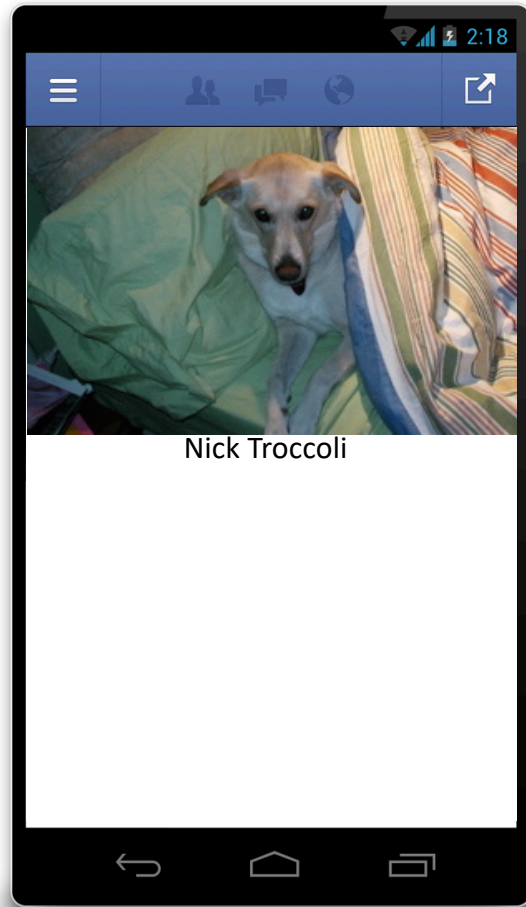


Confirmed.
troccoli@stanford.edu
is now logged in.

Facebook

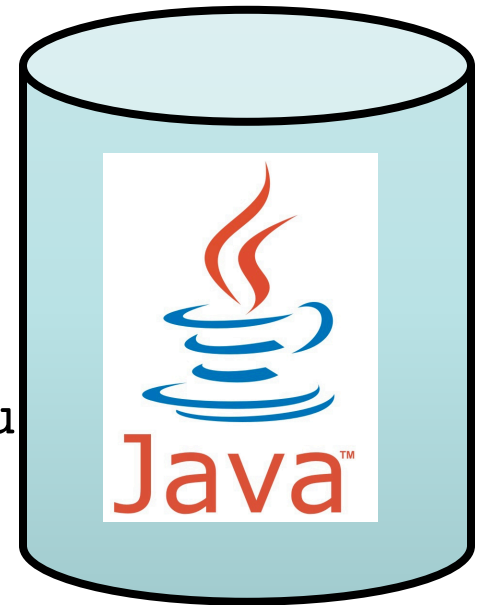


Facebook



Send me the **cover**
photo for
`troccoli@stanford.edu`

Facebook Server



Programs and the Internet

There are two types
of internet
programs: **clients**
and **servers**.

**Servers are
constantly running,
listening for requests
to respond to.**

Programs and the Internet

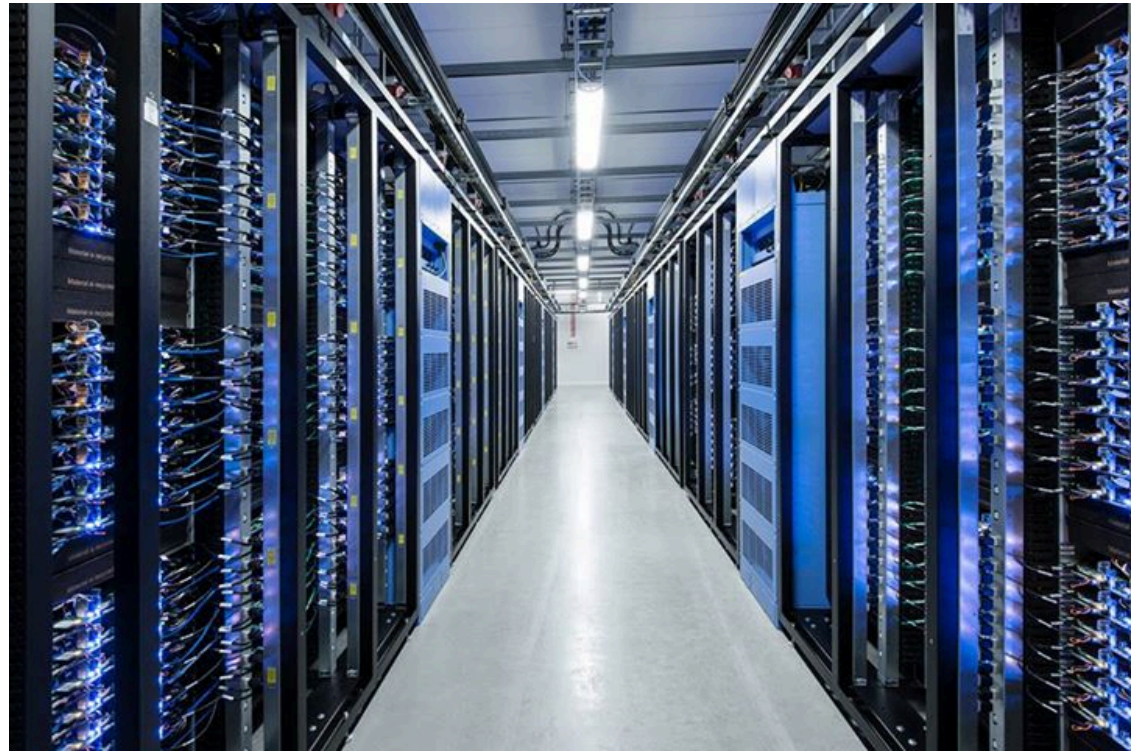
**Clients send requests
to servers for
information they need.**

Servers Are Computer Programs!

Facebook Server



=

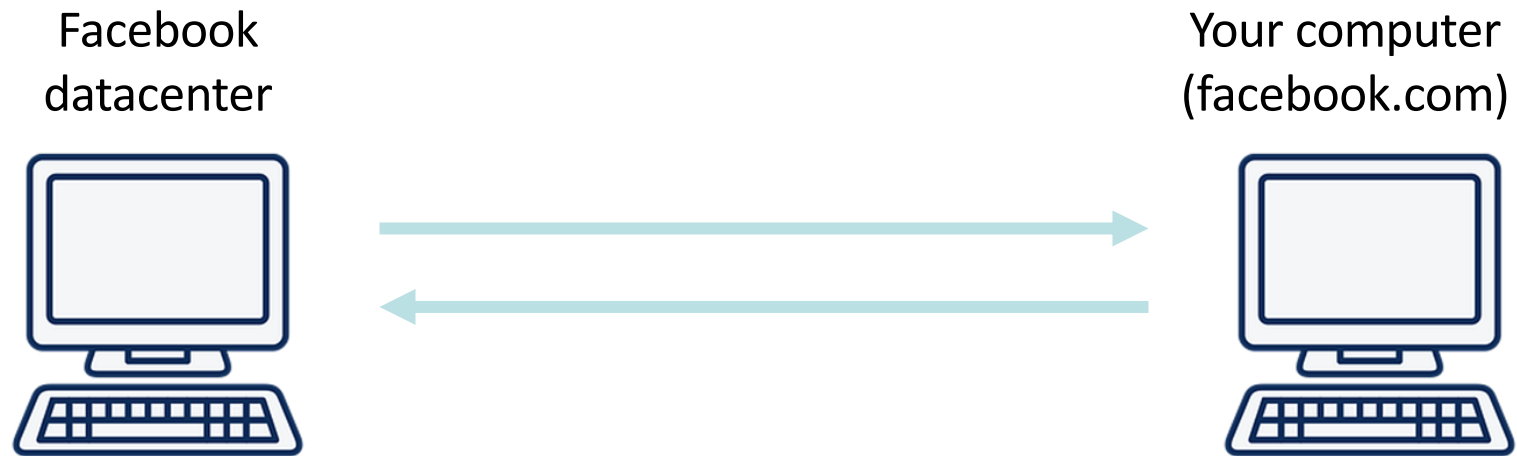


Background: The Internet



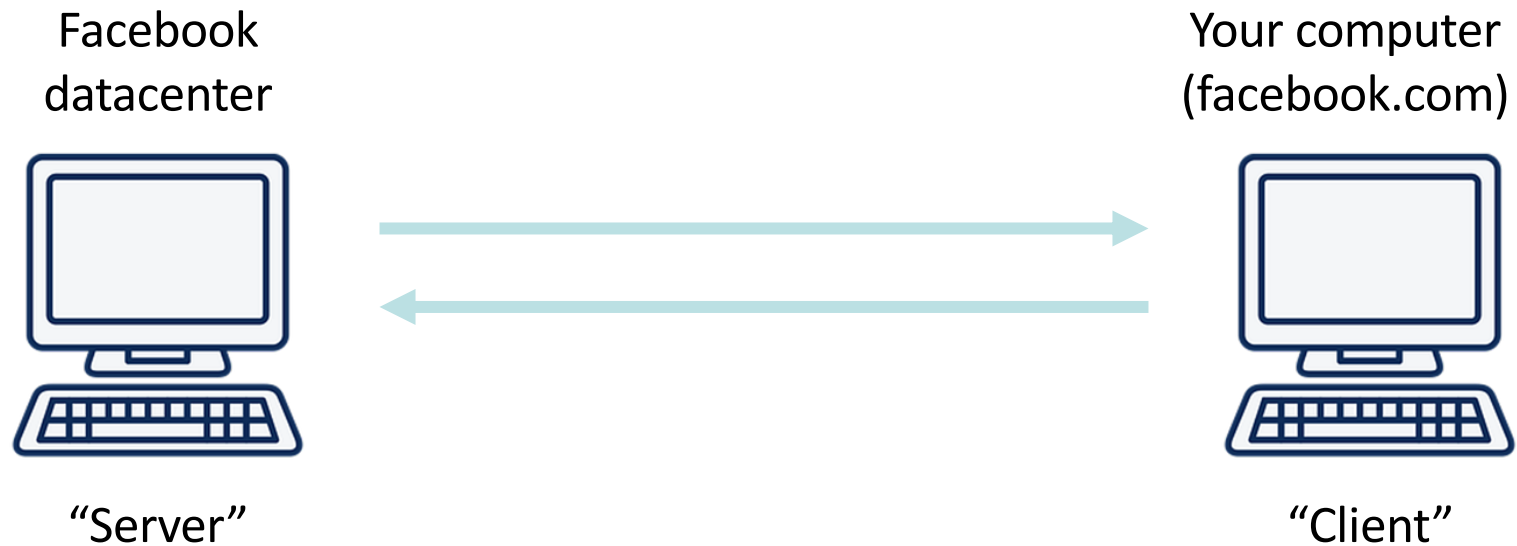
The internet is just many programs sending messages (as *Strings*)

Background: The Internet



The internet is just many programs sending messages (as *Strings*)

Background: The Internet



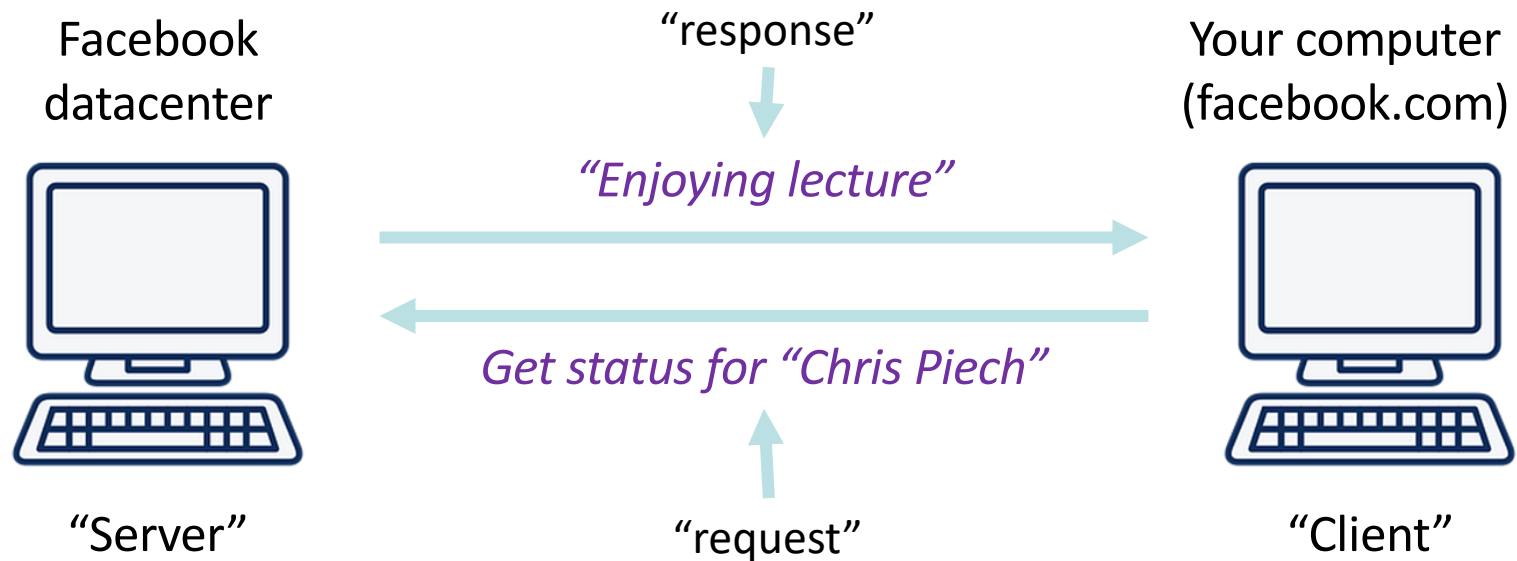
The internet is just many programs sending messages (as *Strings*)

Background: The Internet



The internet is just many programs sending messages (as *Strings*)

Background: The Internet



The internet is just many programs sending messages (as *Strings*)

The Internet



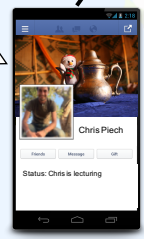
The Internet



Face Book Server



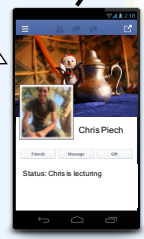
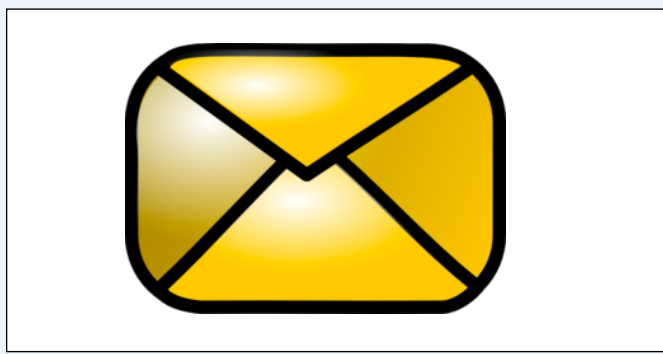
Get status for piech@cs.stanford.edu



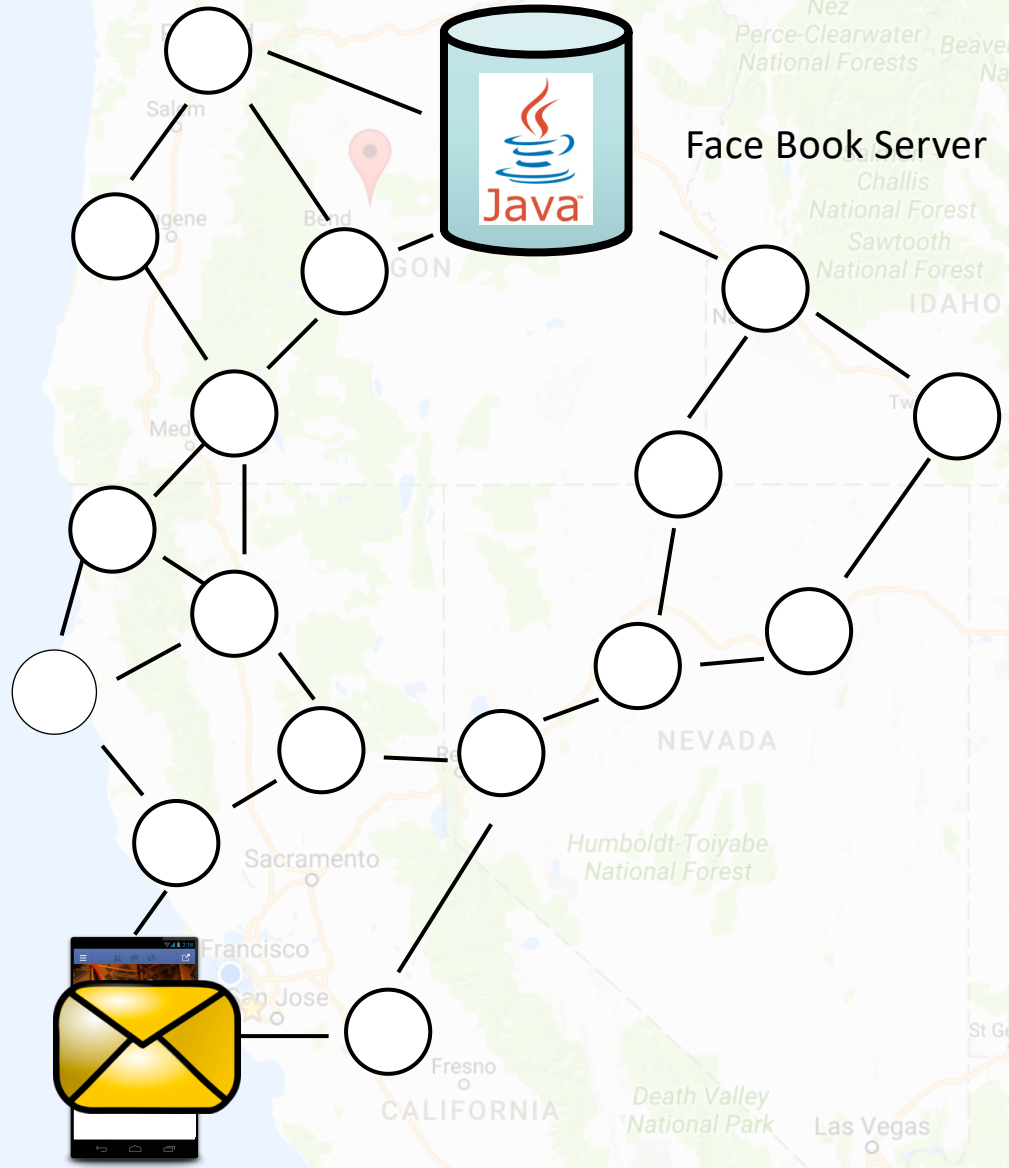
The Internet



Face Book Server

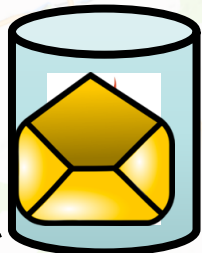


The Internet

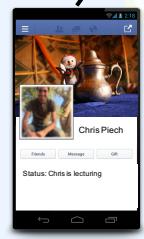
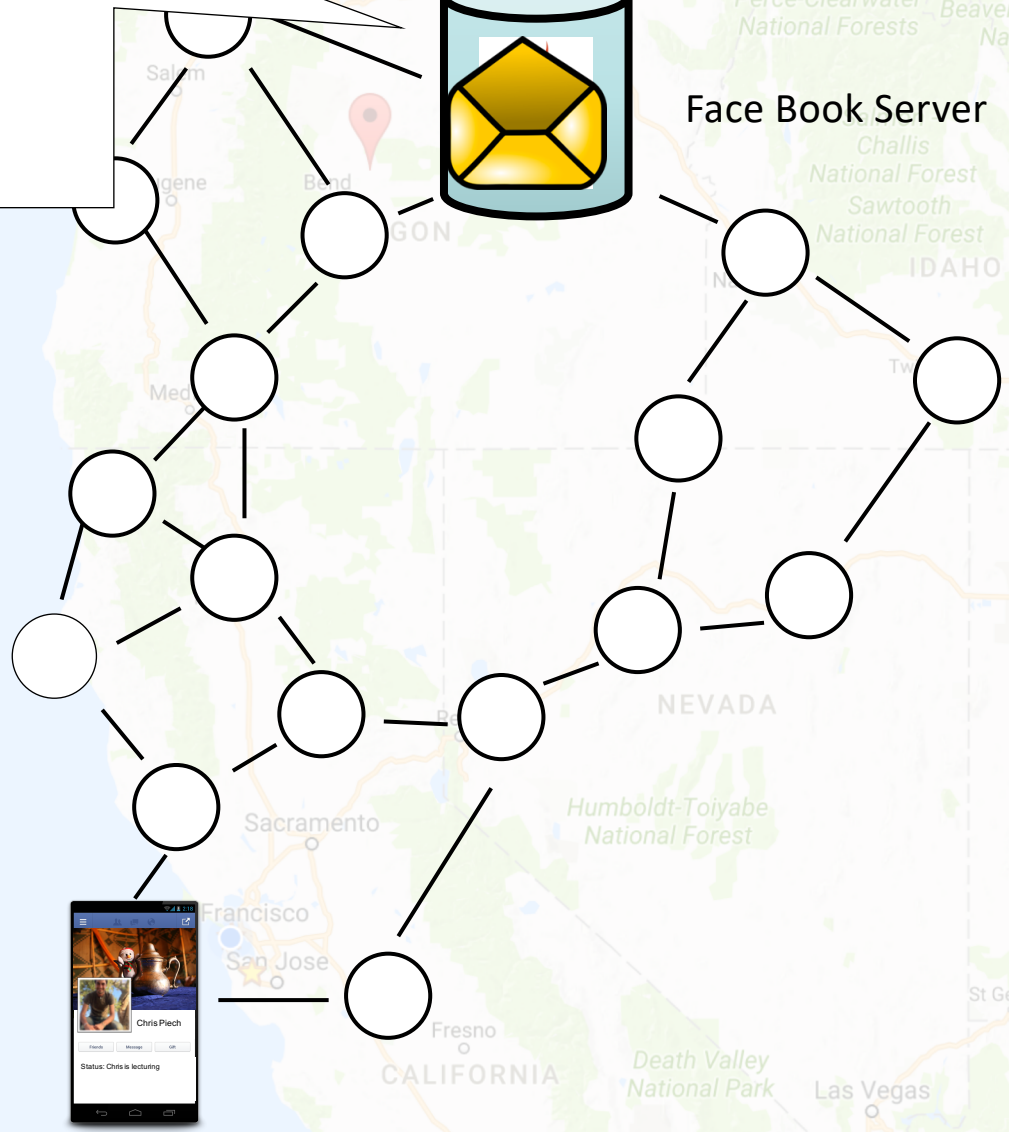


teaching

The Internet



Face Book Server

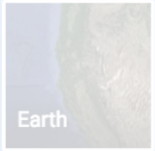




The Internet



Face Book Server

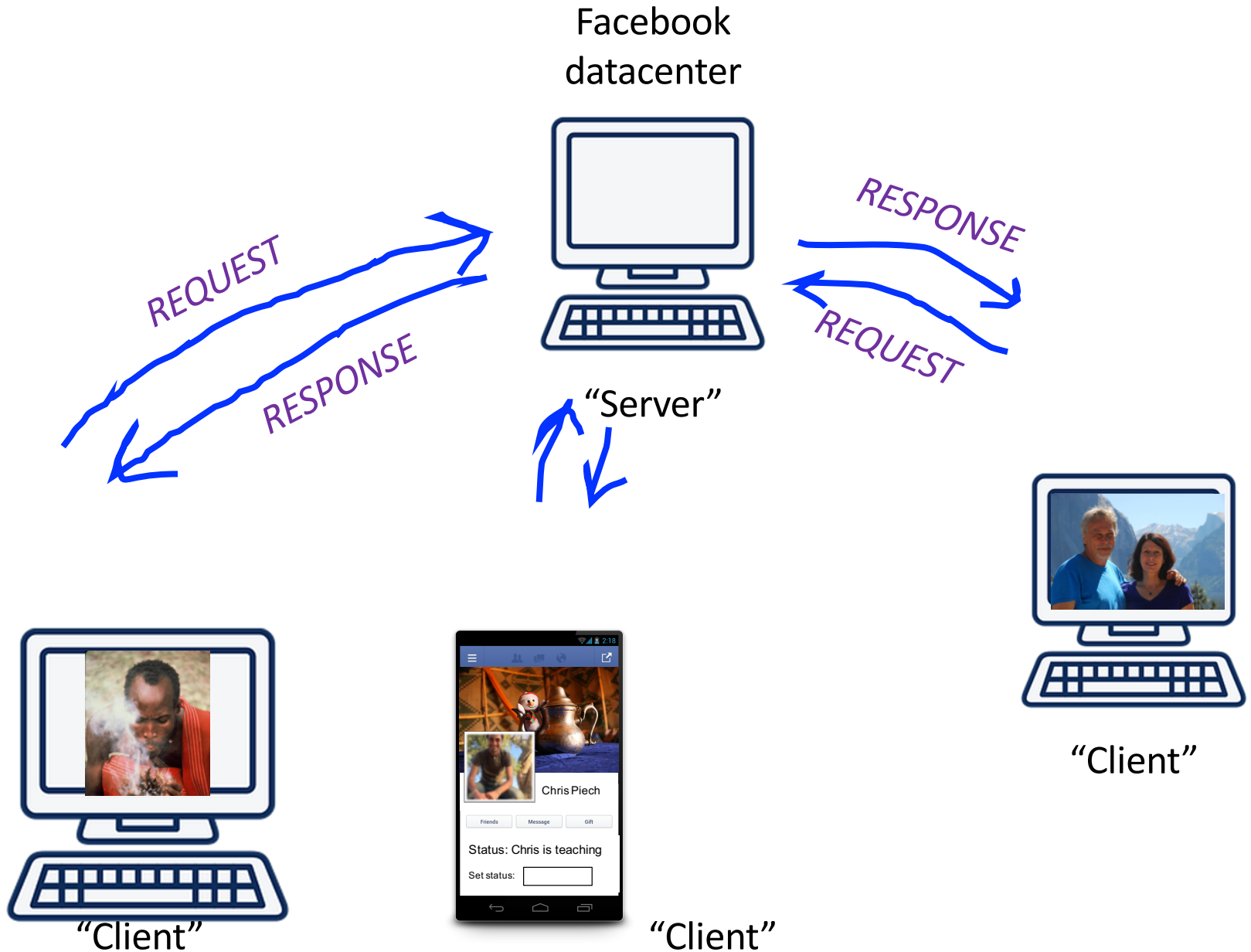


The Internet

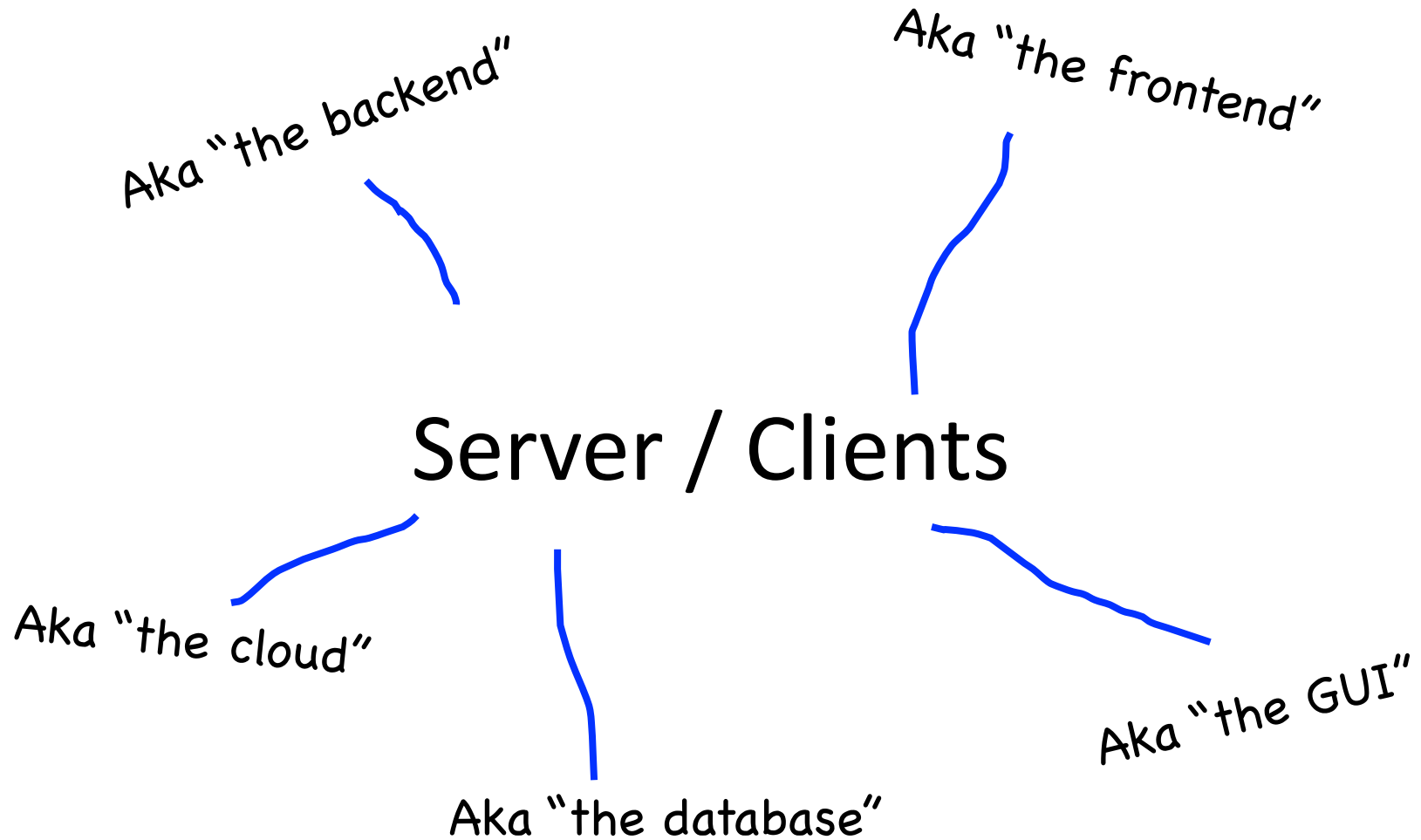


Many programs can connect
to the same server

The Internet



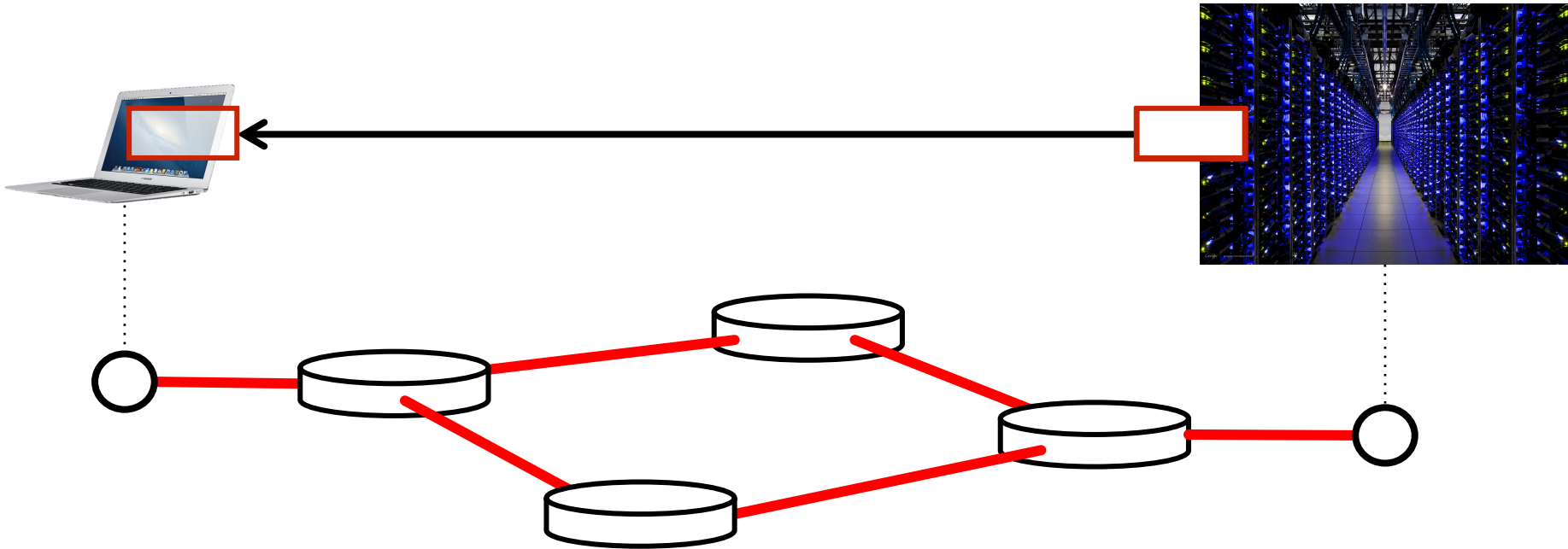
Most of the Internet



Plan For Today

- How do internet programs work?
- **Sending Data Over The Internet**
- Writing a client and server
- *Demo: echo*
- *Demo: Chat*

Google YouTube facebook

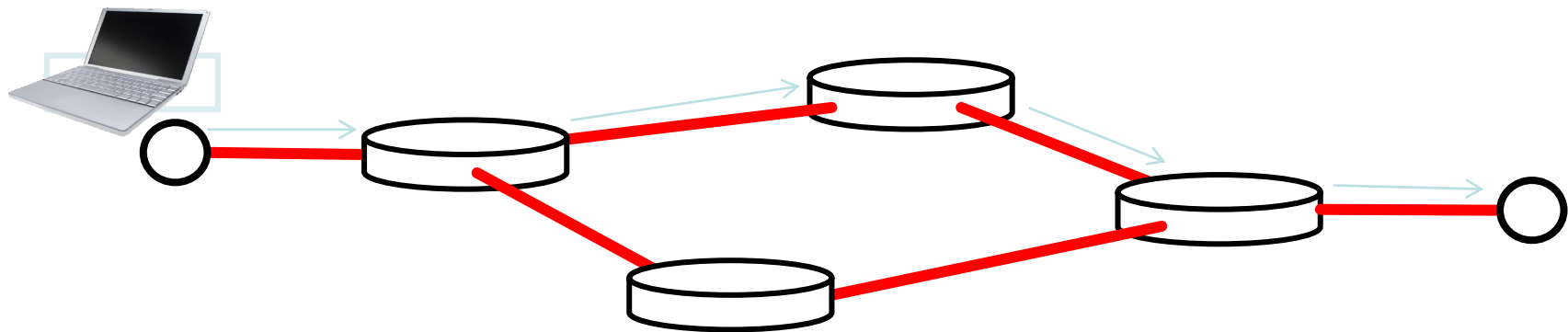


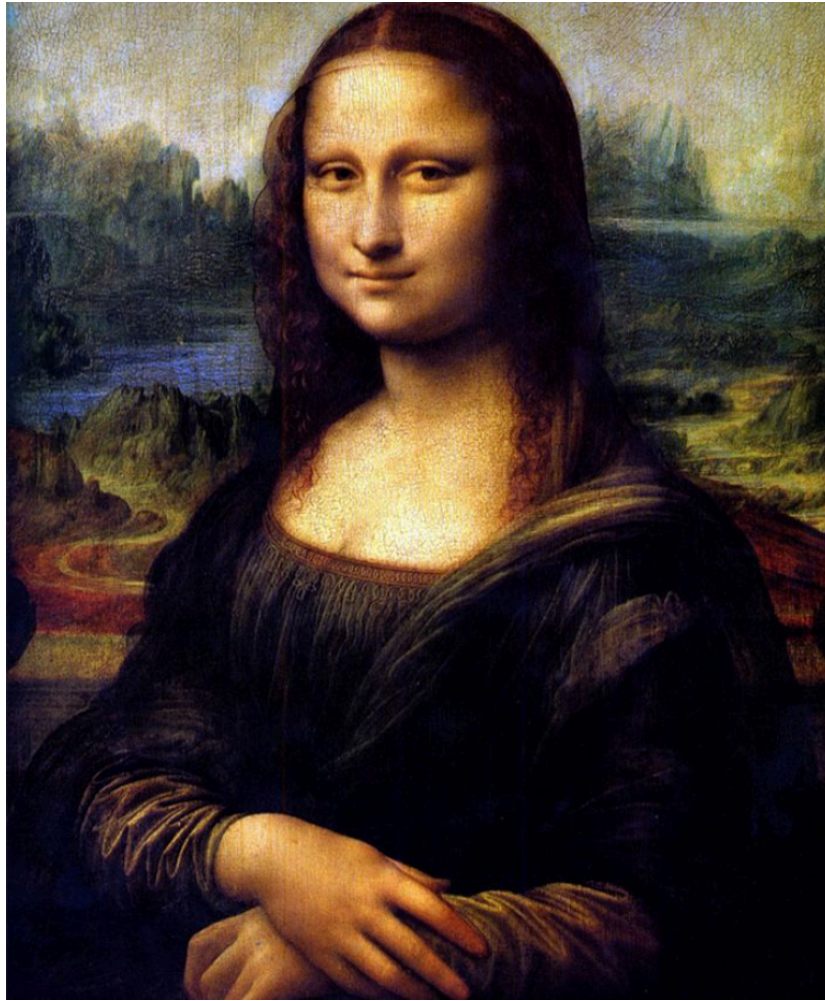
My Java Program

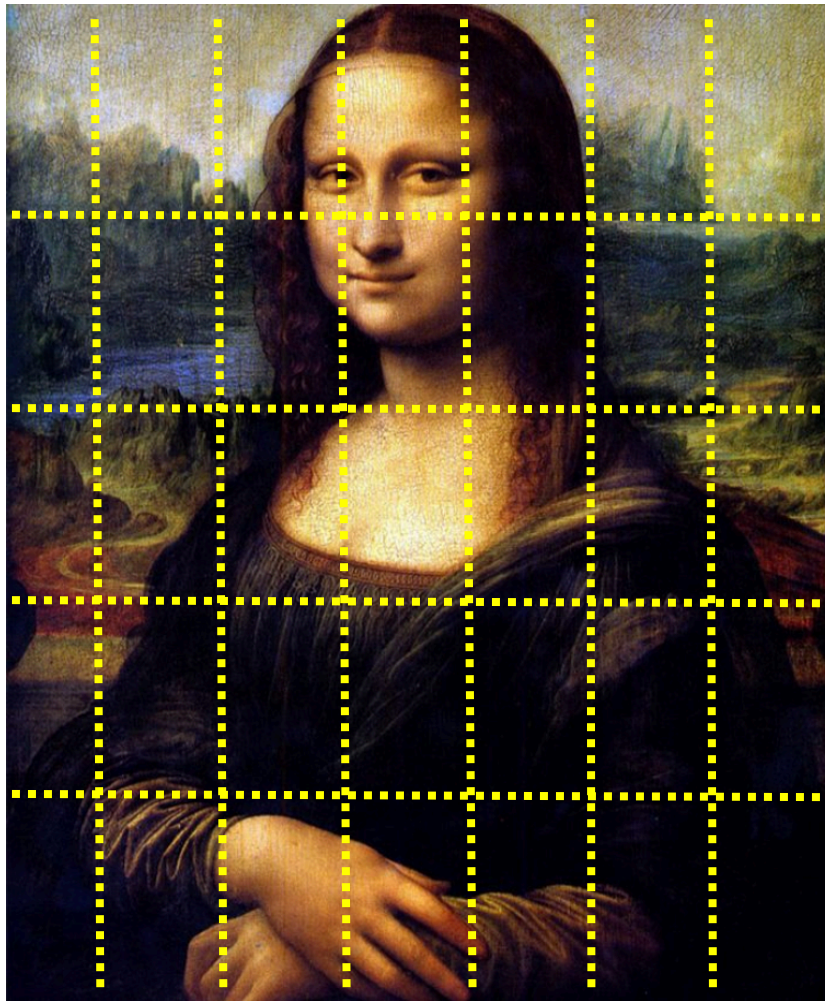


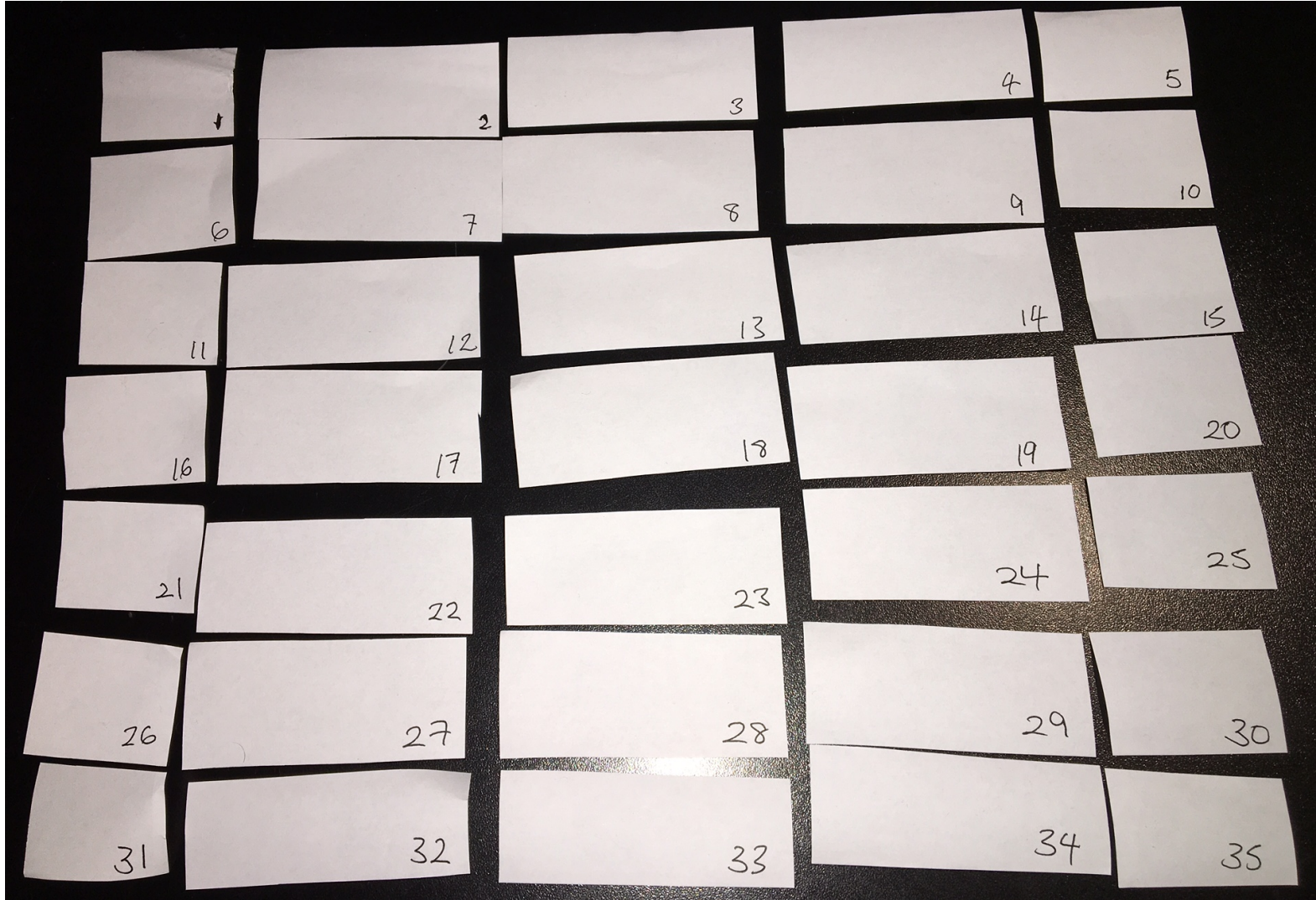
Someone else's Java Program



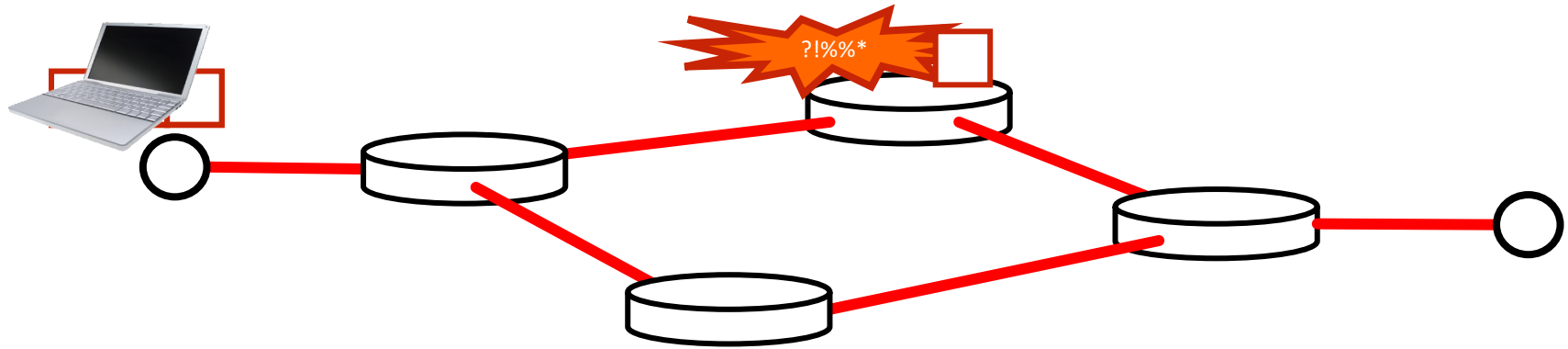




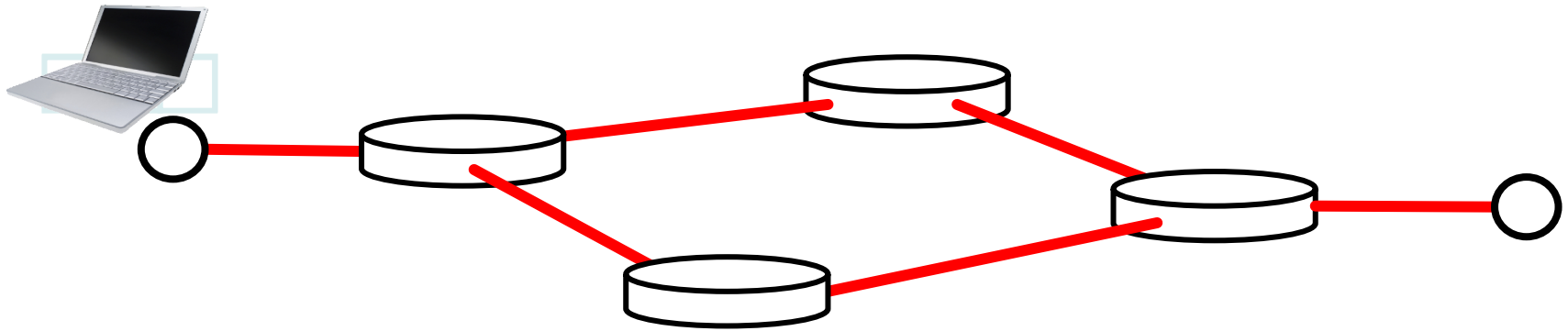




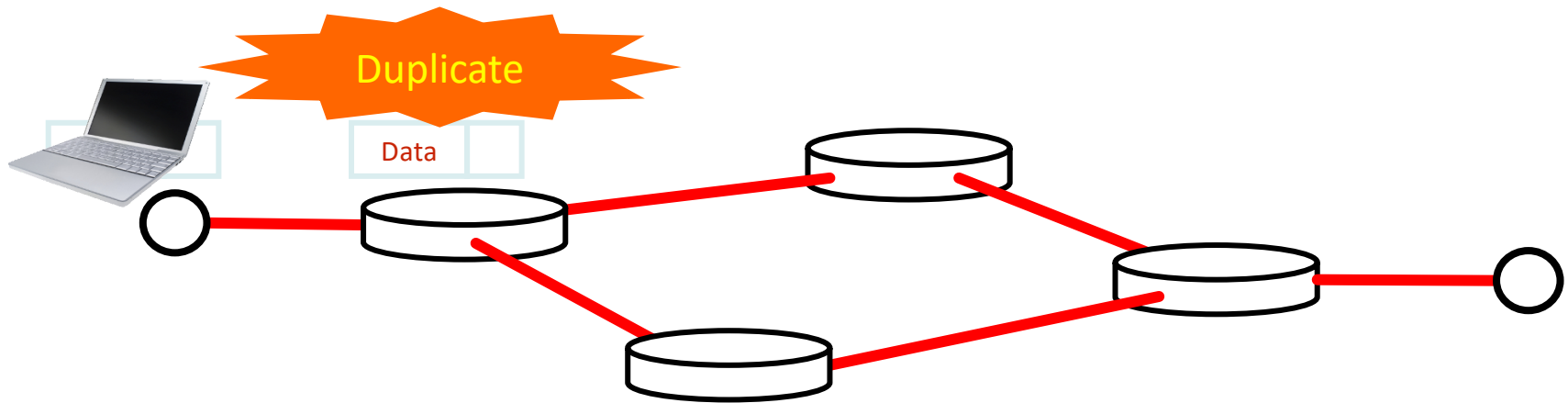
Packets may be **damaged**



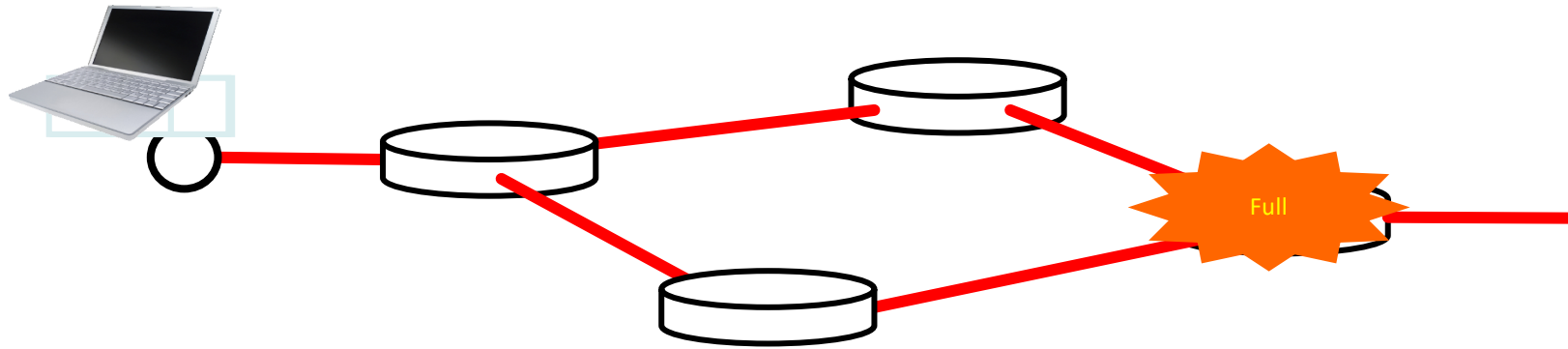
Packets may arrive **out of order**



Packets may be **deduplicated**

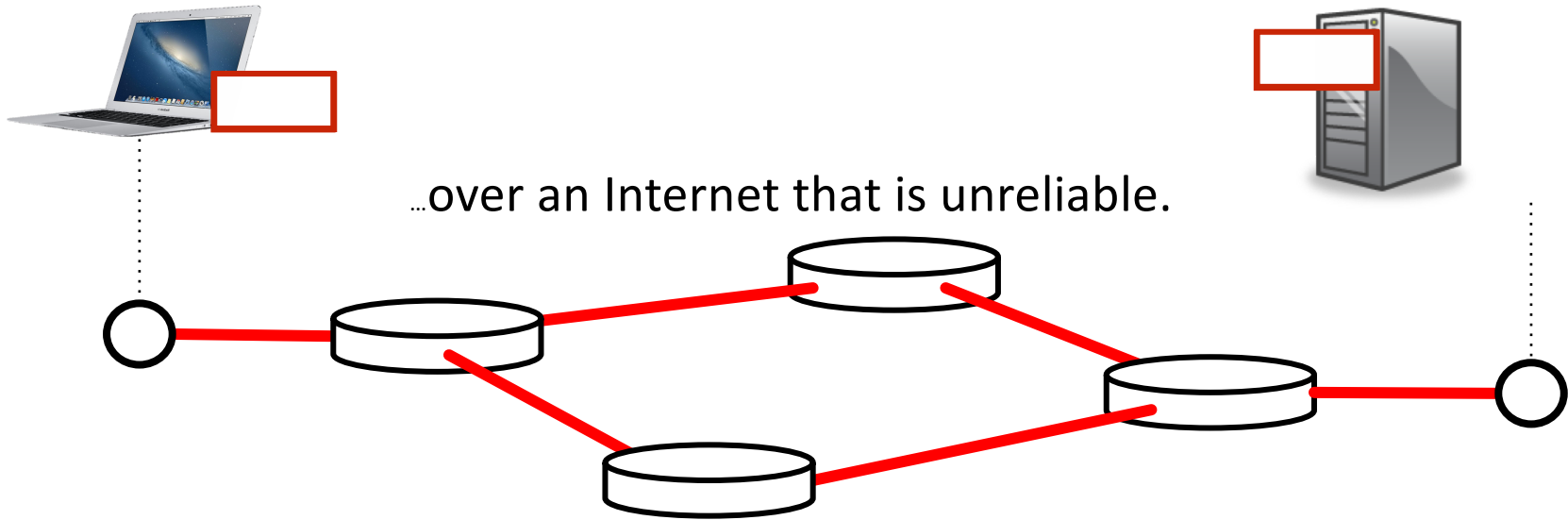


They may not arrive at all!



The Internet Is Unreliable

Applications send and receive data in packets....



Software on the devices themselves compensates for this.

Let's send a message over the "CSBridgeNet"

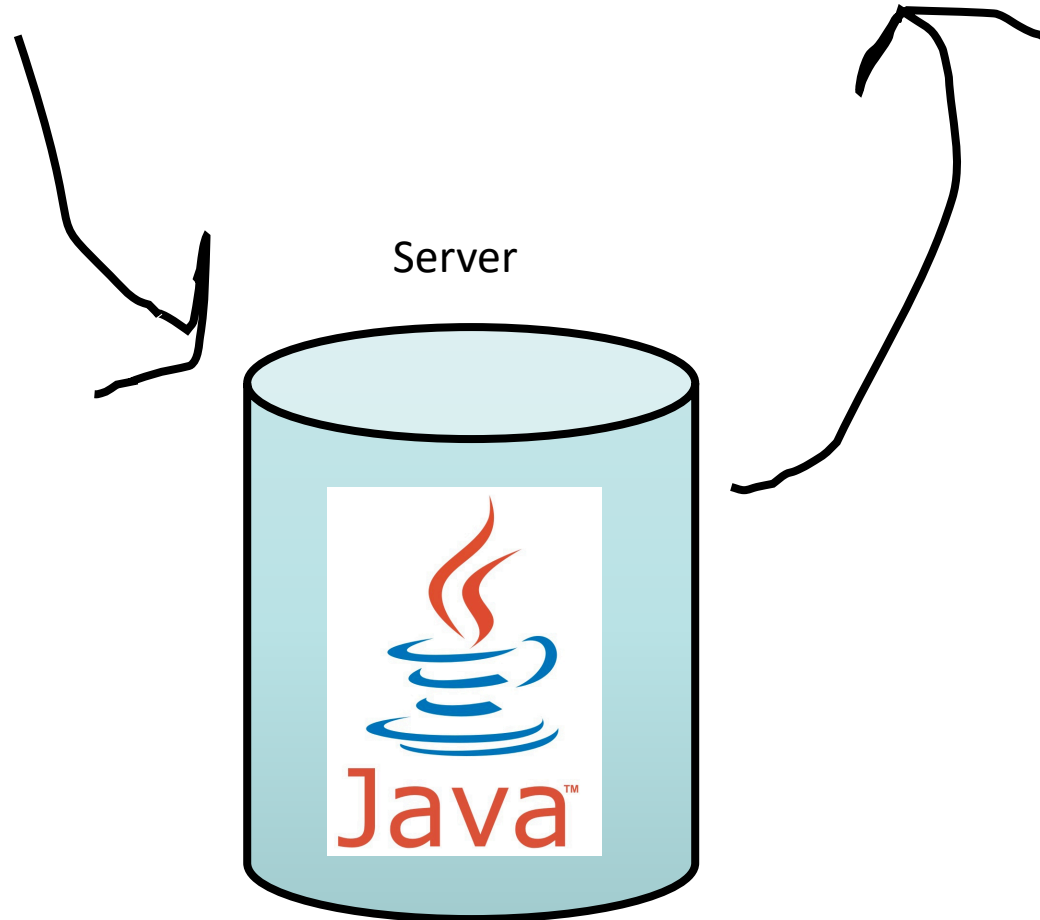
Plan For Today

- How do internet programs work?
- Sending Data Over The Internet
- **Writing a client and server**
- *Demo: echo*
- *Demo: Chat*

Servers

Request
someRequest

String
serverResponse



Servers on one slide

1

```
public String requestMade(Request request) {  
    // server code goes here  
}
```

2

```
// make a Server object  
private SimpleServer server  
    = new SimpleServer(this, 8000);
```

3

```
public void run(){  
    // start the server  
    server.start();  
}
```

A Server's Simple Purpose

1

```
public String requestMade(Request request) {  
    // server code goes here  
}
```

2

```
// make a Server object  
private SimpleServer server  
    = new SimpleServer(this, 8000);
```

3

```
public void run(){  
    // start the server  
    server.start();  
}
```


A Server's Simple Purpose

1

```
public String requestMade(Request request) {  
    // server code goes here  
}
```

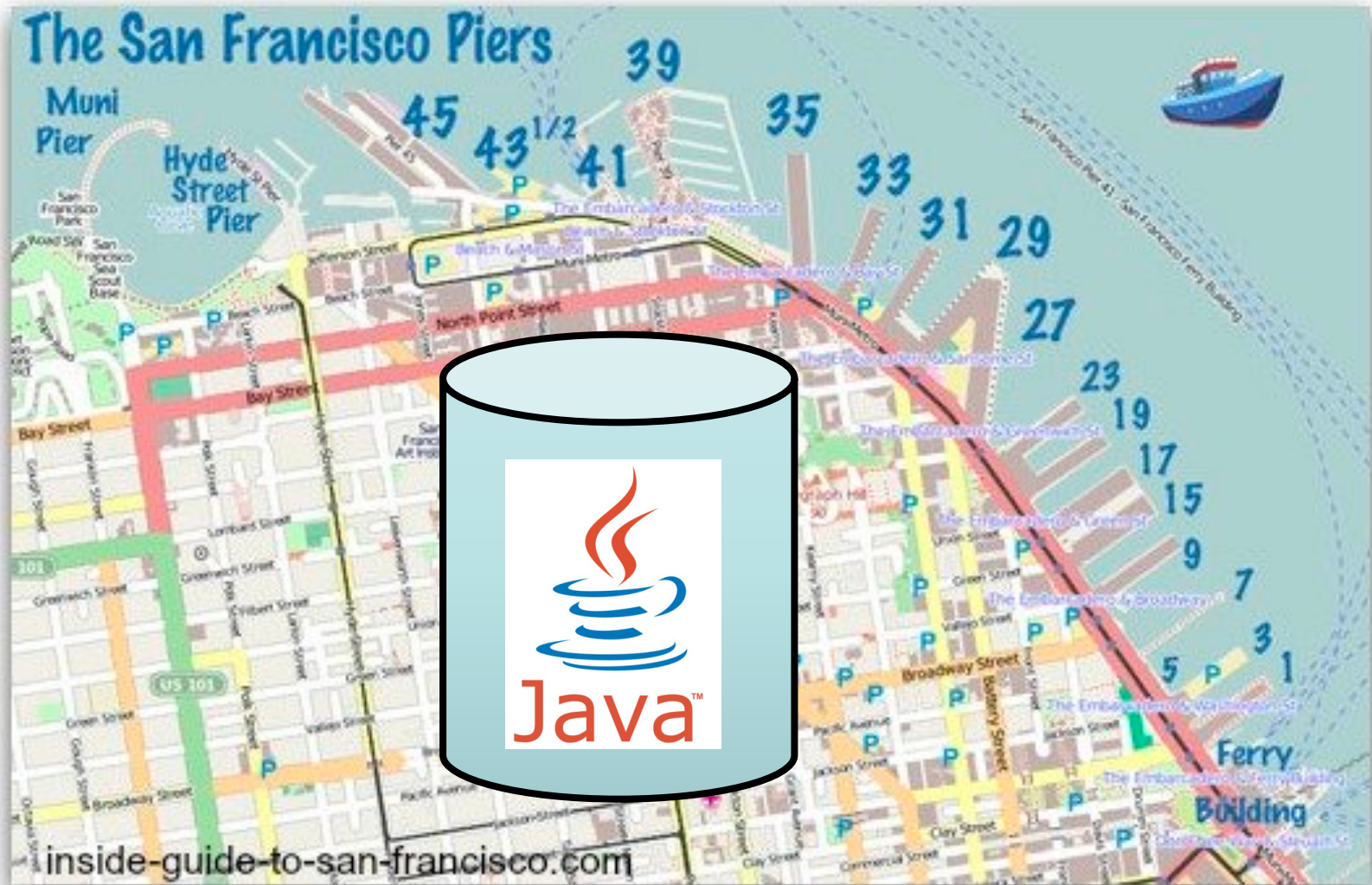
2

```
// make a Server object  
private SimpleServer server  
    = new SimpleServer(this, 8000);
```

3

```
public void run(){  
    // start the server  
    server.start();  
}
```

What is a Port?



Servers on one slide

1

```
public String requestMade(Request request) {  
    // server code goes here  
}
```

2

```
// make a Server object  
private SimpleServer server  
    = new SimpleServer(this, 8000);
```

3

```
public void run(){  
    // start the server  
    server.start();  
}
```

What is a Request?



```
/* Request has a command */  
String command;
```

```
/* Request has parameters */  
HashMap<String,String> params;
```

Request request

```
// methods that the server calls on requests  
request.getCommand();  
request.getParam(key); //returns associated value
```

Plan For Today

- How do internet programs work?
- Sending Data Over The Internet
- Writing a client and server
- *Demo: echo*
- *Demo: Chat*

Our First Server

```
public class EchoServer extends ConsoleProgram implements
SimpleServerListener {

    private SimpleServer server = new SimpleServer(this, 8090);

    public void run() {
        server.start();
        println("Starting server...");
    }

    public String requestMade(Request request) {
        String cmd = request.getCommand();
        int cmdLength = cmd.length();
        return "Your command was " + cmdLength + " chars long.";
    }

}
```

Our First Server

```
public class EchoServer extends ConsoleProgram implements
SimpleServerListener {

    private SimpleServer server = new SimpleServer(this, 8090);

    public void run() {
        server.start();
        println("Starting server...");
    }

    public String requestMade(Request request) {
        String cmd = request.getCommand();
        int cmdLength = cmd.length();
        return "Your command was " + cmdLength + " chars long.";
    }

}
```

Our First Server

```
public class EchoServer extends ConsoleProgram implements
SimpleServerListener {

    private SimpleServer server = new SimpleServer(this, 8090);

    public void run() {
        server.start();
        println("Starting server...");
    }

    public String requestMade(Request request) {
        String cmd = request.getCommand();
        int cmdLength = cmd.length();
        return "Your command was " + cmdLength + " chars long.";
    }

}
```


Our First Server

```
public class EchoServer extends ConsoleProgram implements
SimpleServerListener {

    private SimpleServer server = new SimpleServer(this, 8090);

    public void run() {
        server.start();
        println("Starting server...");
    }

    public String requestMade(Request request) {
        String cmd = request.getCommand();
        int cmdLength = cmd.length();
        return "Your command was " + cmdLength + " chars long.";
    }

}
```

Our First Server

```
public class EchoServer extends ConsoleProgram implements
SimpleServerListener {

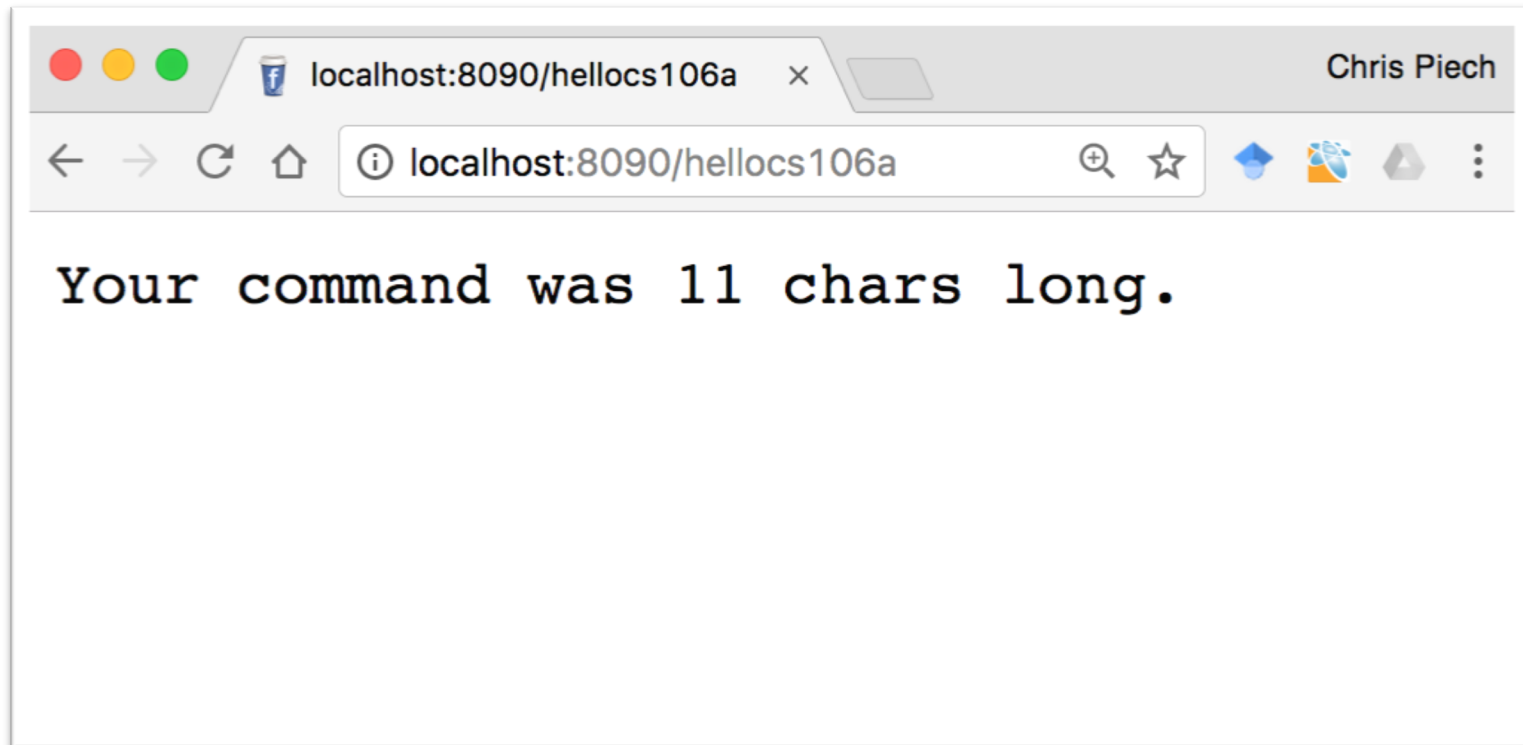
    private SimpleServer server = new SimpleServer(this, 8090);

    public void run() {
        server.start();
        println("Starting server...");
    }

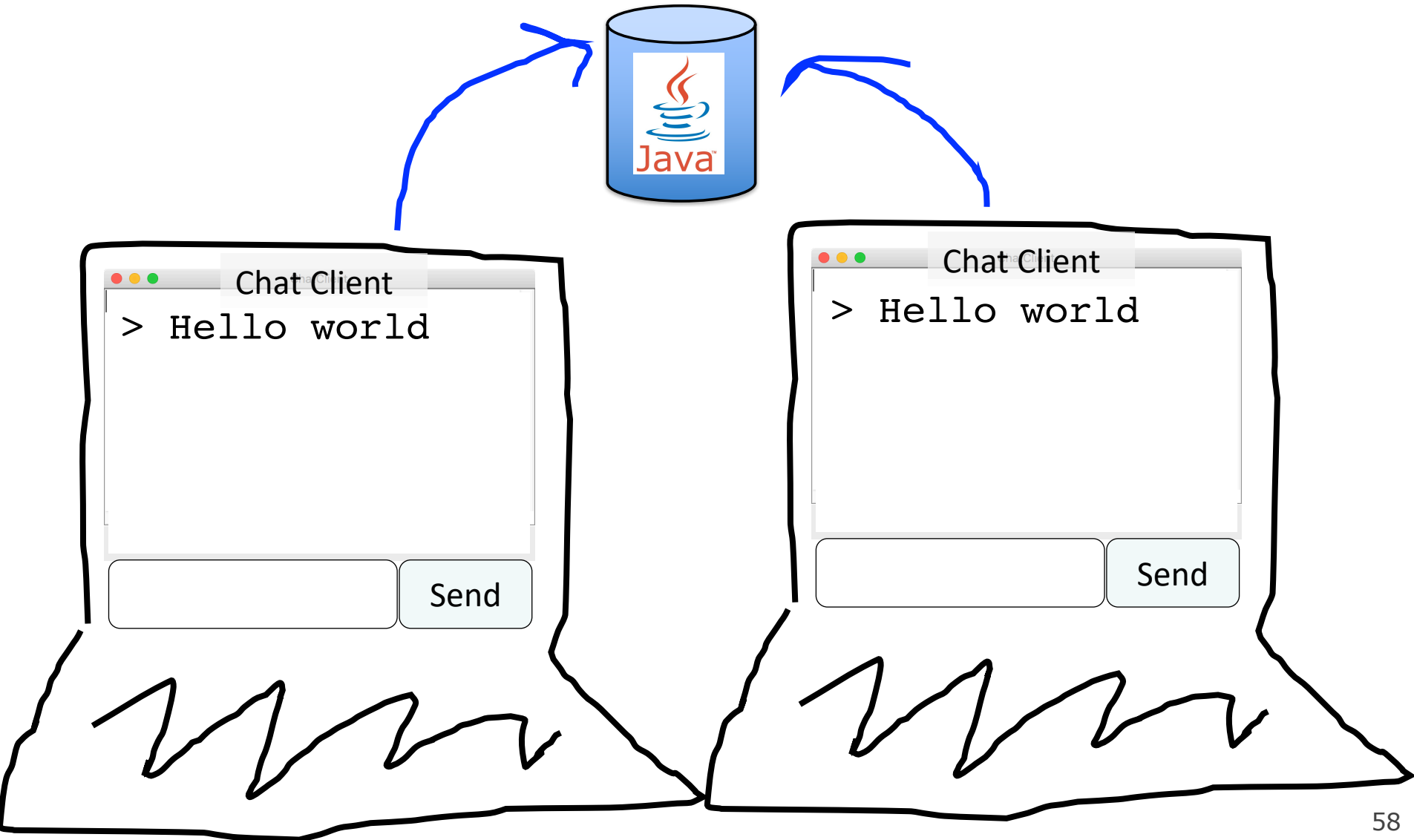
    public String requestMade(Request request) {
        String cmd = request.getCommand();
        int cmdLength = cmd.length();
        return "Your command was " + cmdLength + " chars long.";
    }

}
```

Echo Server



Clients



Clients on one slide

```
try {  
  
    // 1. construct a new request  
    Request example = new Request("getStatus");  
  
    // 2. add parameters to the request  
    example.addParam("name", "chris");  
  
    // 3. send the request to a computer on the internet  
    String result = SimpleClient.makeRequest(HOST, example);  
  
} catch(IOException e) {  
  
    // The internet is a fast and wild world my friend  
  
}
```

Clients on one slide

```
try {  
    // 1. construct a new request  
    Request example = new Request("getStatus");  
  
    // 2. add parameters to the request  
    example.addParam("name", "chris");  
  
    // 3. send the request to a computer on the internet  
    String result = SimpleClient.makeRequest(HOST, example);  
} catch(IOException e) {  
    // The internet is a fast and wild world my friend  
}
```

Clients on one slide

```
try {  
    // 1. construct a new request  
    Request example = new Request("getStatus");  
  
    // 2. add parameters to the request  
    example.addParam("name", "chris");  
  
    // 3. send the request to a computer on the internet  
    String result = SimpleClient.makeRequest(HOST, example);  
} catch(IOException e) {  
    // The internet is a fast and wild world my friend  
}
```

Clients on one slide

```
try {  
  
    // 1. construct a new request  
    Request example = new Request("getStatus");  
  
    // 2. add parameters to the request  
    example.addParam("name", "chris");  
  
    // 3. send the request to a computer on the internet  
    String result = SimpleClient.makeRequest(HOST, example);  
  
} catch(IOException e) {  
  
    // The internet is a fast and wild world my friend  
  
}
```


Clients on one slide

```
try {  
  
    // 1. construct a new request  
    Request example = new Request("getStatus");  
  
    // 2. add parameters to the request  
    example.addParam("name", "chris");  
  
    // 3. send the request to a computer on the internet  
    String result = SimpleClient.makeRequest(HOST, example);  
  
} catch(IOException e) {  
  
    // The internet is a fast and wild world my friend  
  
}
```

Clients on one slide

```
try {  
    // 1. construct a new request  
    Request example = new Request("getStatus");  
  
    // 2. add parameters to the request  
    example.addParam("name", "chris");  
  
    // 3. send the request to a computer on the internet  
    String result = SimpleClient.makeRequest(HOST, example);  
} catch(IOException e) {  
    // The internet is a fast and wild world my friend  
}
```

Clients on one slide

```
try {  
    // 1. construct a new request  
    Request example = new Request("getStatus");  
  
    // 2. add parameters to the request  
    example.addParam("name", "chris");  
  
    // 3. send the request to a computer on the internet  
    String result = SimpleClient.makeRequest(HOST, example);  
} catch(IOException e) {  
    // The internet is a fast and wild world my friend  
}
```

Clients on one slide

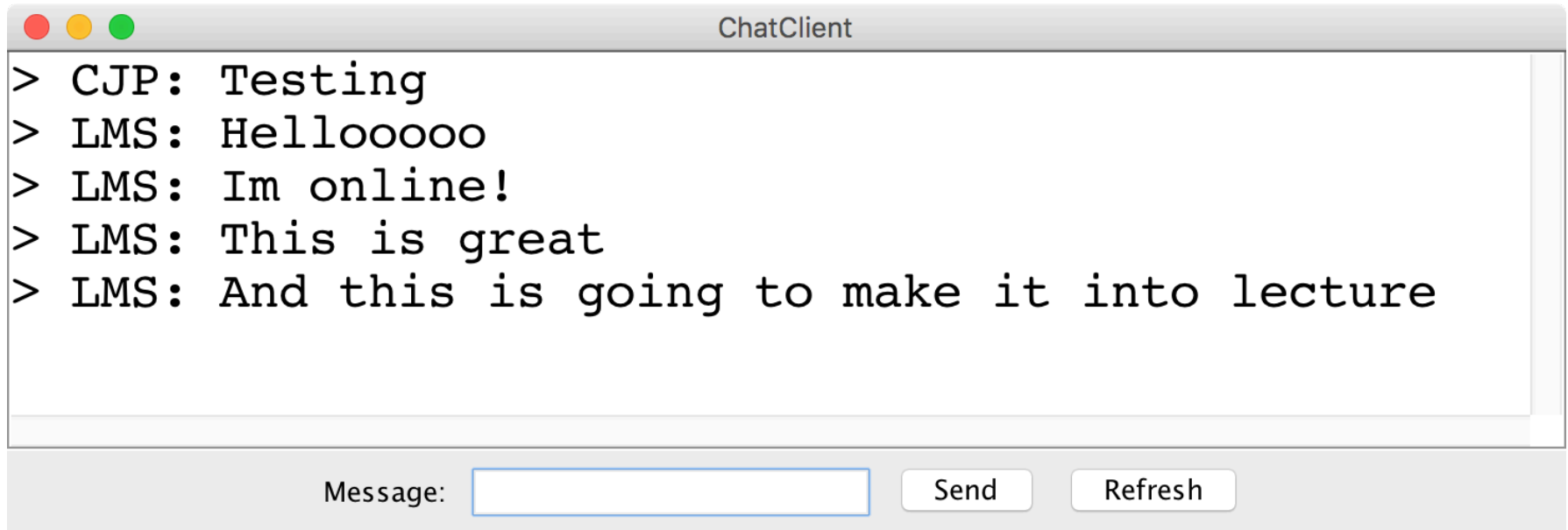
```
try {  
    // 1. construct a new request  
    Request example = new Request("getStatus");  
  
    // 2. add parameters to the request  
    example.addParam("name", "chris");  
  
    // 3. send the request to a computer on the internet  
    String result = SimpleClient.makeRequest(HOST, example);  
} catch(IOException e) {  
    // The internet is a fast and wild world my friend  
}
```

Example: Echo Client

Plan For Today

- How do internet programs work?
- Sending Data Over The Internet
- Writing a client and server
- *Demo: echo*
- ***Demo: Chat***

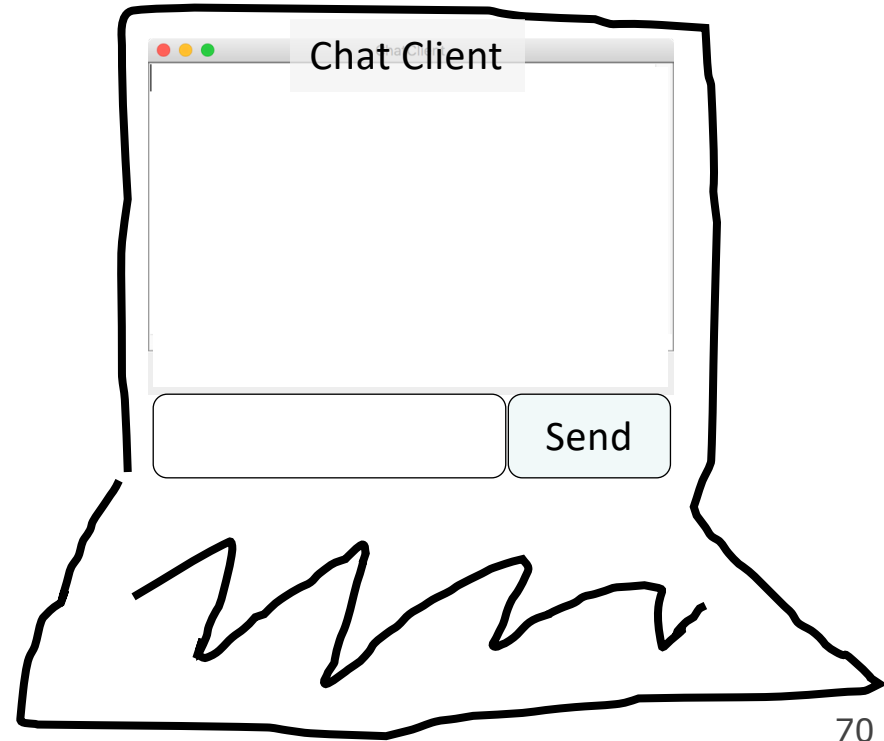
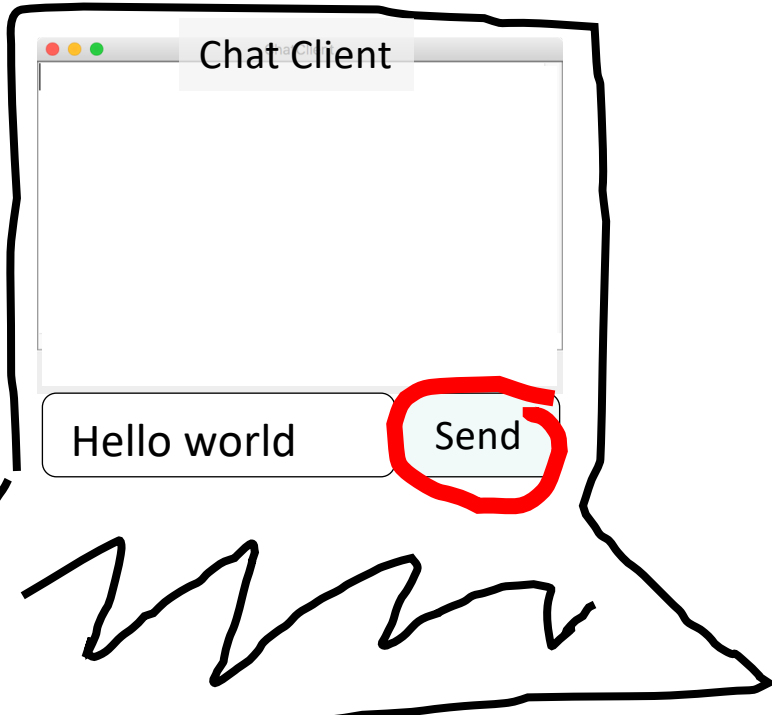
Chat Server and Client





```
history = [  
]
```

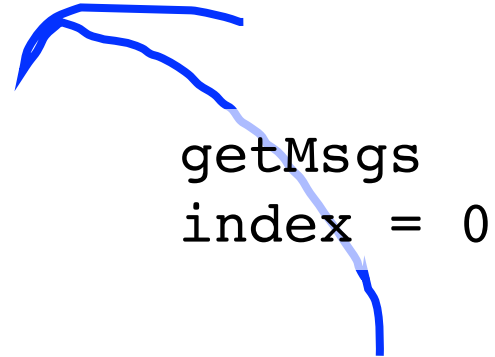
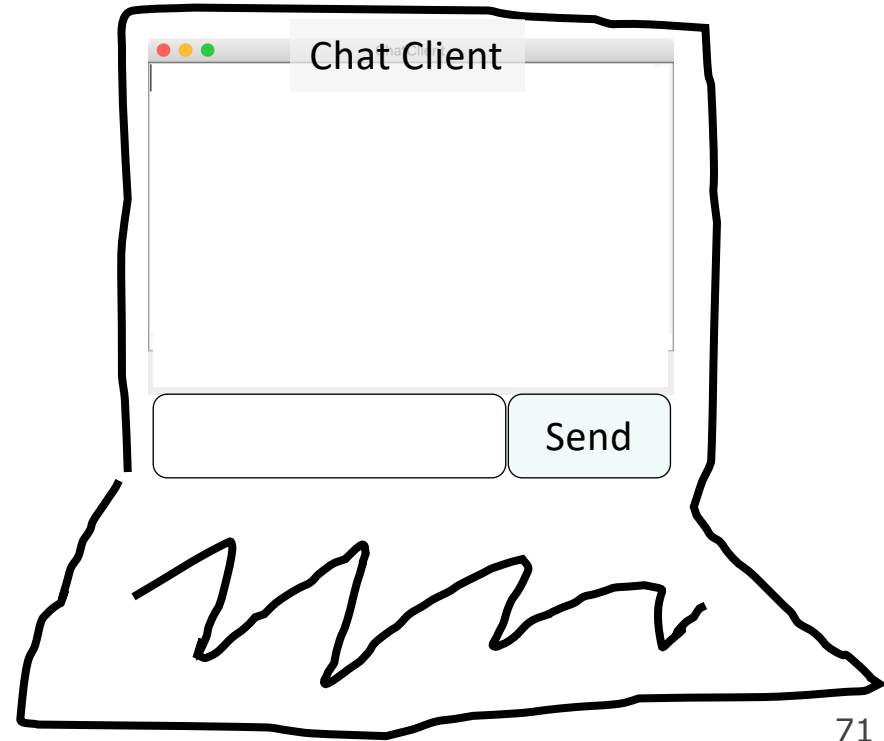
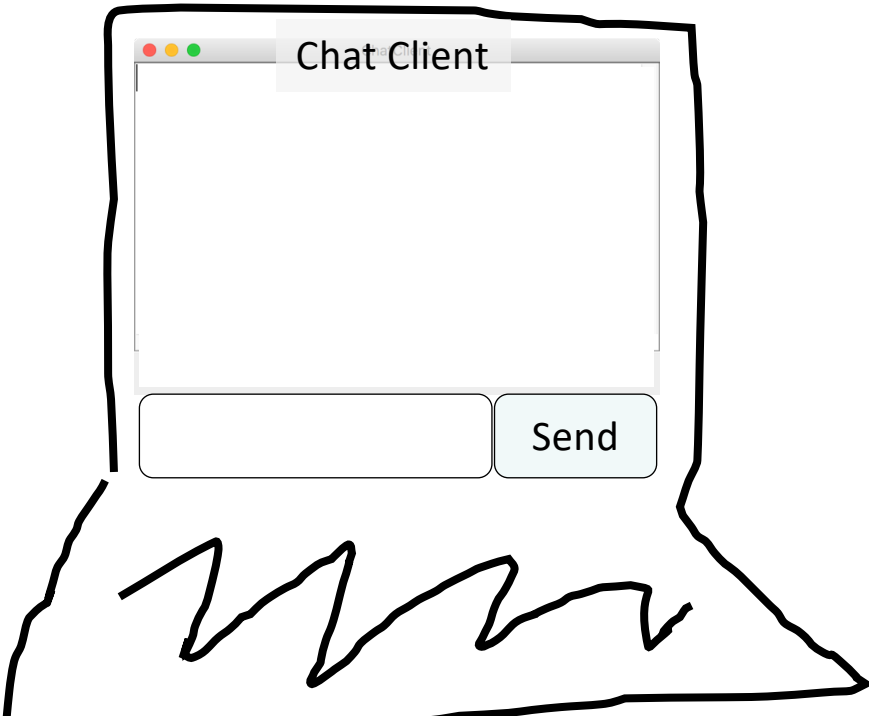
```
addMsg  
msg = C: Hello world
```





```
history = [  
    C: Hello world  
]
```

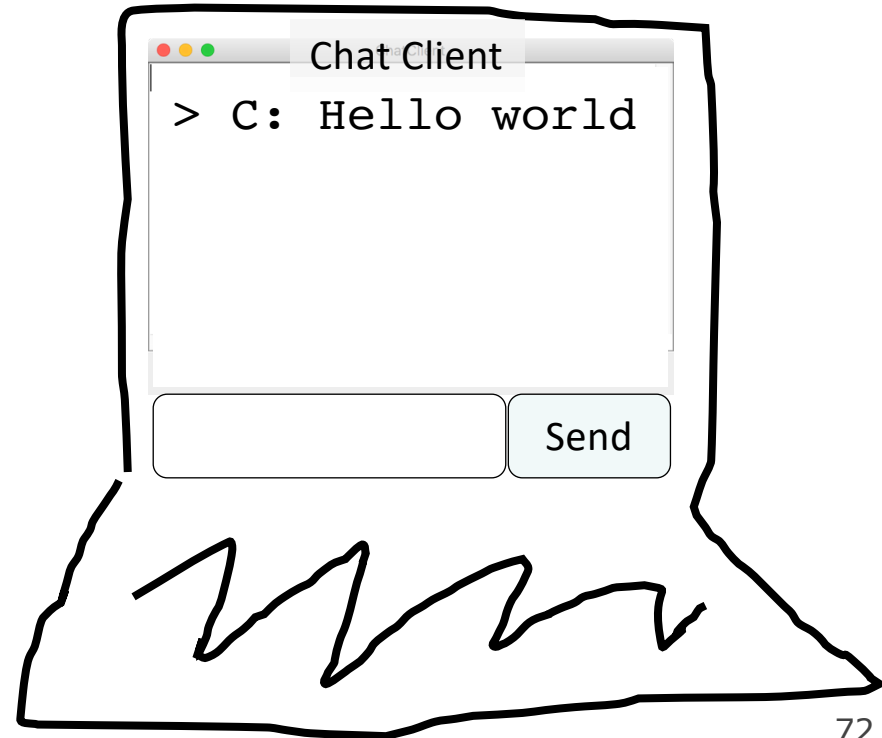
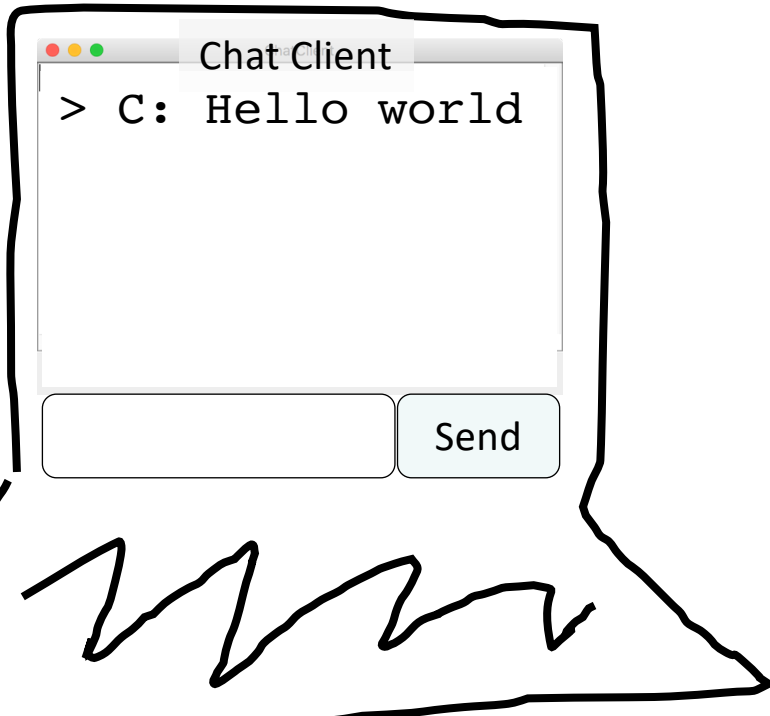
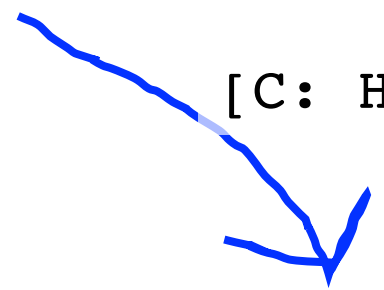
getMsgs
index = 0

A blue arrow originates from the text "getMsgs" and points towards the right-hand "Chat Client" window.



```
history = [  
    C: Hello world  
]
```

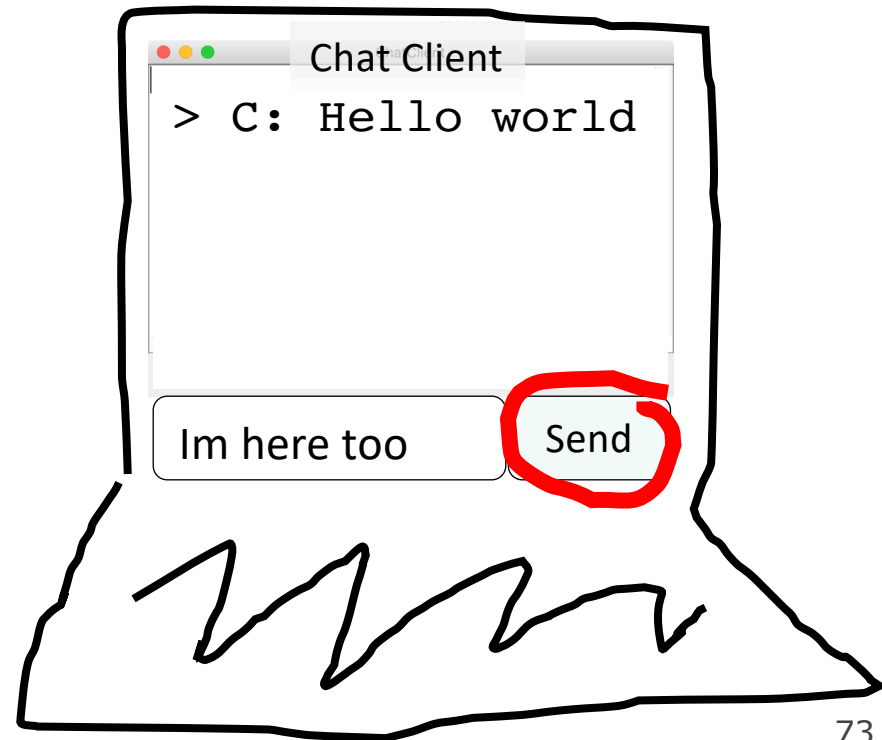
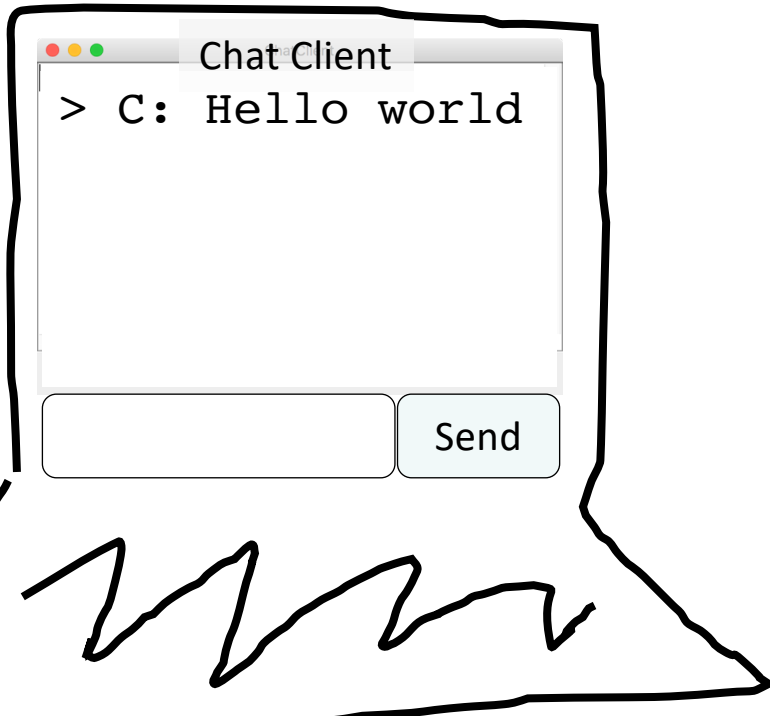
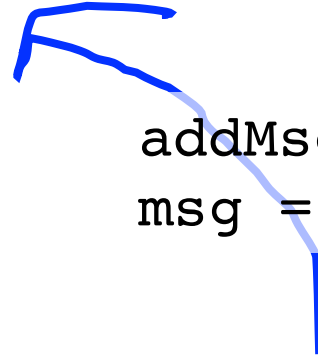
[C: Hello world]





```
history = [  
    C: Hello world  
]
```

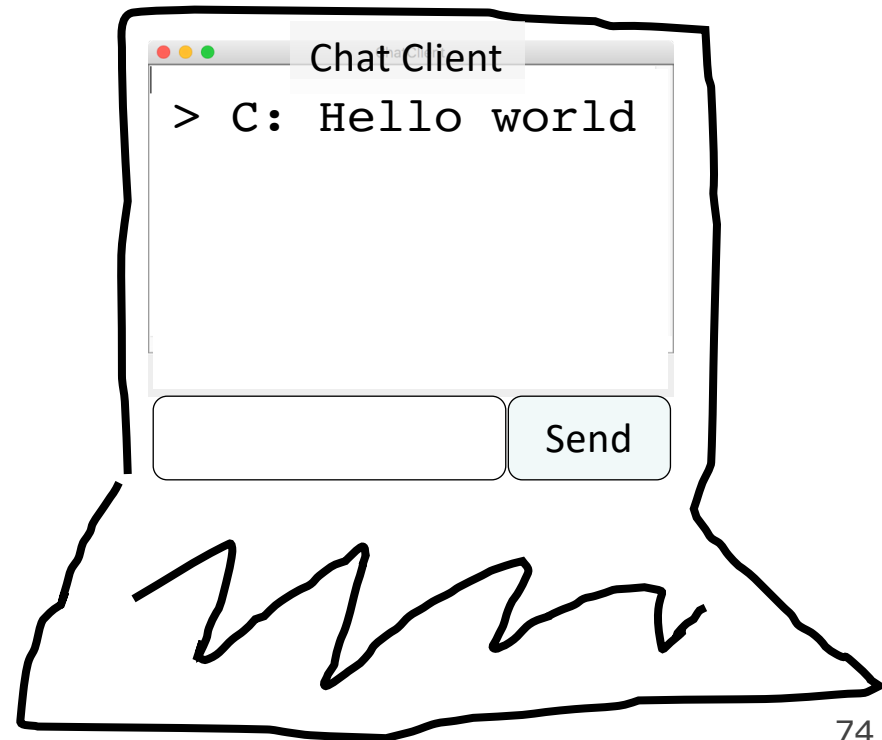
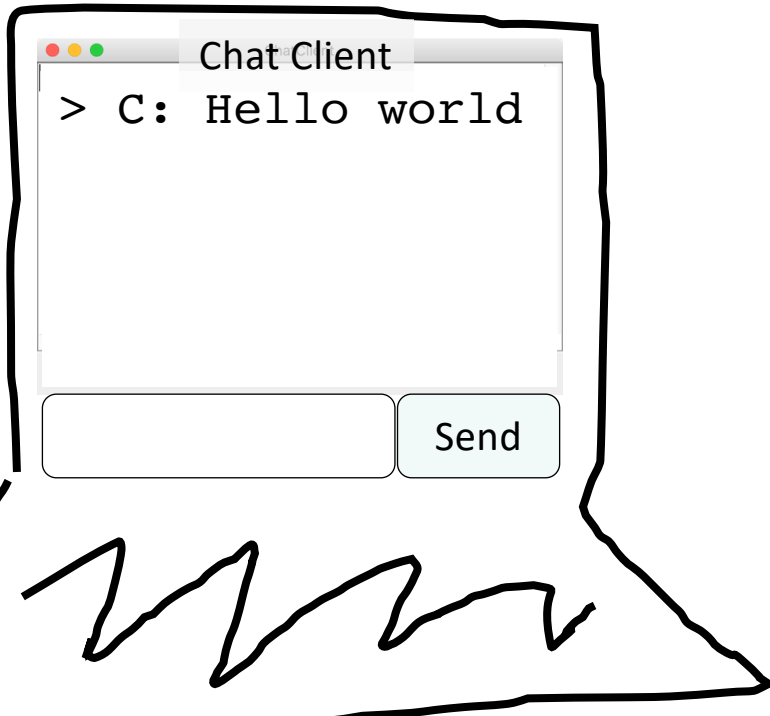
```
addMsg  
msg = B: Im here too
```





```
history = [  
    C: Hello world,  
    B: Im here too  
]
```

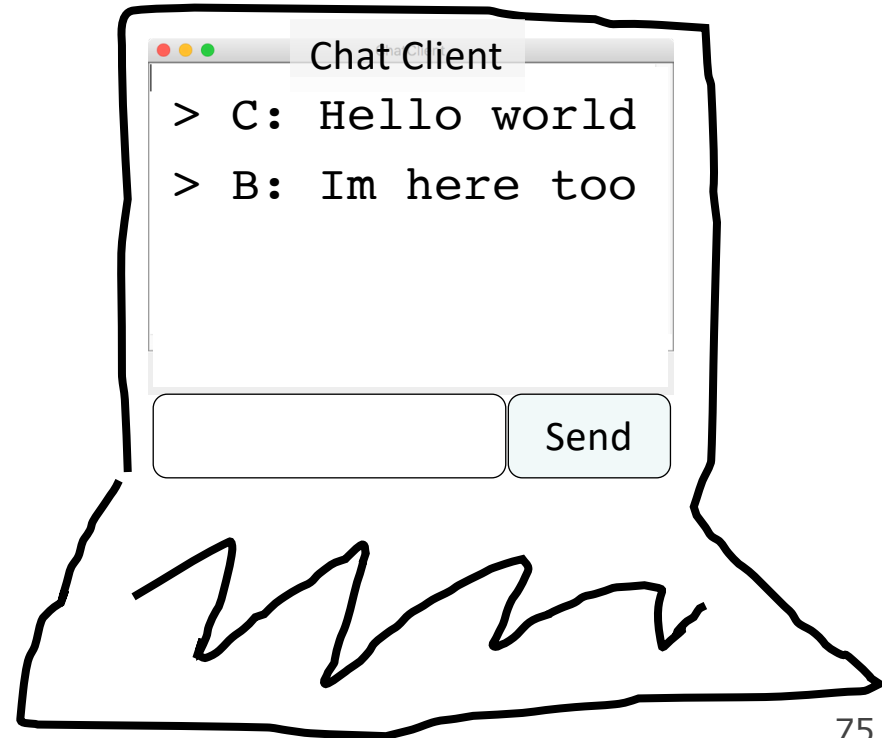
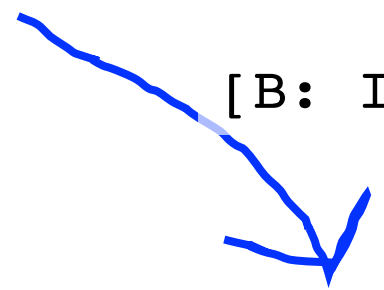
```
getMsgs  
index = 1
```





```
history = [  
    C: Hello world,  
    B: Im here too  
]
```

[B: Im here too]



Chat Server

Chat Server



```
addMsg  
msg = text
```



```
getMsgs  
index = startIndex
```



**Let's Code It
Together!**

Recap

- How do internet programs work?
- Sending Data Over The Internet
- Writing a client and server
- *Demo: echo*
- *Demo: Chat*