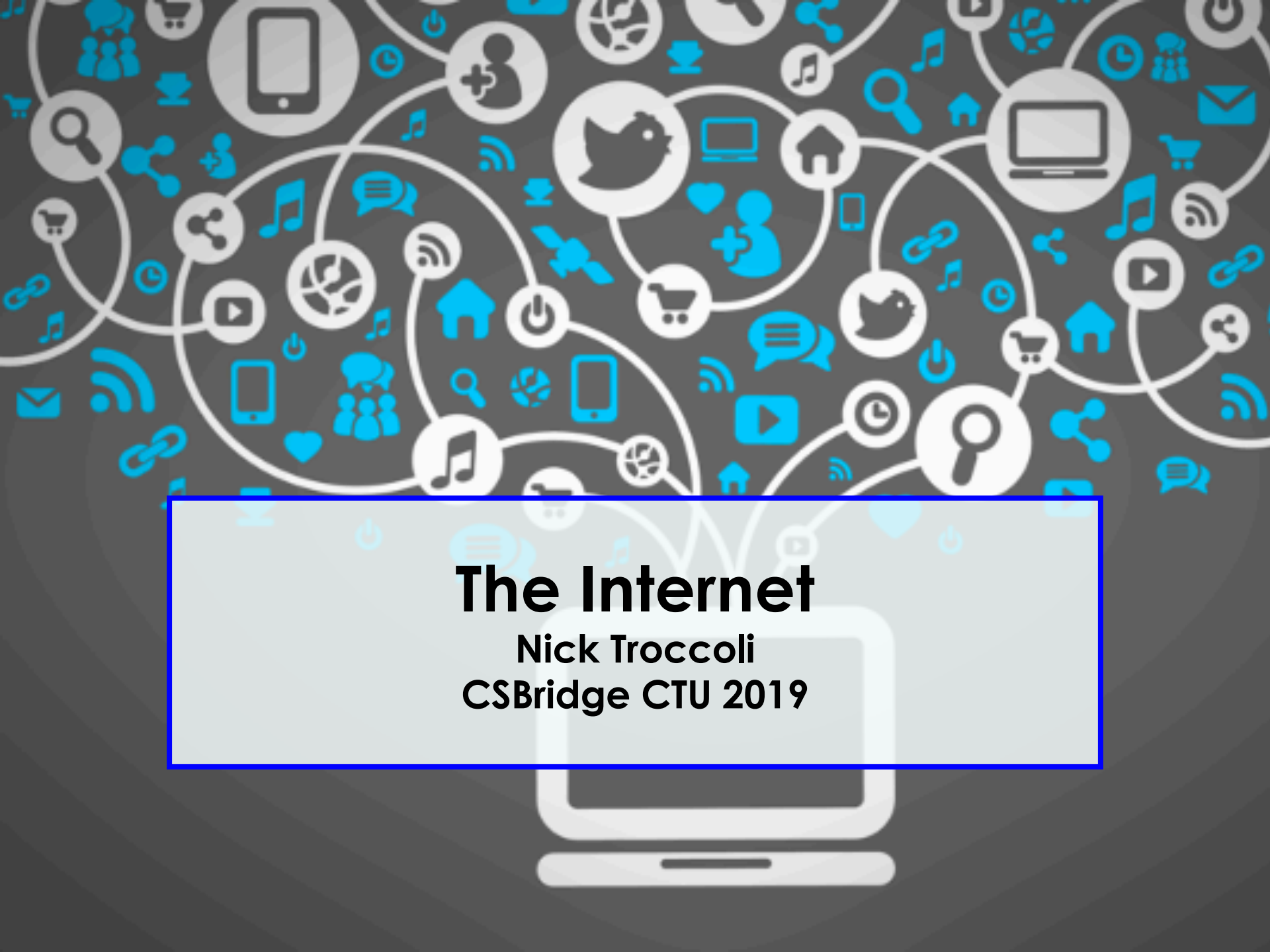


The Internet

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

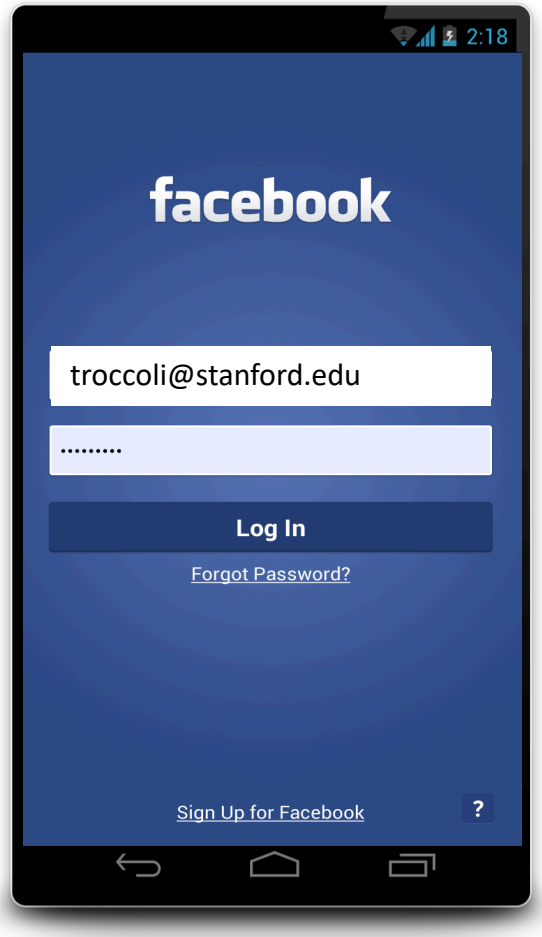
The Java program on your **phone** talks to the Java program at **Facebook**.

Facebook

Facebook Server

* Android phones run Java. So do Facebook servers!

Facebook

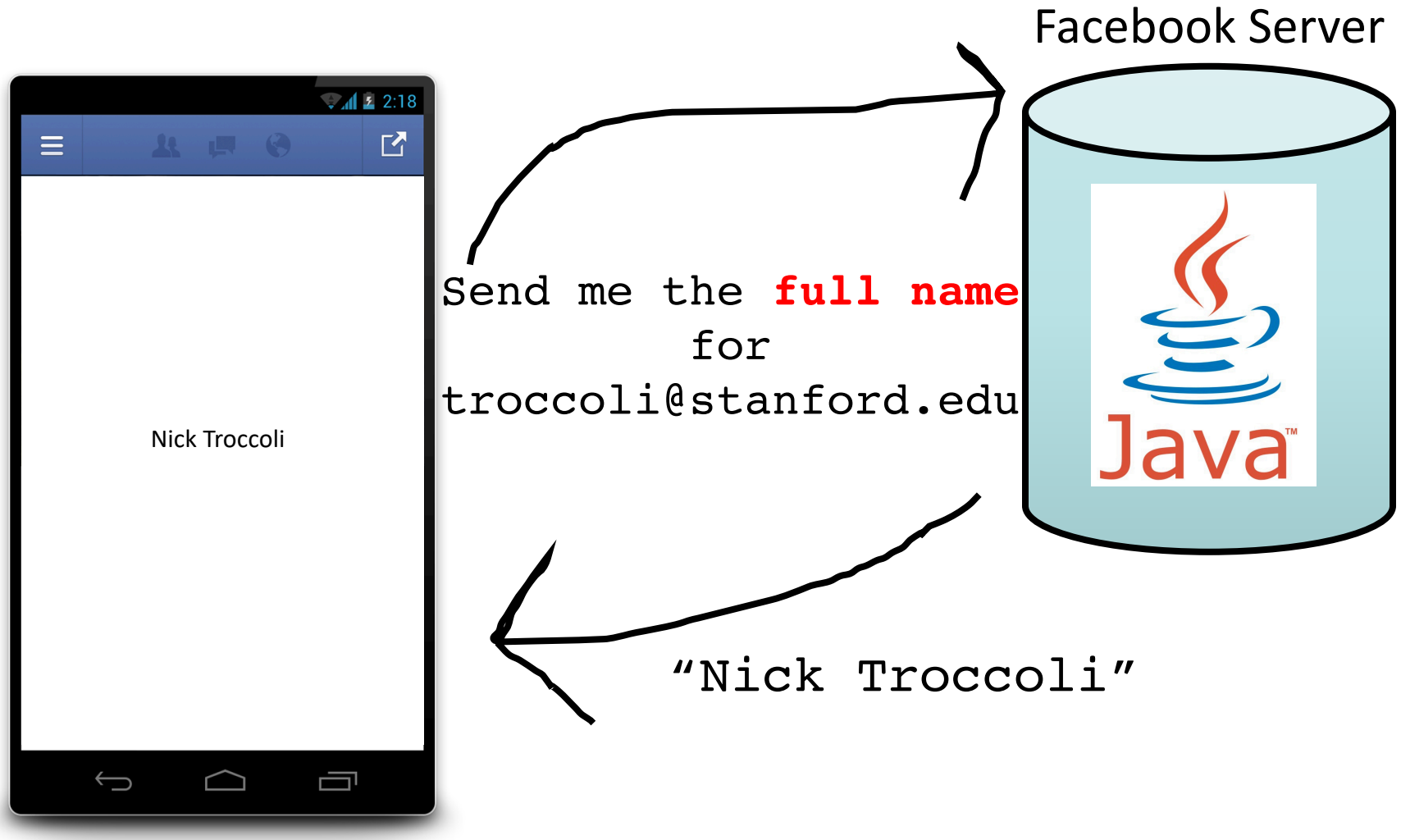


Facebook Server

Is this login ok?

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

Facebook


Facebook Server
2:18
Nick Troccoli
Send me the full name for troccoli@stanford.edu
"Nick Troccoli"
Java

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**.

Programs and the Internet

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

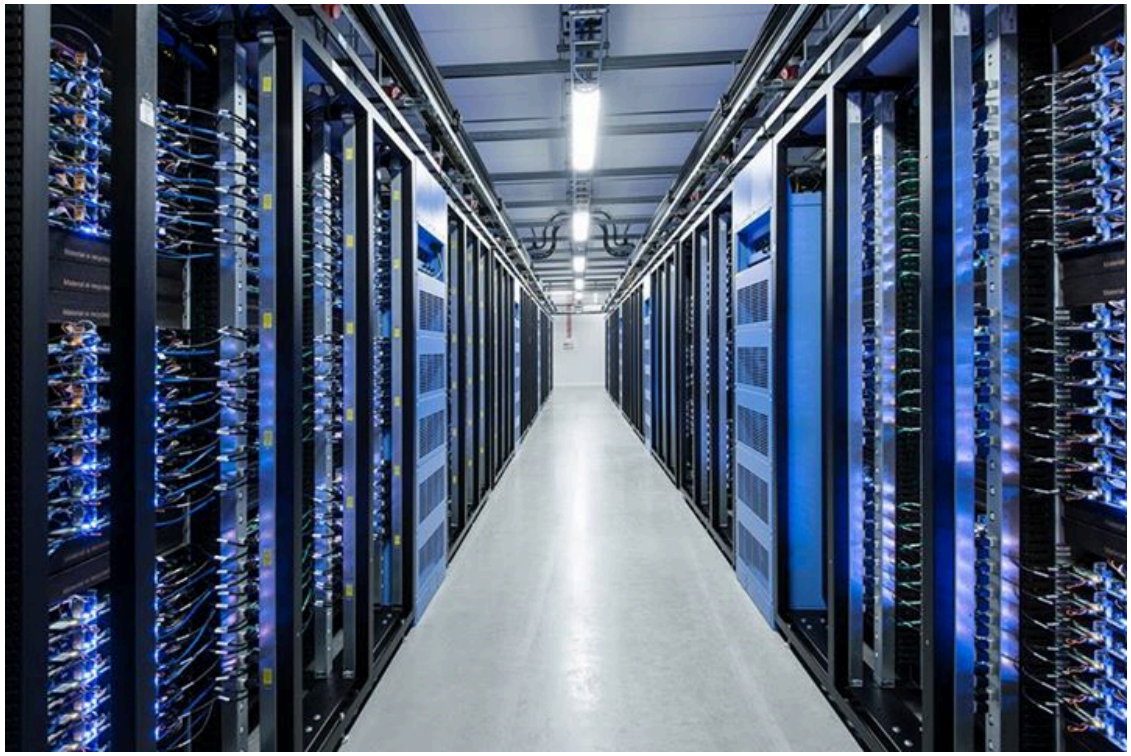
Programs and the Internet

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*)

Thanks Nick for the teaching YEAH

Background: The Internet

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

Thanks Nick for the teaching YEAH

Background: The Internet

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

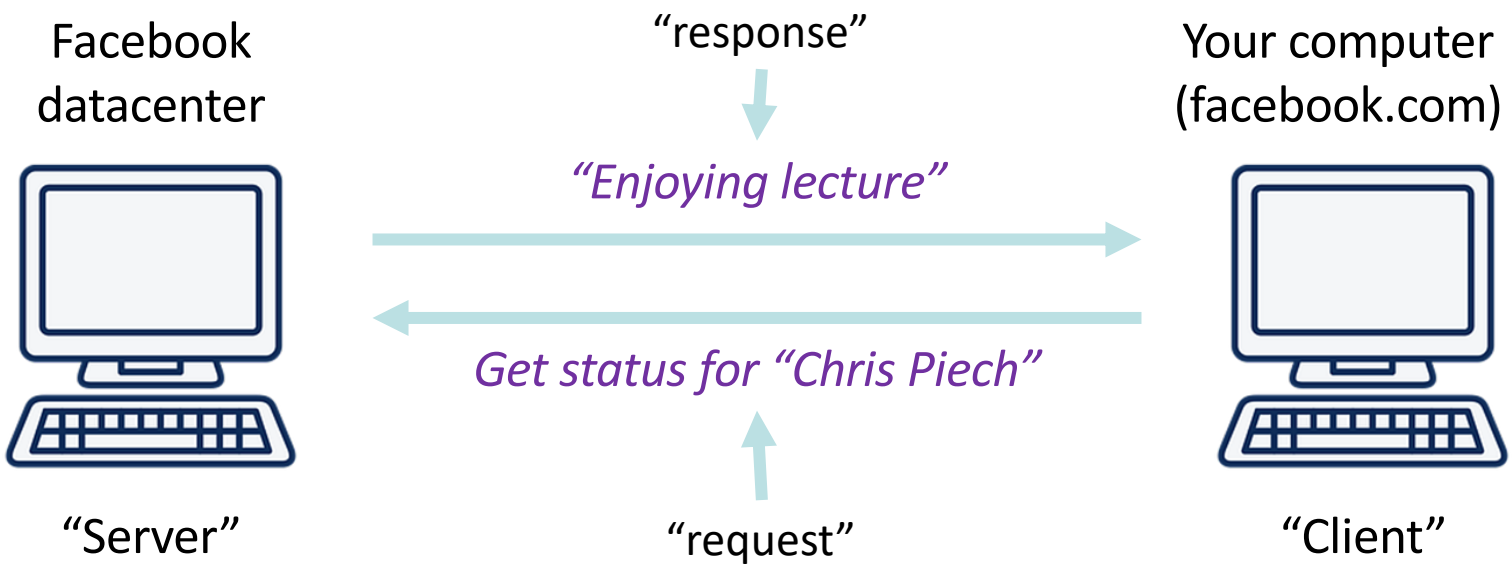
Thanks Nick for the teaching YEAH

Background: The Internet

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

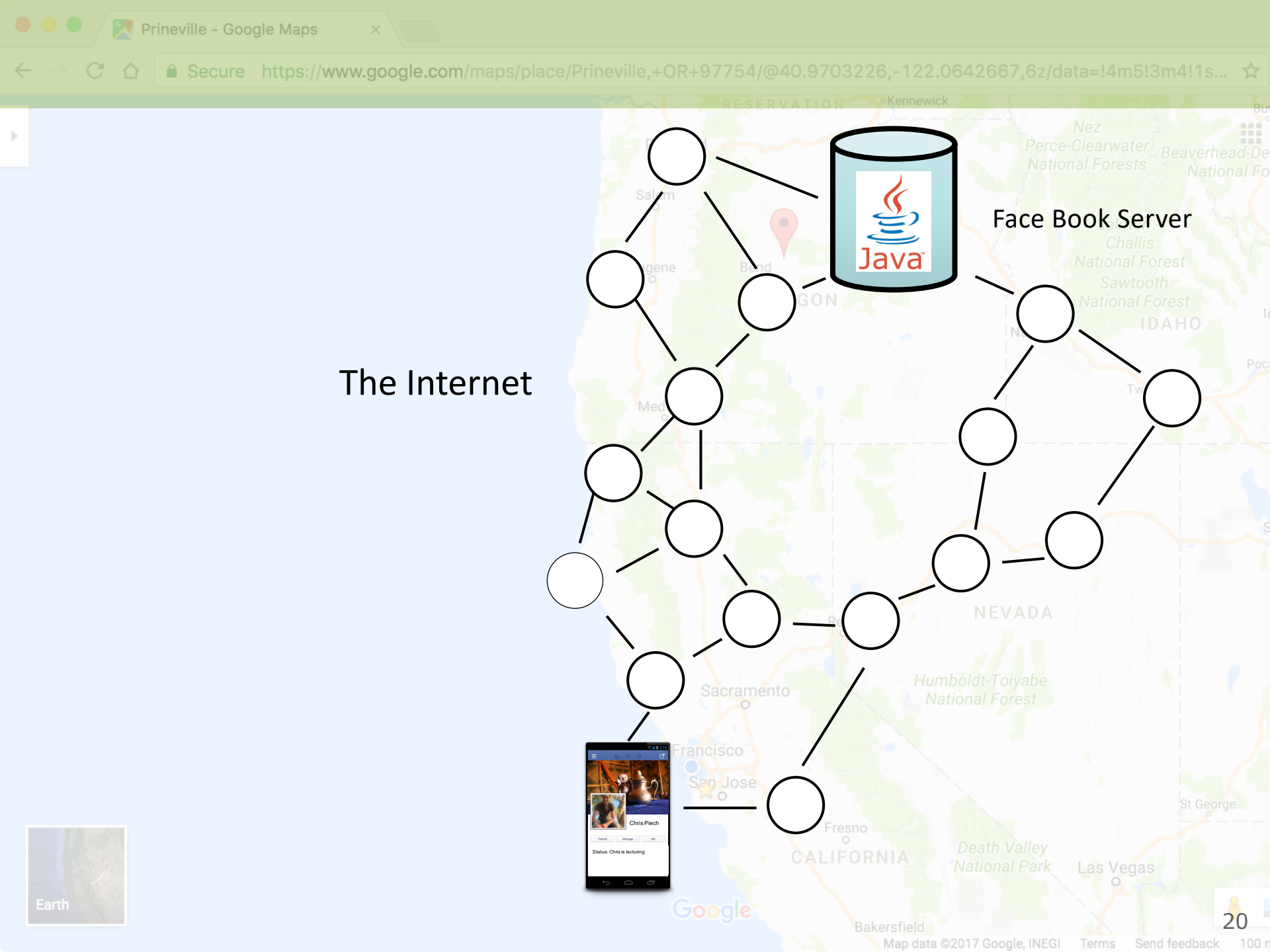
Thanks Nick for the teaching YEAH

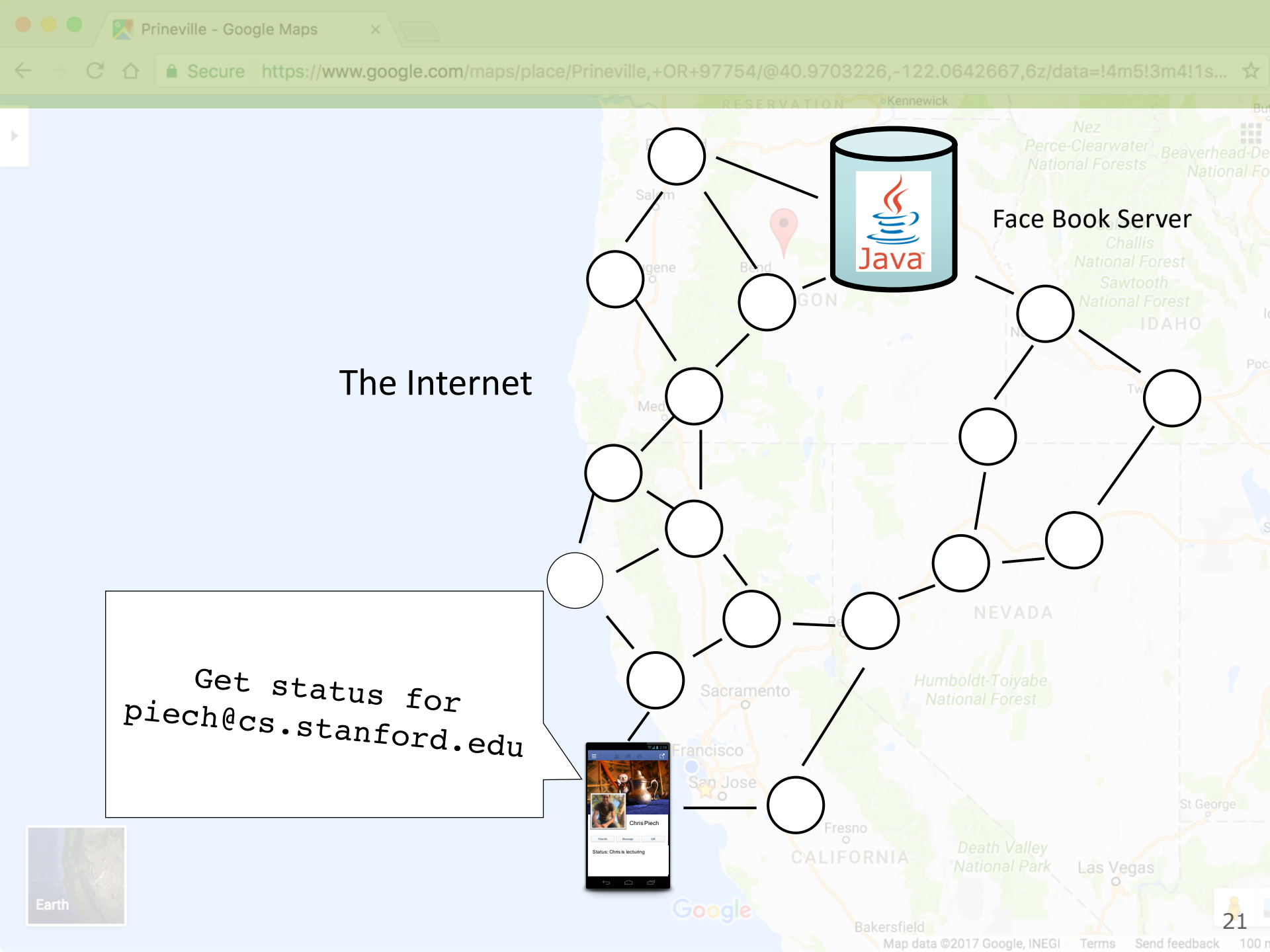
Background: The Internet

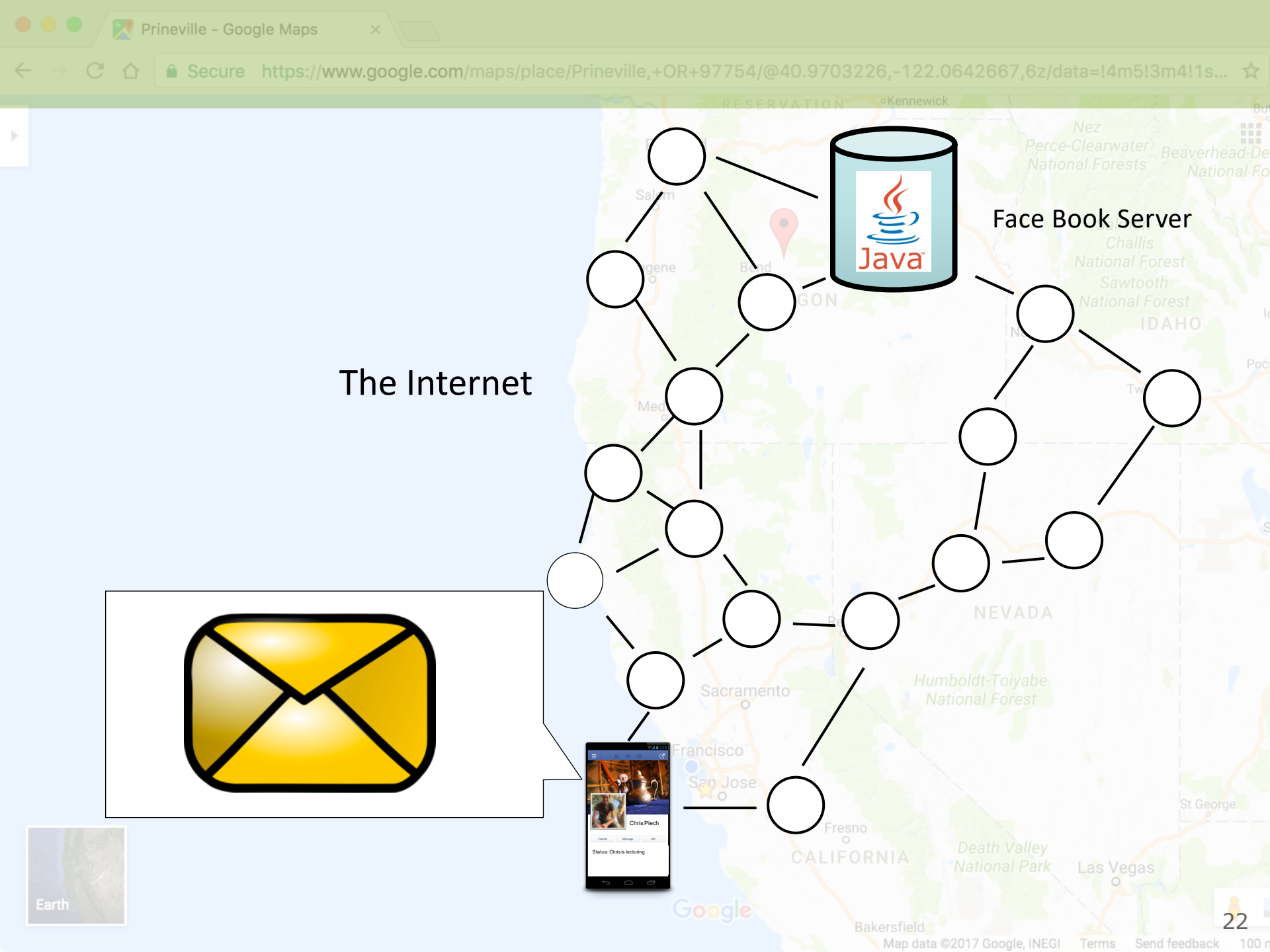


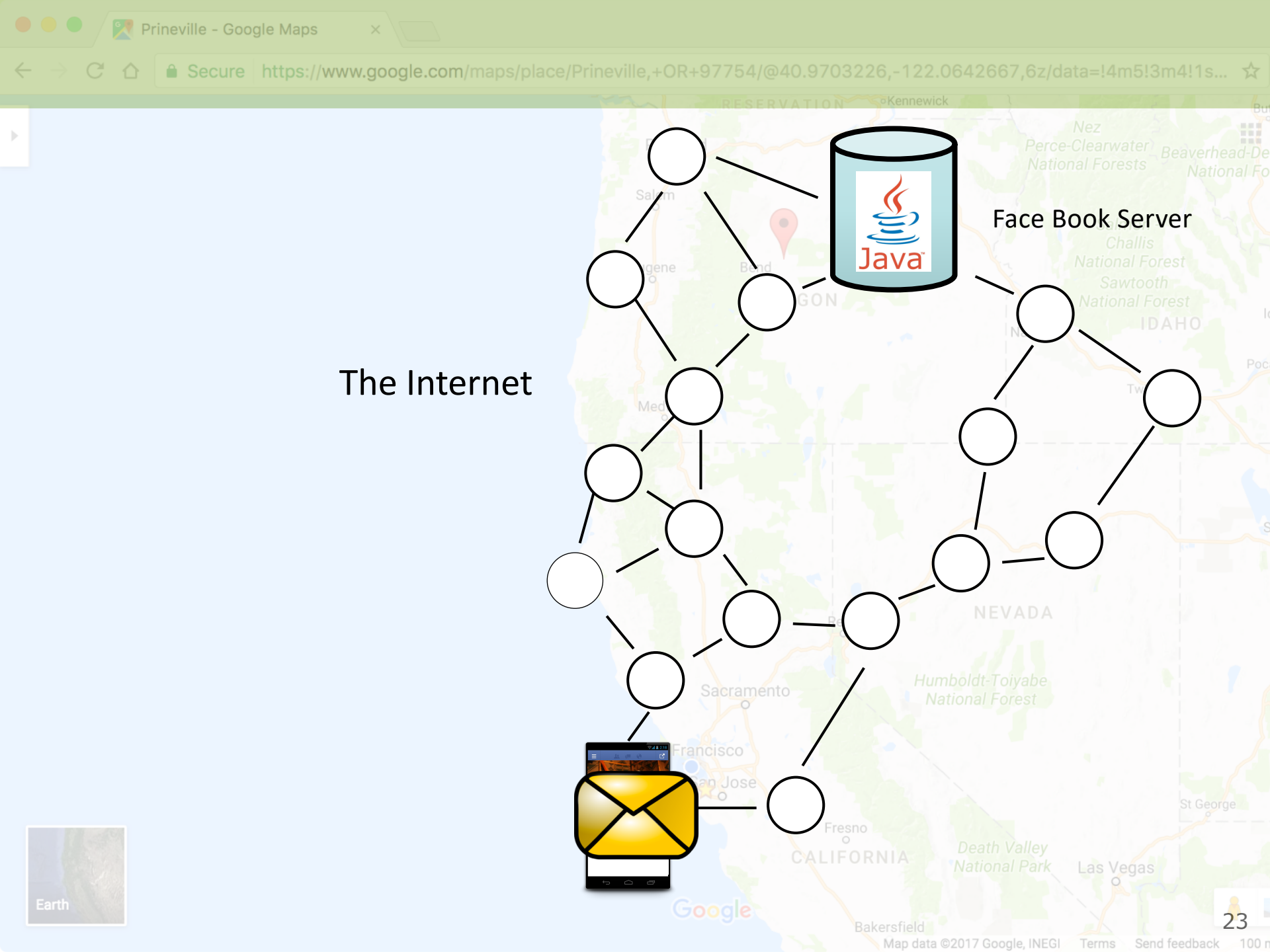
The internet is just many programs sending messages (as *Strings*)

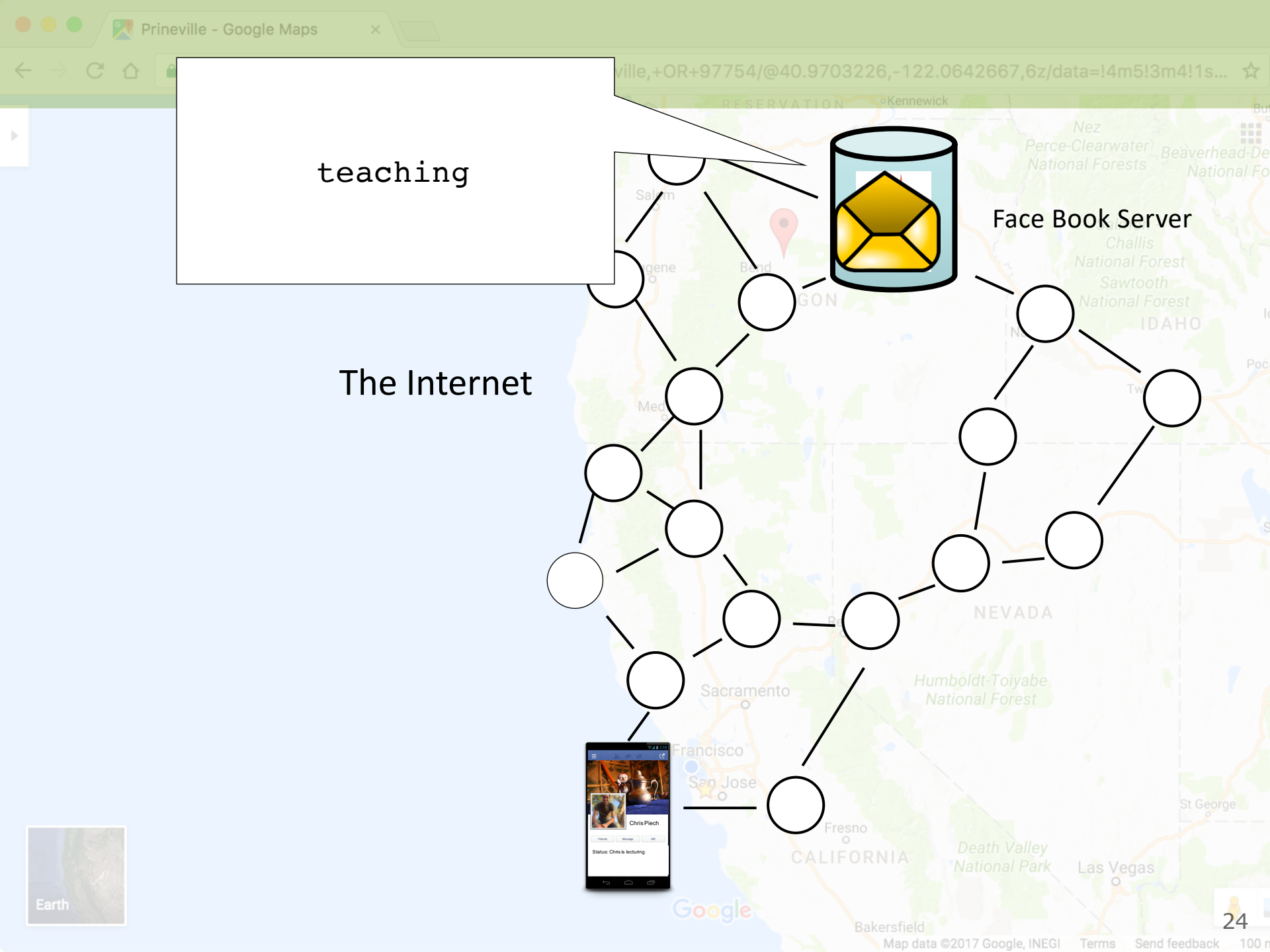
Thanks Nick for the teaching YEAH

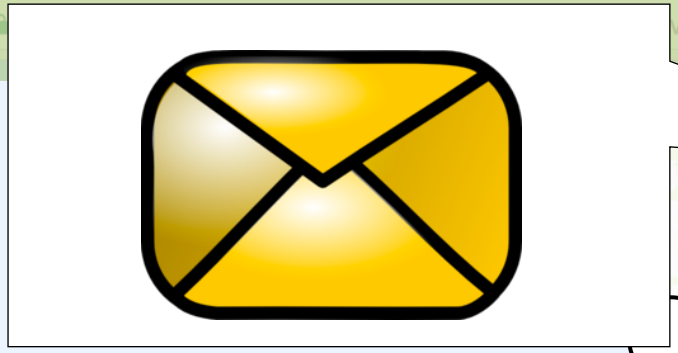

Prineville - Google Maps
Secure
https://www.google.com/maps/place/Prineville,+OR+97754/@40.9703226,-122.0642667,6z/data=!4m5!3m4!1s...
Face Book Server
Java
The Internet
Chris Piech
Status: Chris is lecturing
Earth
Kennewick
Sacramento
San Jose
Fresno
Las Vegas
NEVADA
CALIFORNIA
IDAHO
Google
Map data ©2017 Google, INEGI
Terms
Send feedback


Prineville - Google Maps
Secure
https://www.google.com/maps/place/Prineville,+OR+97754/@40.9703226,-122.0642667,6z/data=!4m5!3m4!1s...
Face Book Server
Java
The Internet
Get status for
piech@cs.stanford.edu
Chris Piech
Status: Chris is lecturing
Earth
Kennewick
Sacramento
San Jose
Fresno
Las Vegas
NEVADA
CALIFORNIA
IDAHO
Google
Map data ©2017 Google, INEGI
Terms
Send feedback


Prineville - Google Maps
Secure
https://www.google.com/maps/place/Prineville,+OR+97754/@40.9703226,-122.0642667,6z/data=!4m5!3m4!1s...
Face Book Server
Java
The Internet
Chris Piech
Status: Chris is lecturing
Kennewick
Sacramento
San Jose
Fresno
Las Vegas
NEVADA
CALIFORNIA
IDAHO
Earth
Google
Map data ©2017 Google, INEGI
Terms
Send feedback

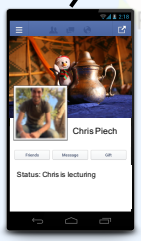

Prineville - Google Maps
Secure
https://www.google.com/maps/place/Prineville,+OR+97754/@40.9703226,-122.0642667,6z/data=!4m5!3m4!1s...
Face Book Server
Java
The Internet
Kennewick
Nez Perce-Clearwater National Forests
Challis National Forest
Sawtooth National Forest
IDAHO
NEVADA
Sacramento
Humboldt-Toiyabe National Forest
Francisco
Fresno
CALIFORNIA
Death Valley National Park
Las Vegas
St George
Bakersfield
Earth
Google
Map data ©2017 Google, INEGI
Terms
Send feedback

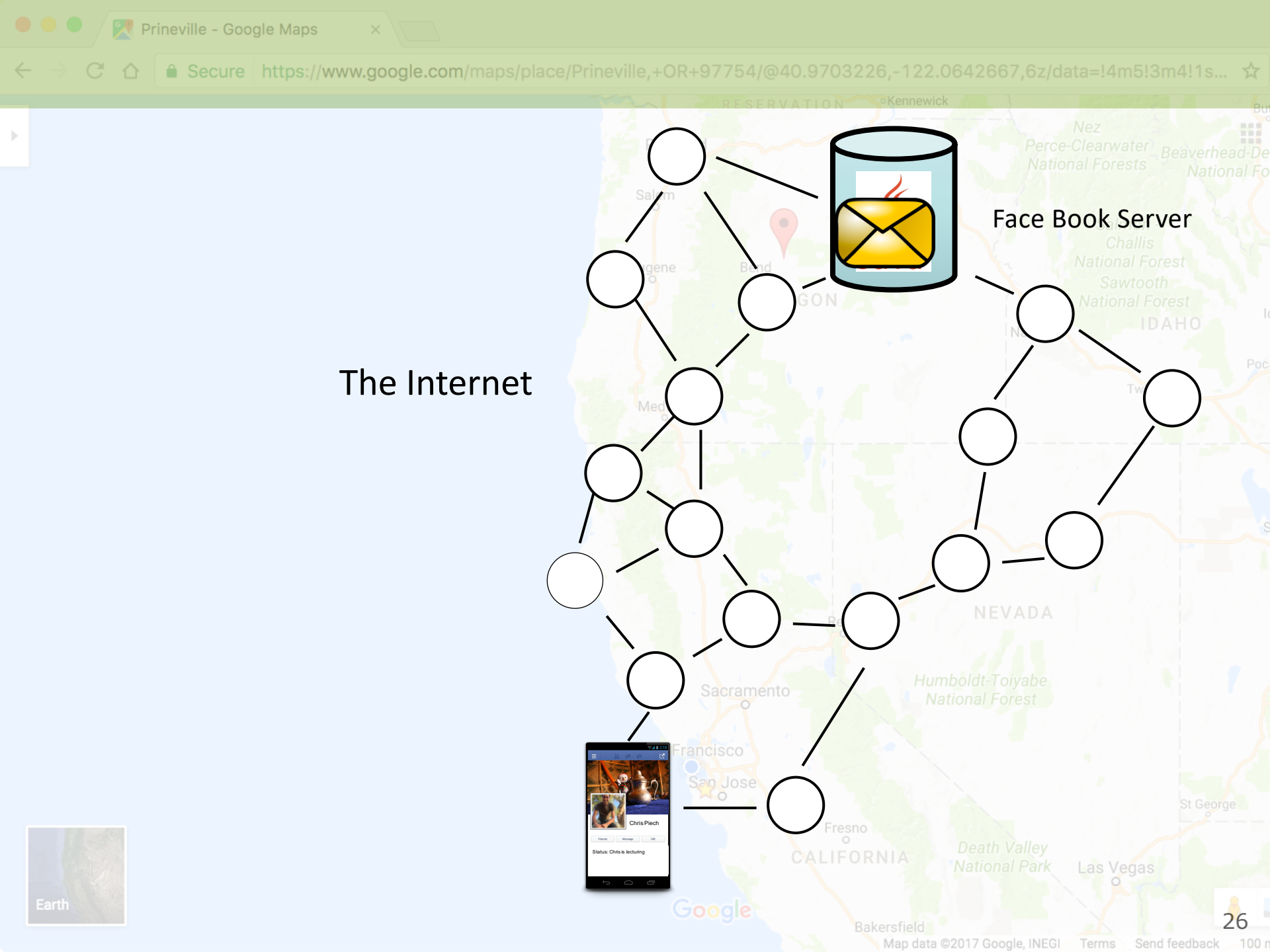

Prineville - Google Maps
teaching
Face Book Server
The Internet
Chris Piech
Status: Chris is lecturing
Earth
Google
Map data ©2017 Google, INEGI
Terms
Send feedback



Face Book Server

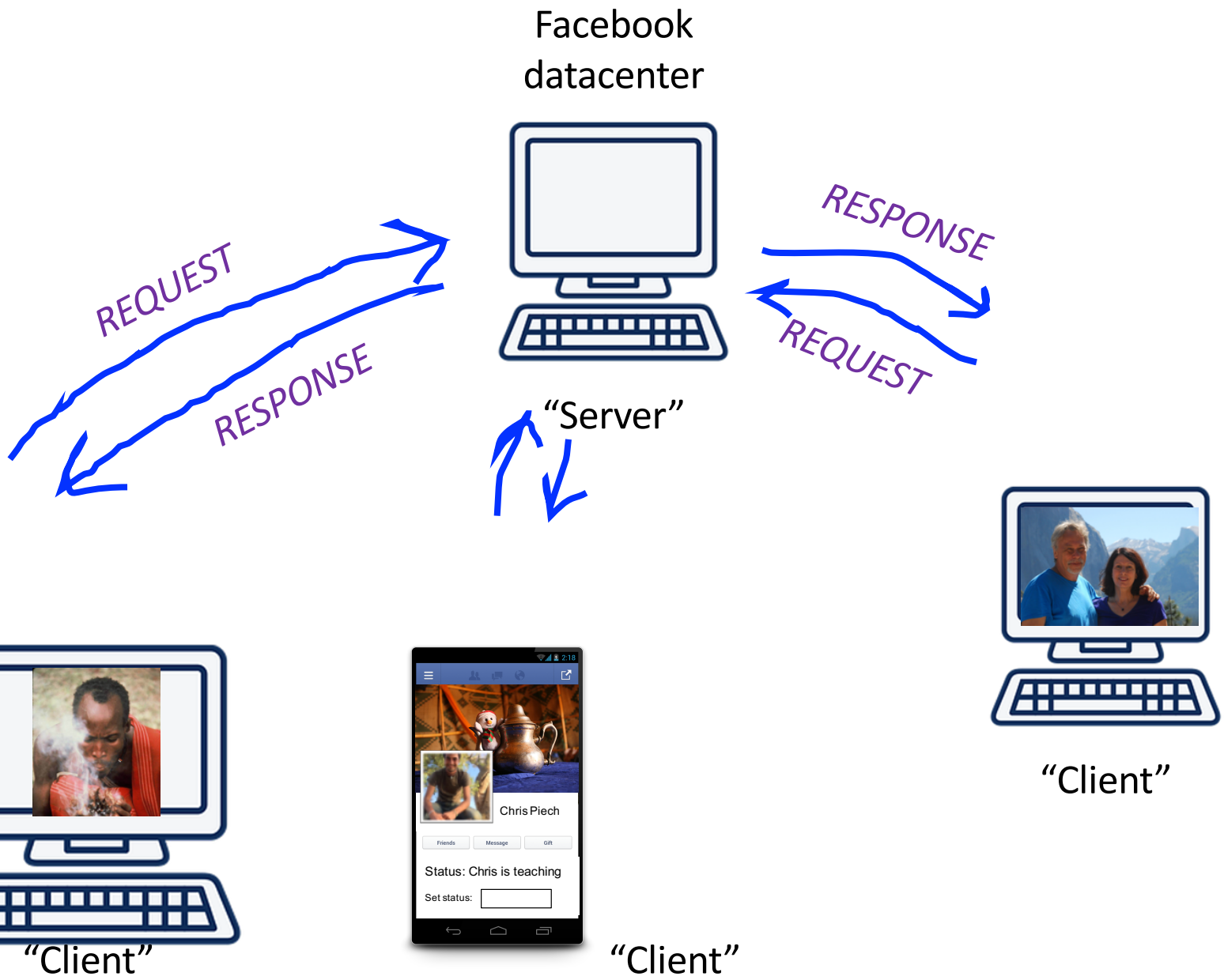
The Internet



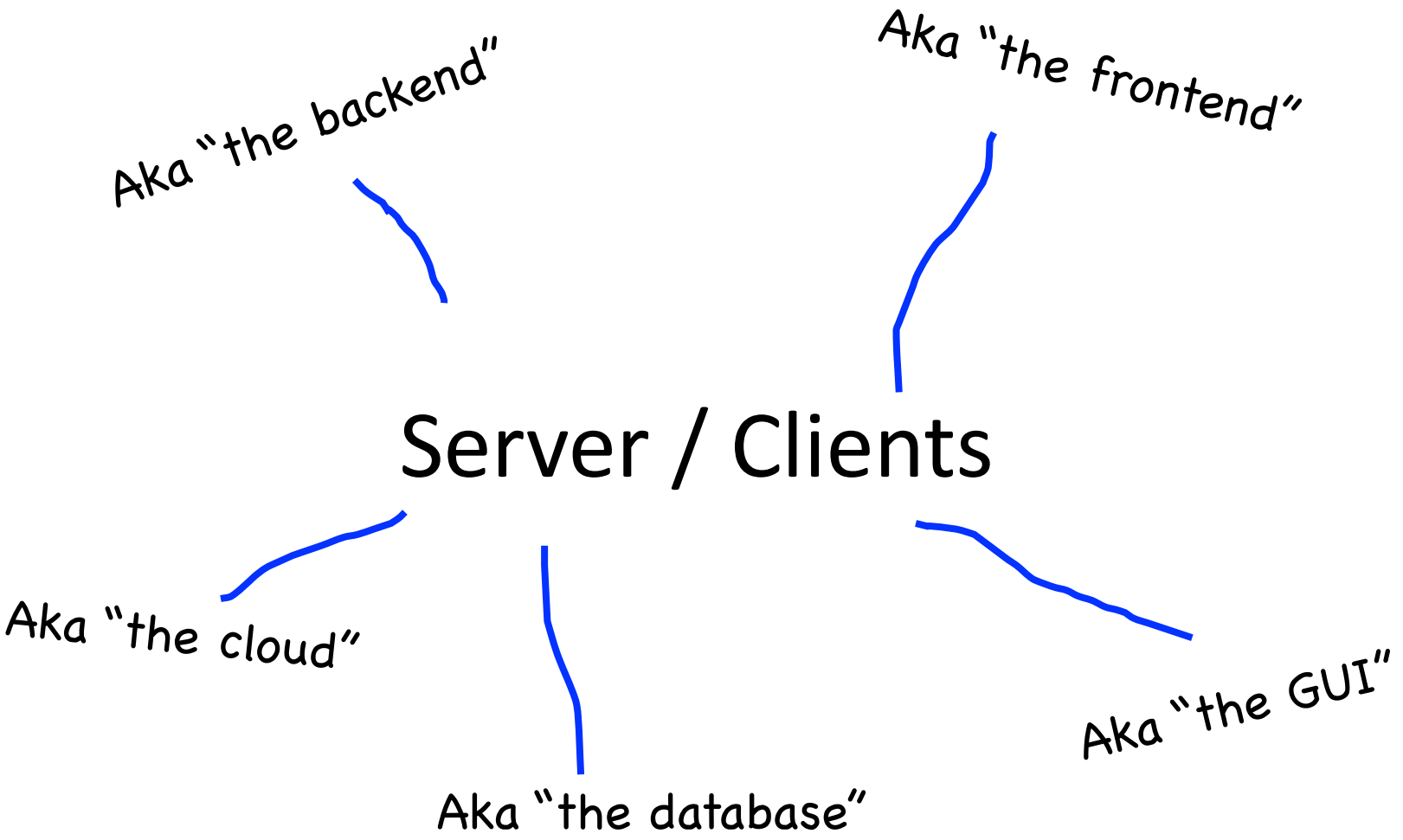

Prineville - Google Maps
Secure
https://www.google.com/maps/place/Prineville,+OR+97754/@40.9703226,-122.0642667,6z/data=!4m5!3m4!1s...
Face Book Server
The Internet
Chris Piech
Status: Chris is lecturing
Earth
Kennewick
Sacramento
San Jose
Fresno
Las Vegas
NEVADA
CALIFORNIA
IDAHO
Bakersfield
Map data ©2017 Google, INEGI
Terms
Send feedback

Many programs can connect
to the same server

The Internet

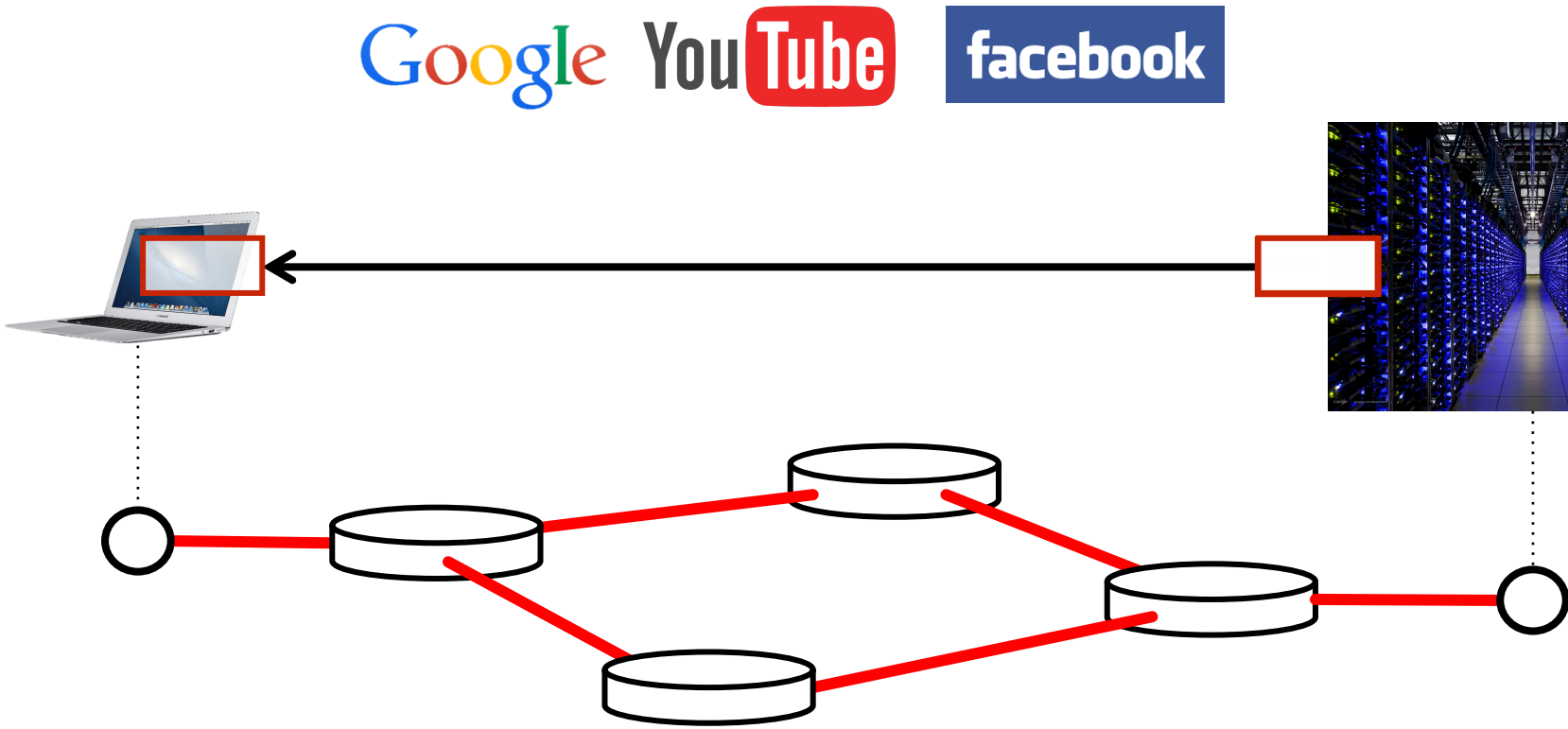


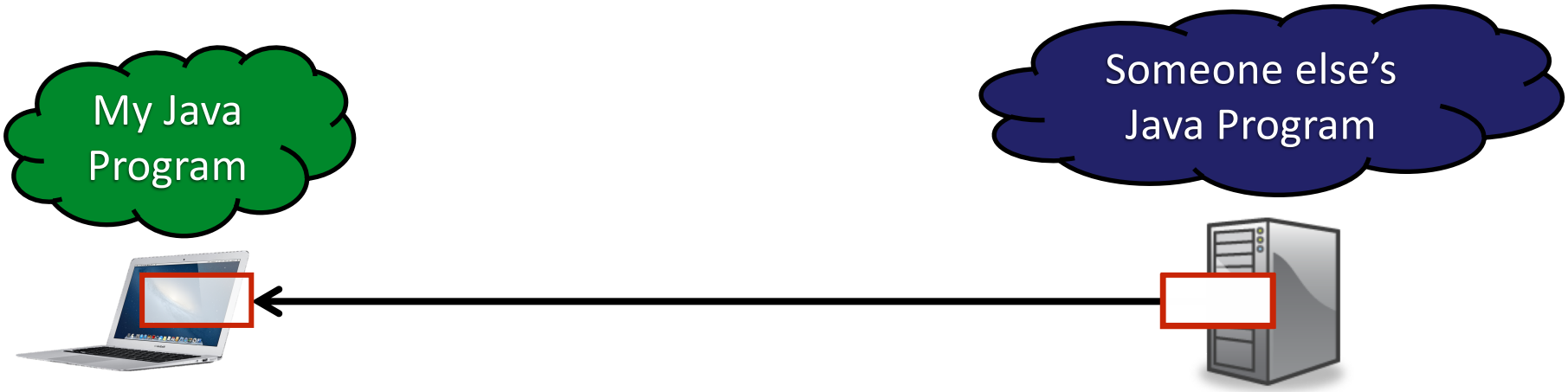
Most of the Internet

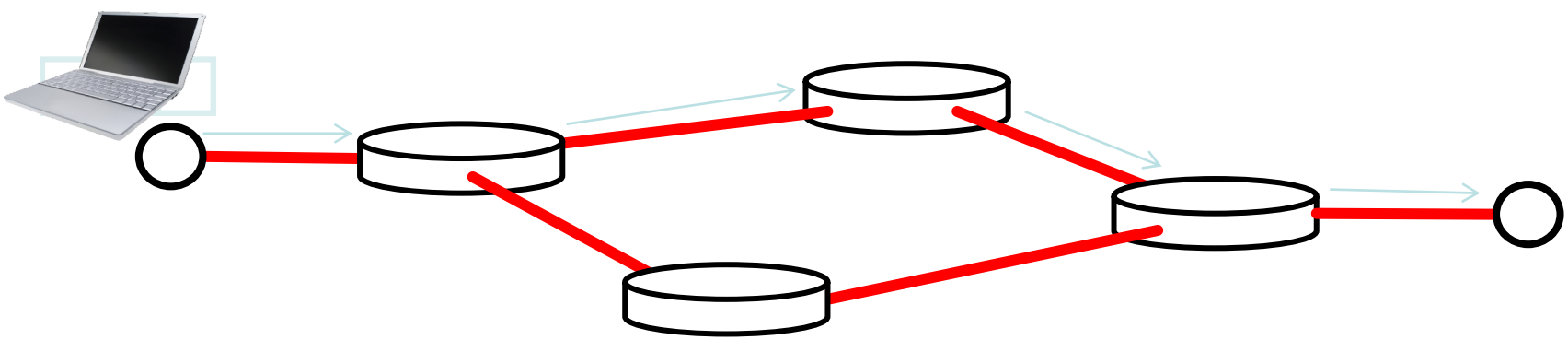


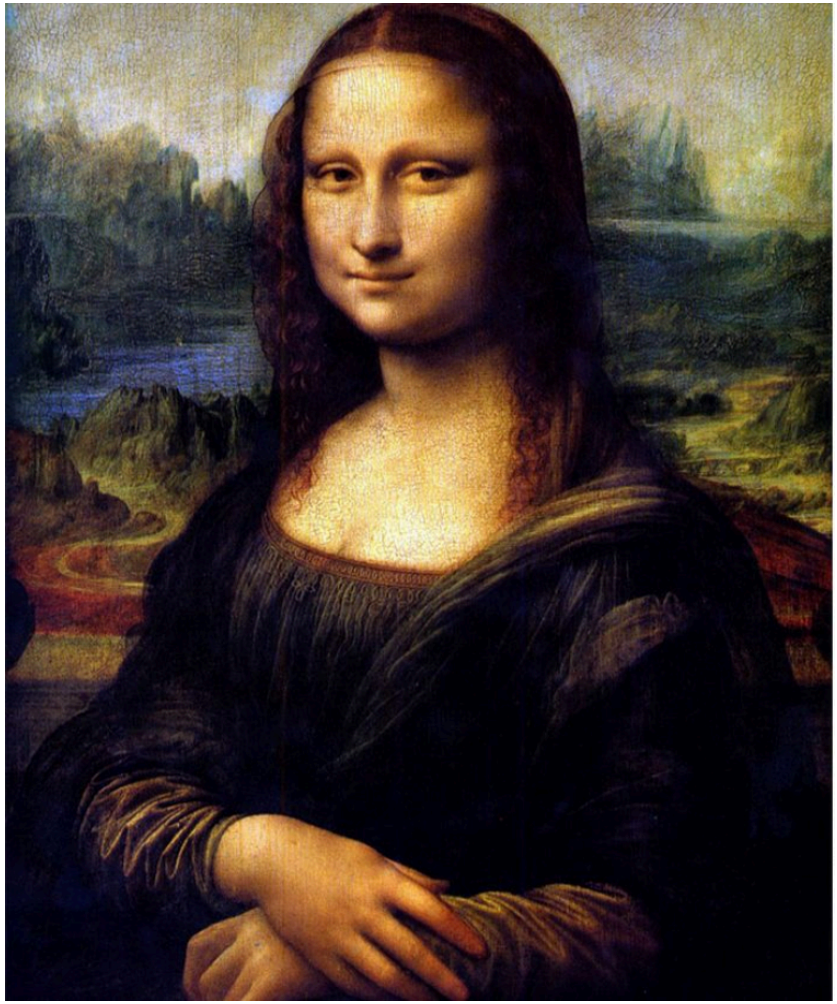
Plan For Today

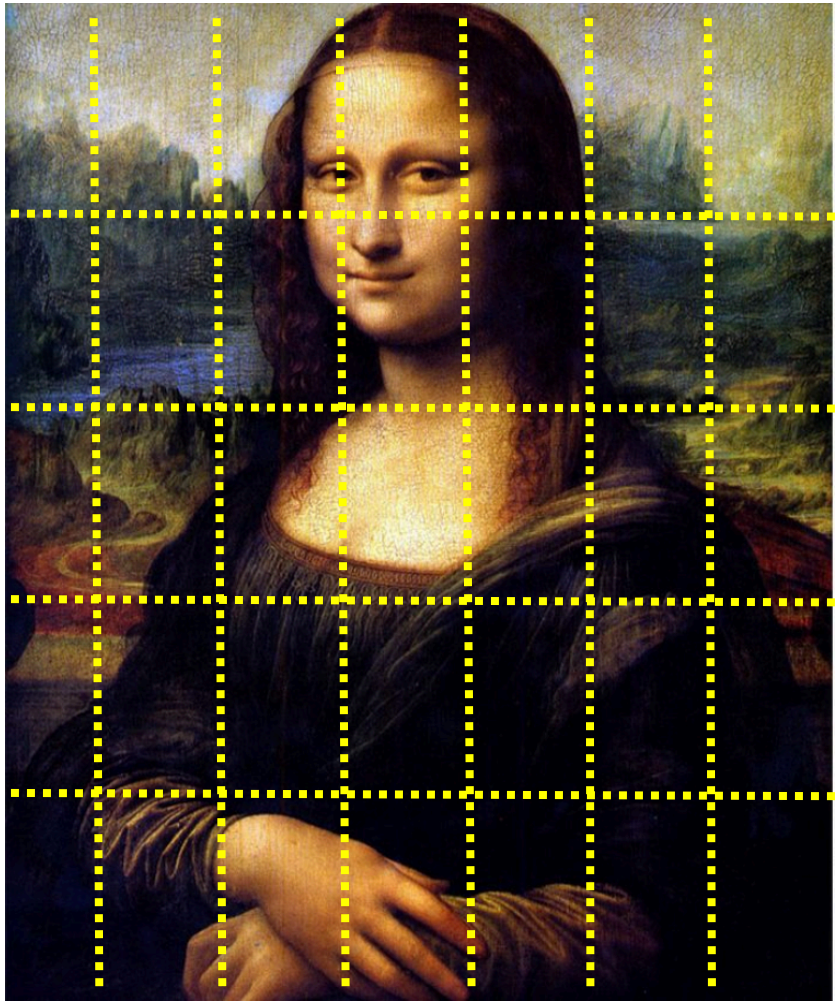
- How do internet programs work?
- **Sending Data Over The Internet**
- Writing a client and server
- *Demo:* echo
- *Demo:* Chat

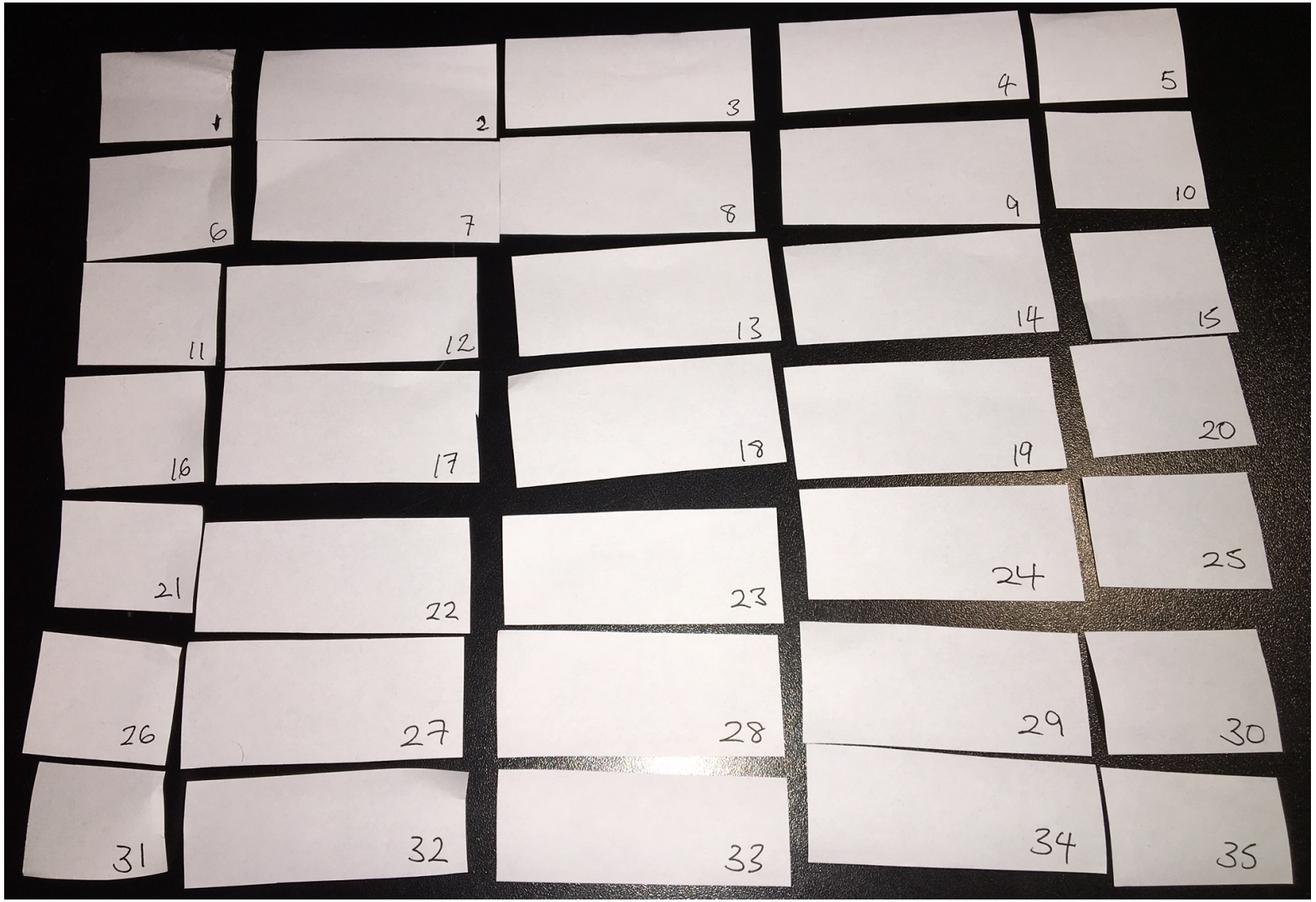

Google
YouTube
facebook


My Java Program
Someone else's Java Program

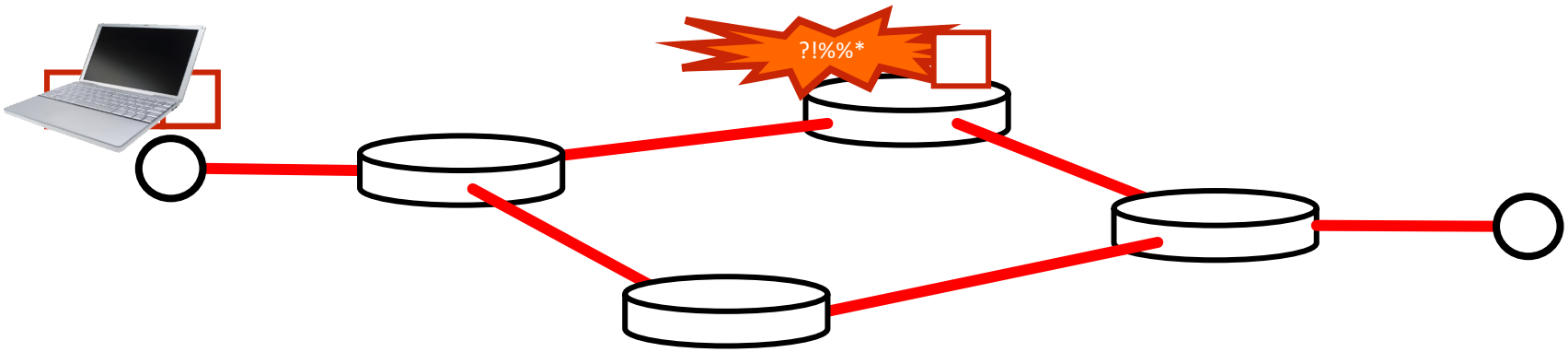




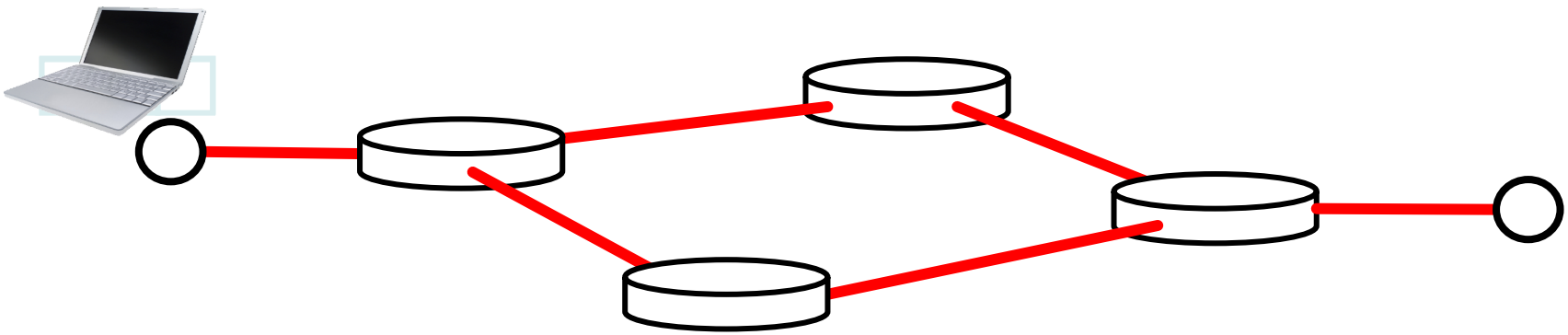



1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35

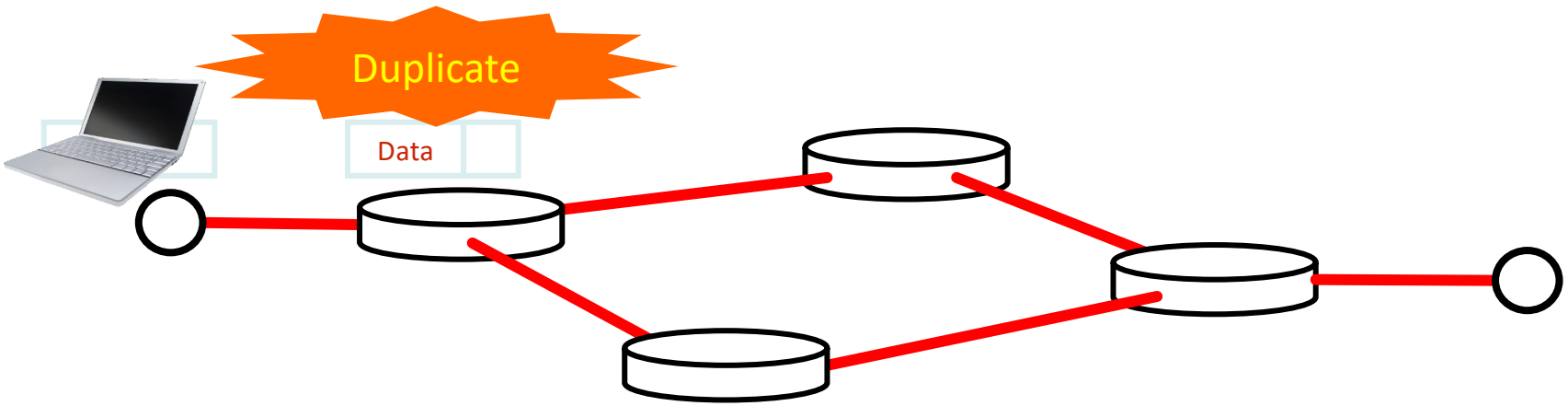
Packets may be damaged



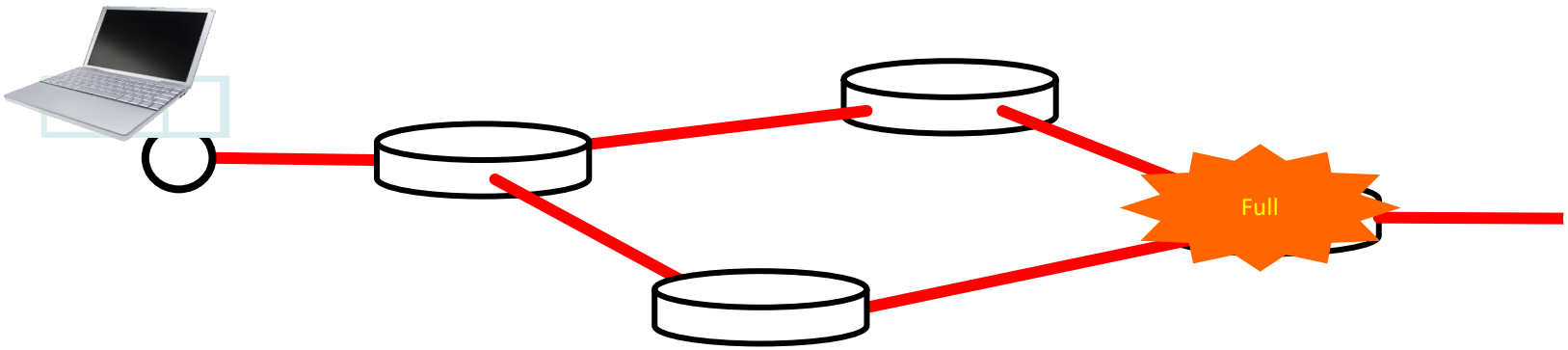
Packets may arrive out of order



Packets may be duplicated

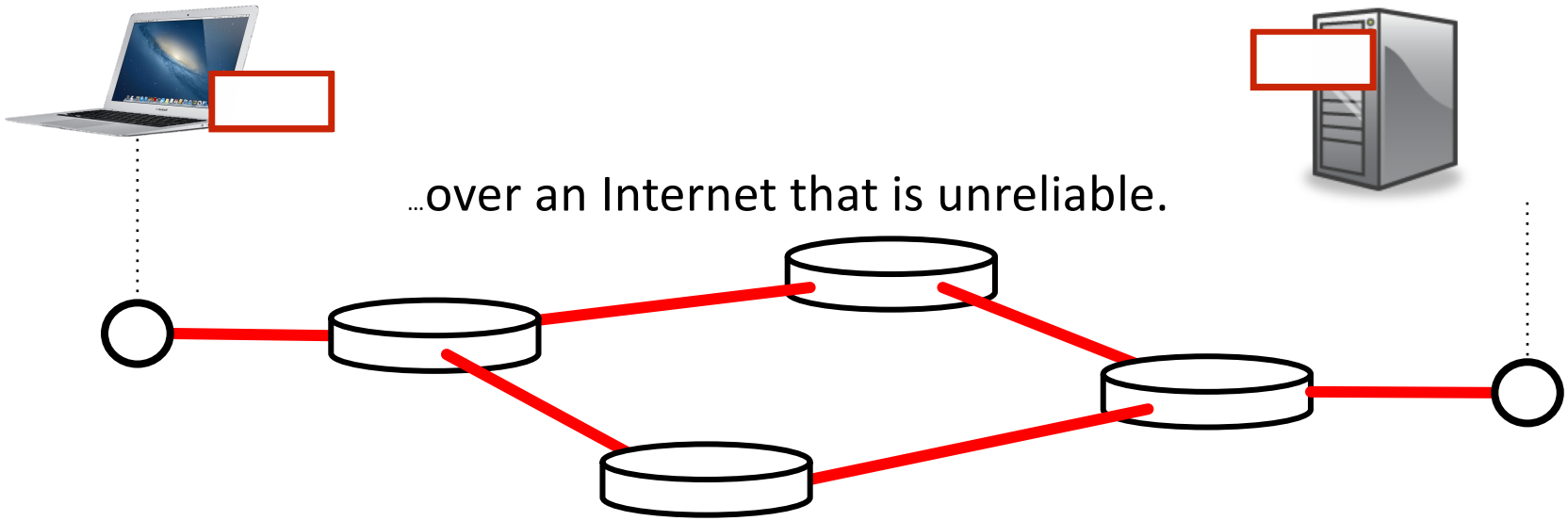


They may not arrive at all!



The Internet Is Unreliable

Applications send and receive data in packets….



…over an Internet that is unreliable.

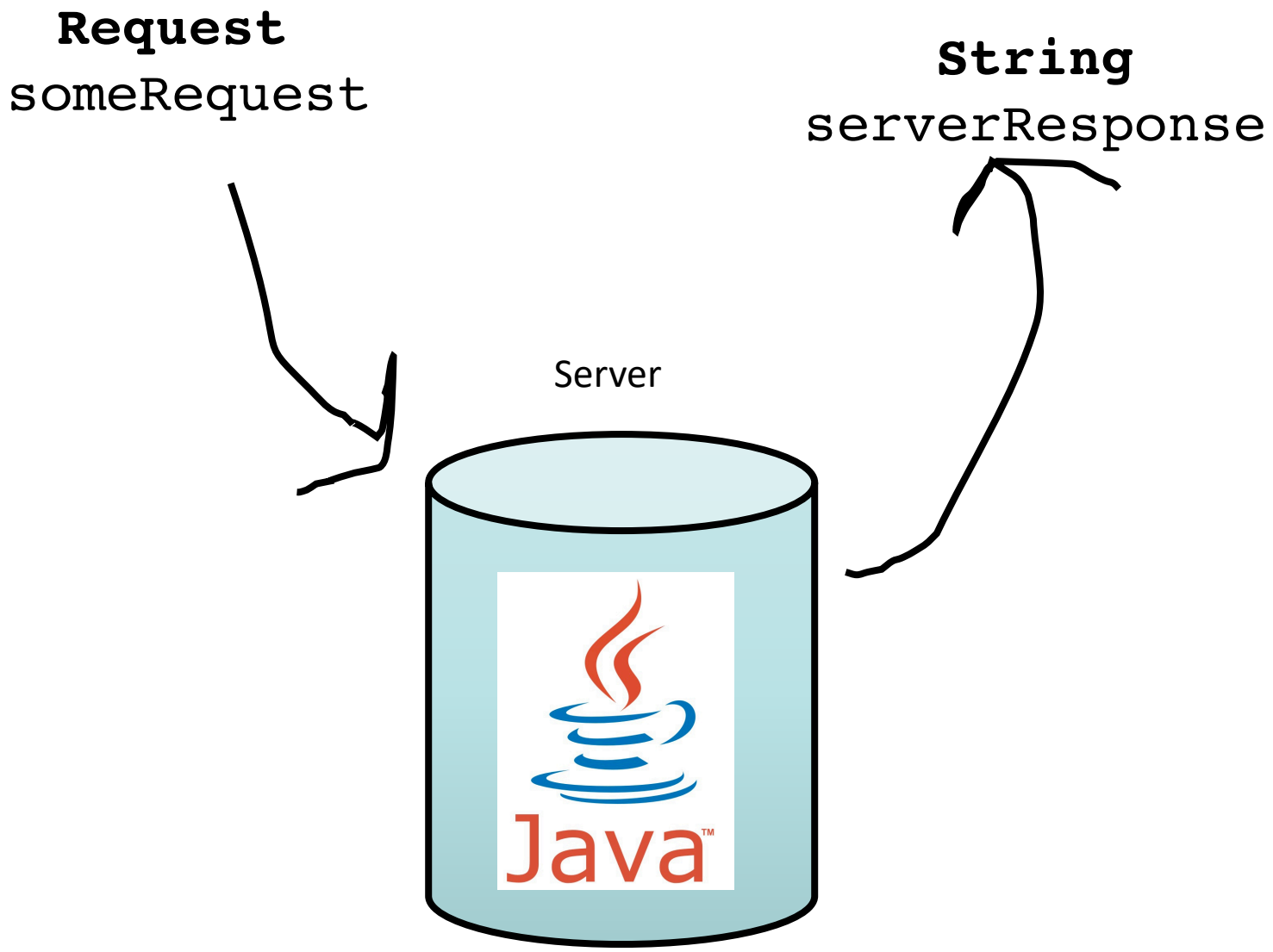
Software on the devices themselves compensates for this.

Let's send a message to Chris 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



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
}
```

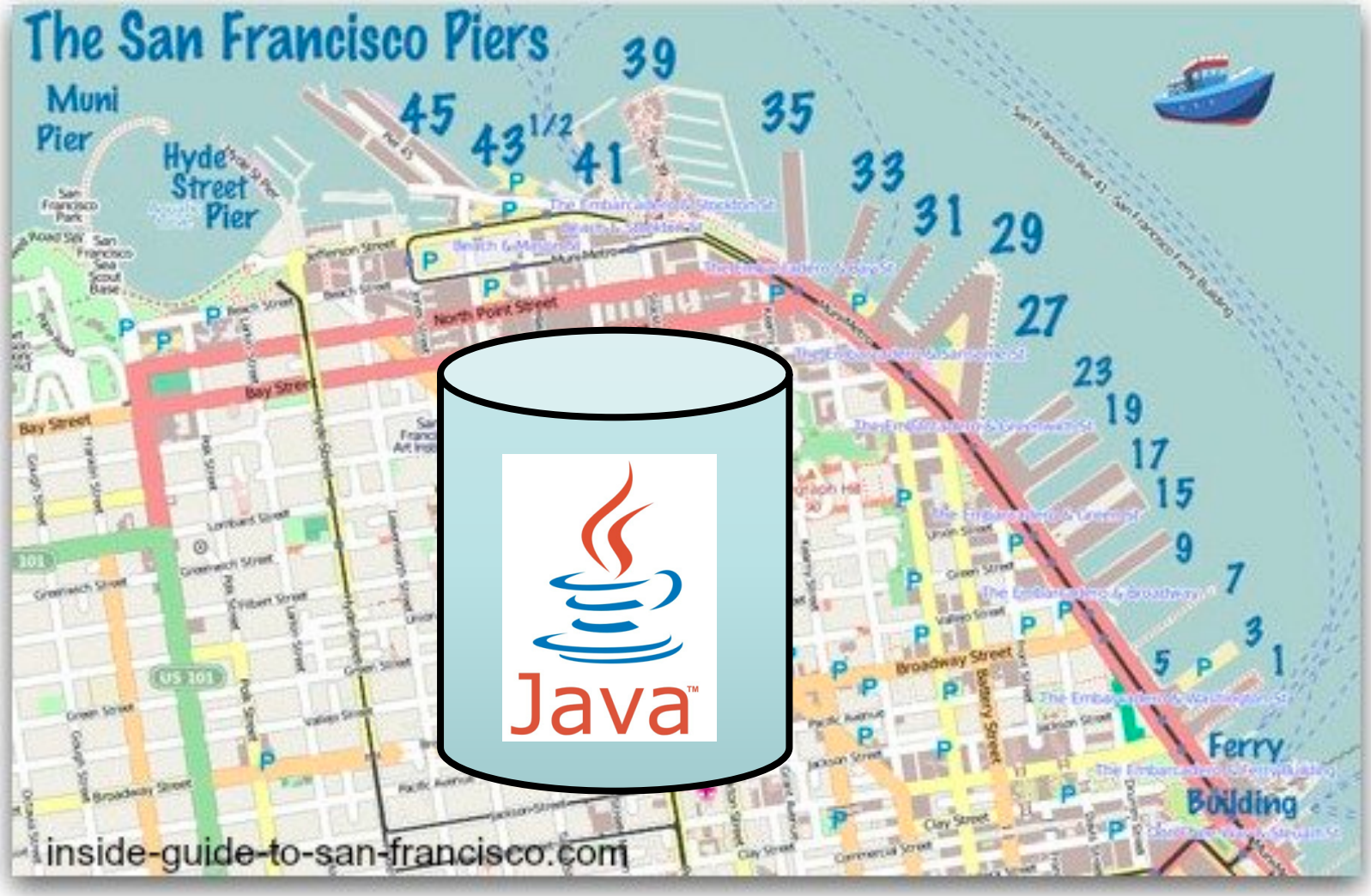
2

```
// 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
}
```

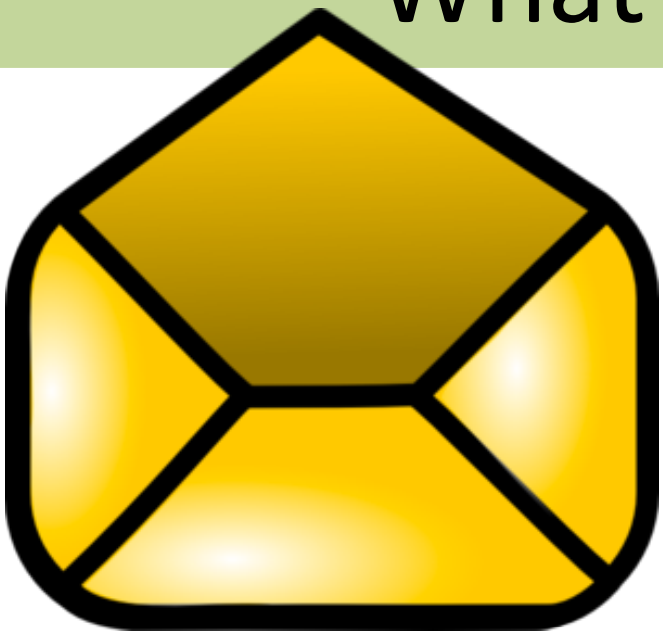
2

```
// 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.";
    }

}
```

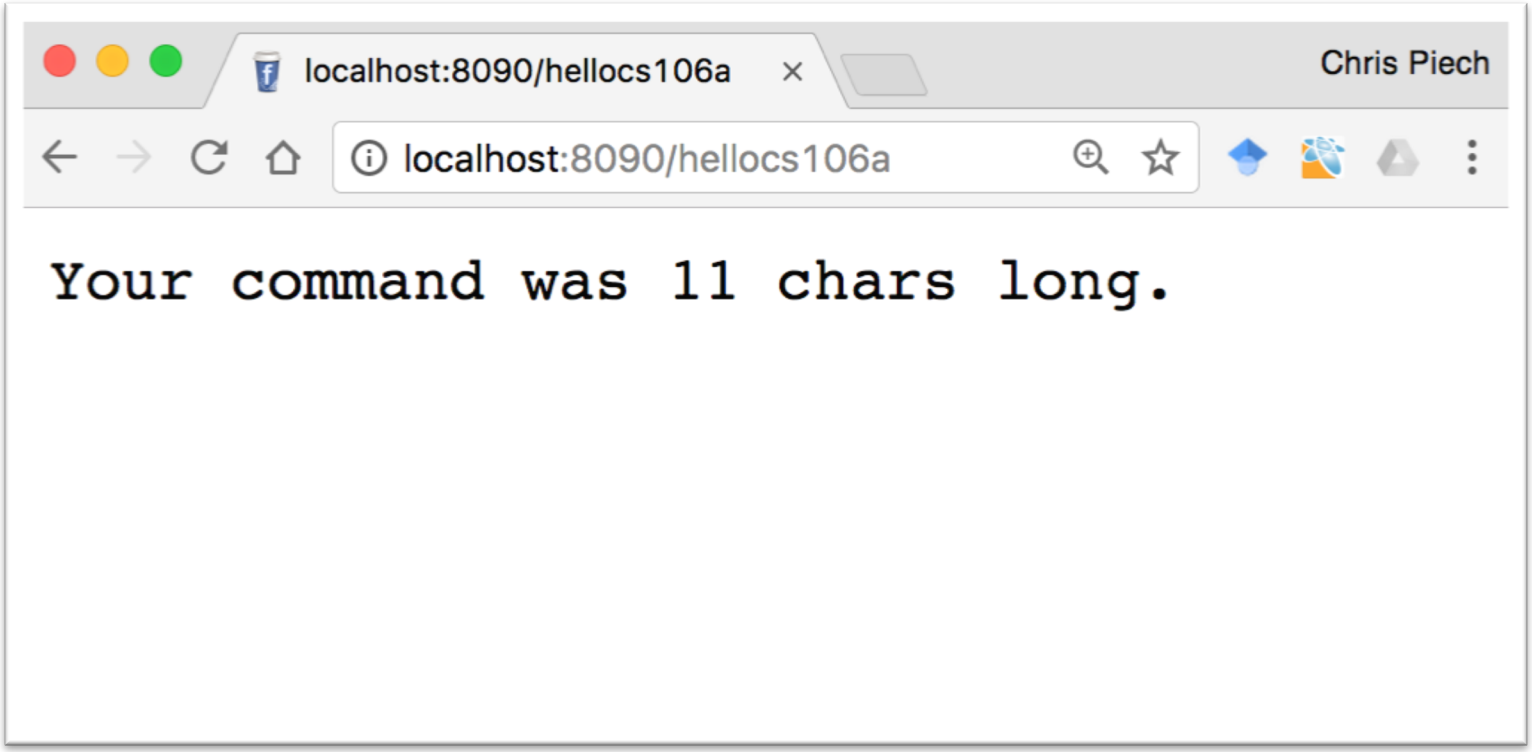
Our First Server

```
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.";
    }

}
```

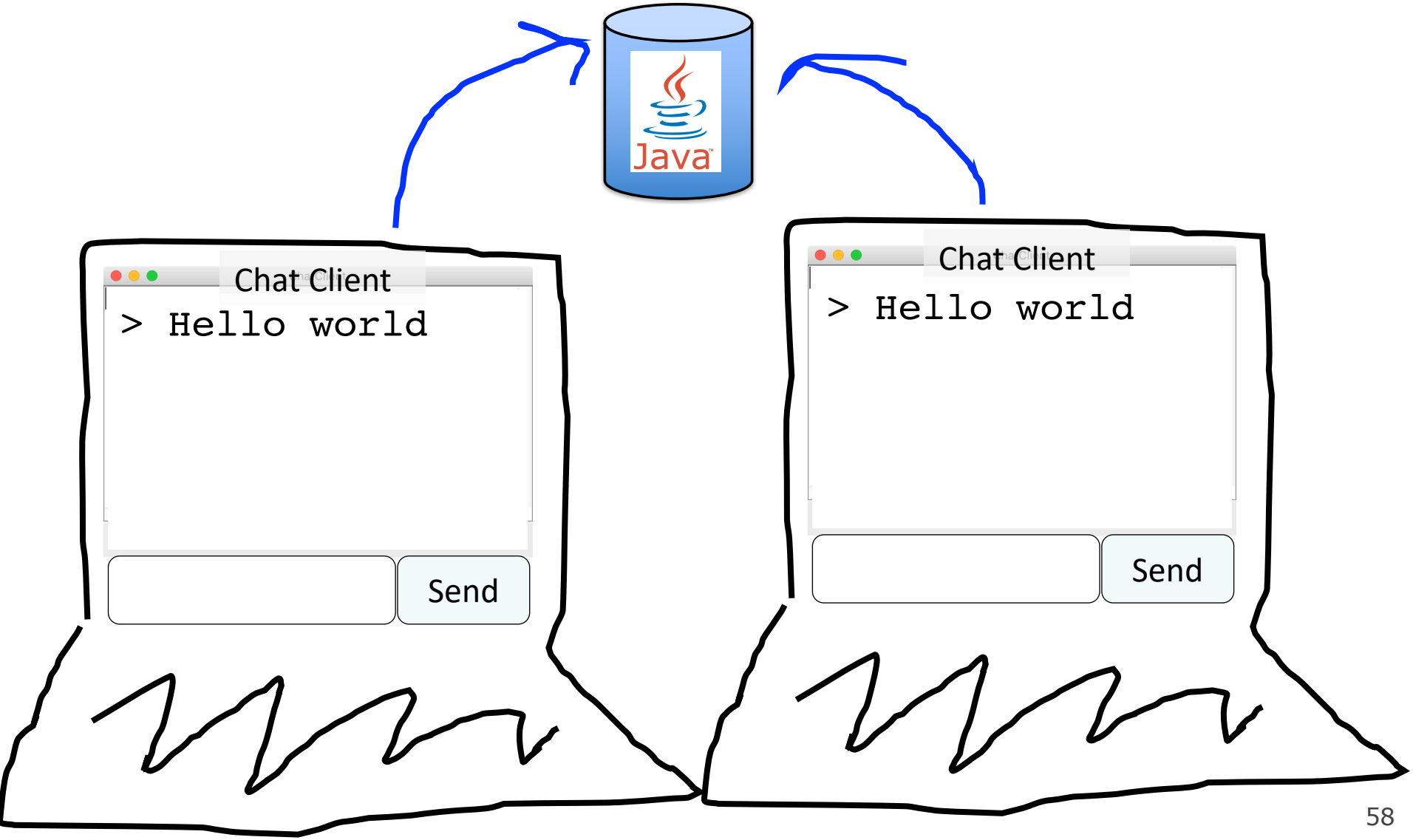
Our First Server

```
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


localhost:8090/hellocs106a
Chris Piech
localhost:8090/hellocs106a
Your command was 11 chars long.

Clients


Java
Chat Client
> Hello world
Send
Chat Client
> Hello world
Send

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

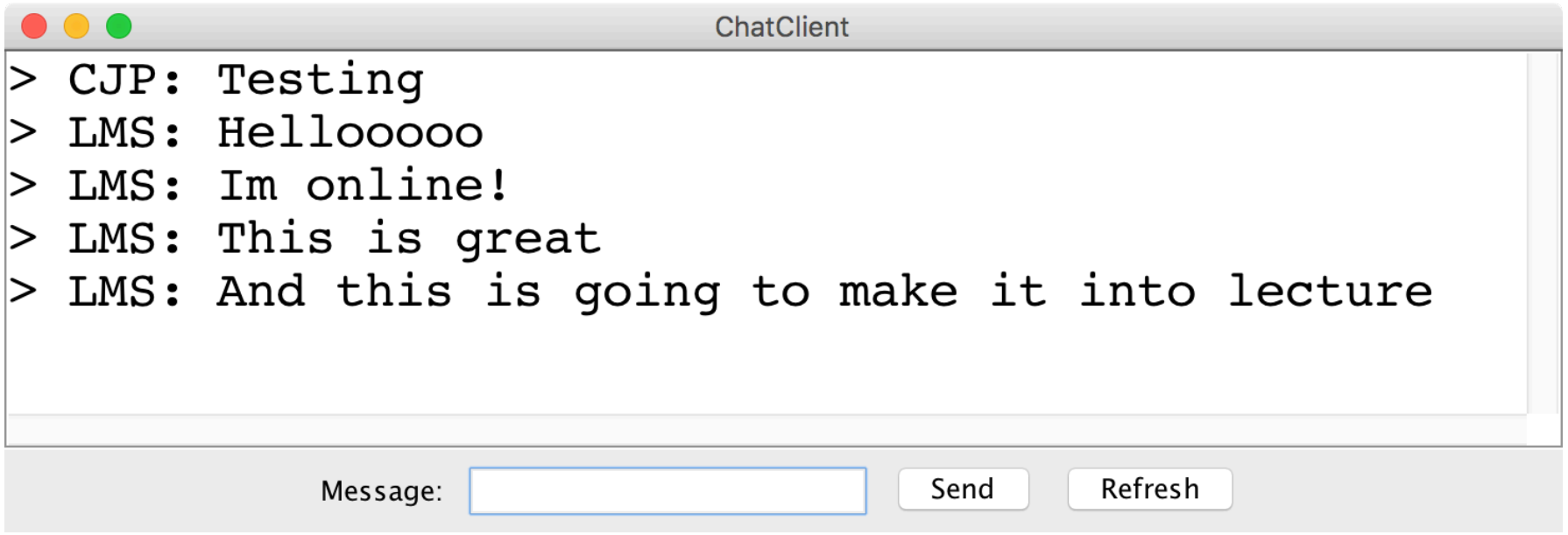
```
try {

  // 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

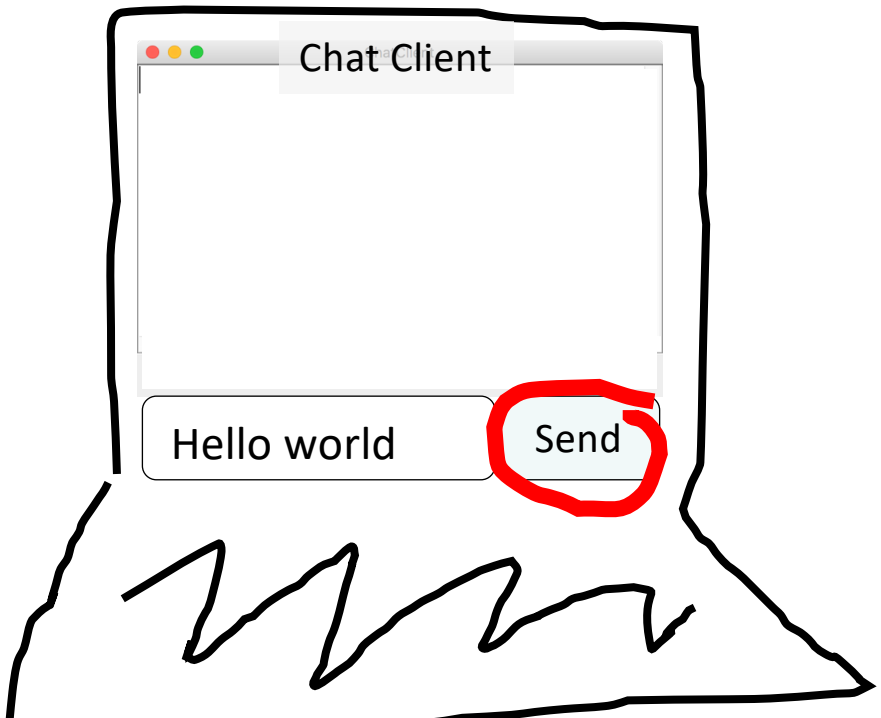
- How do internet programs work?
- Sending Data Over The Internet
- Writing a client and server
- *Demo:* echo
- ***Demo:* Chat**

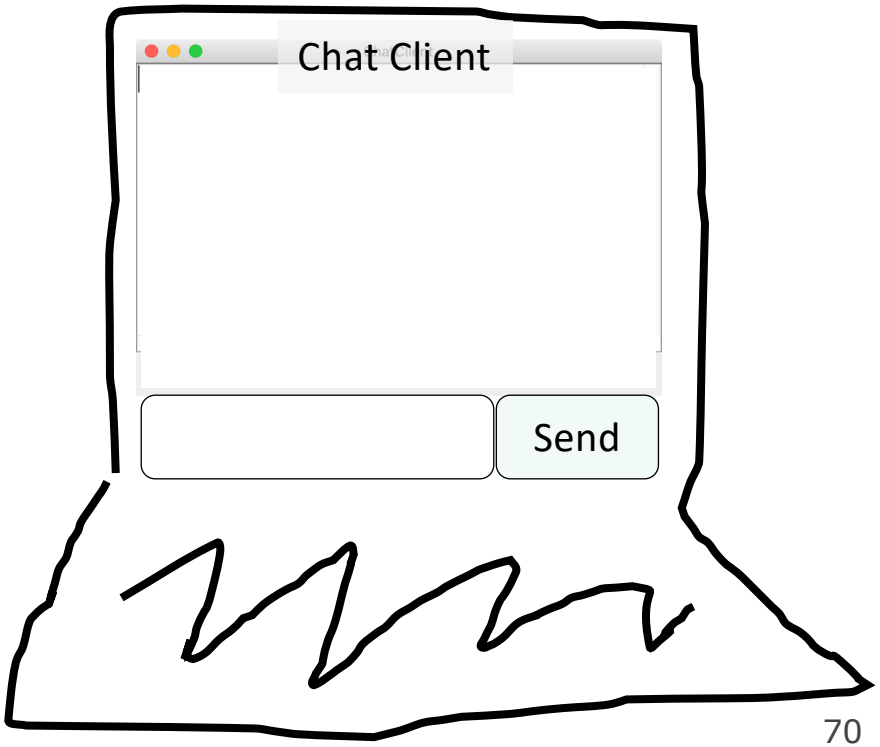
Chat Server and Client




```
history = [
]
```

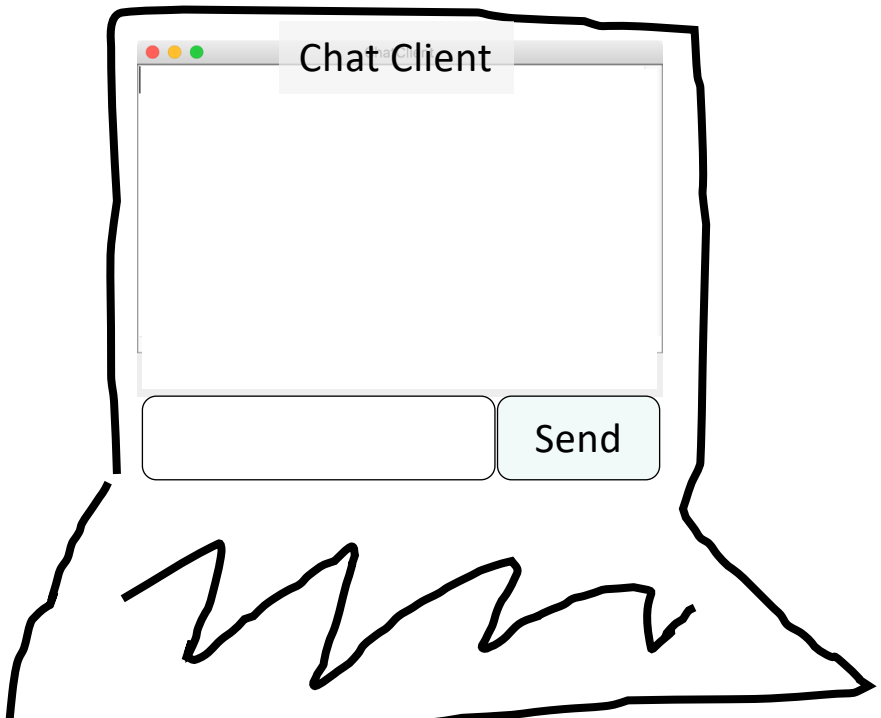
```
addMsg
msg = C: Hello world
```

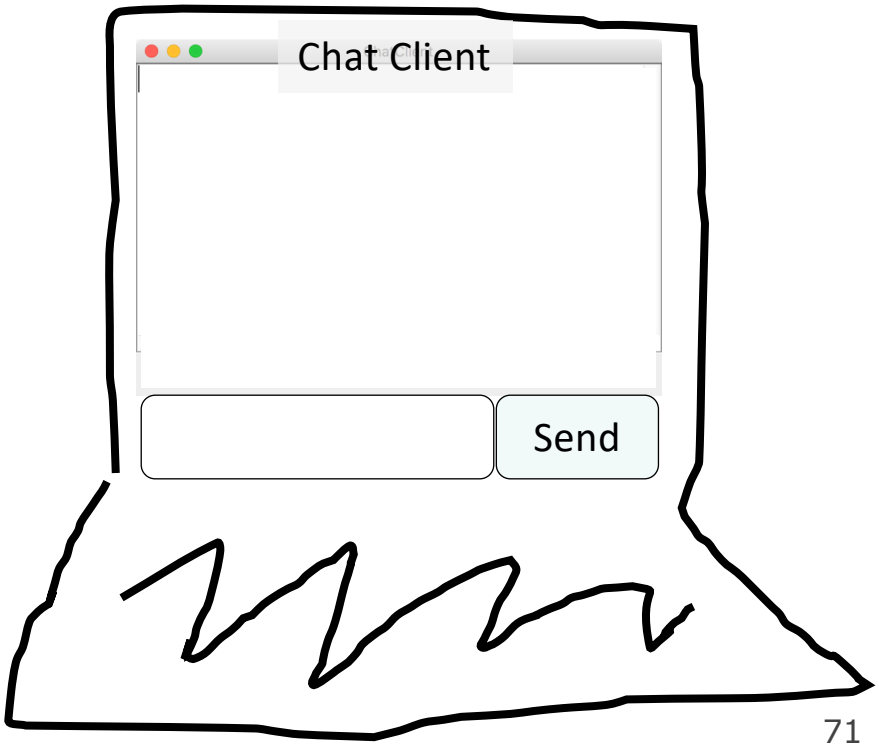





```
history = [
    C: Hello world
]
```

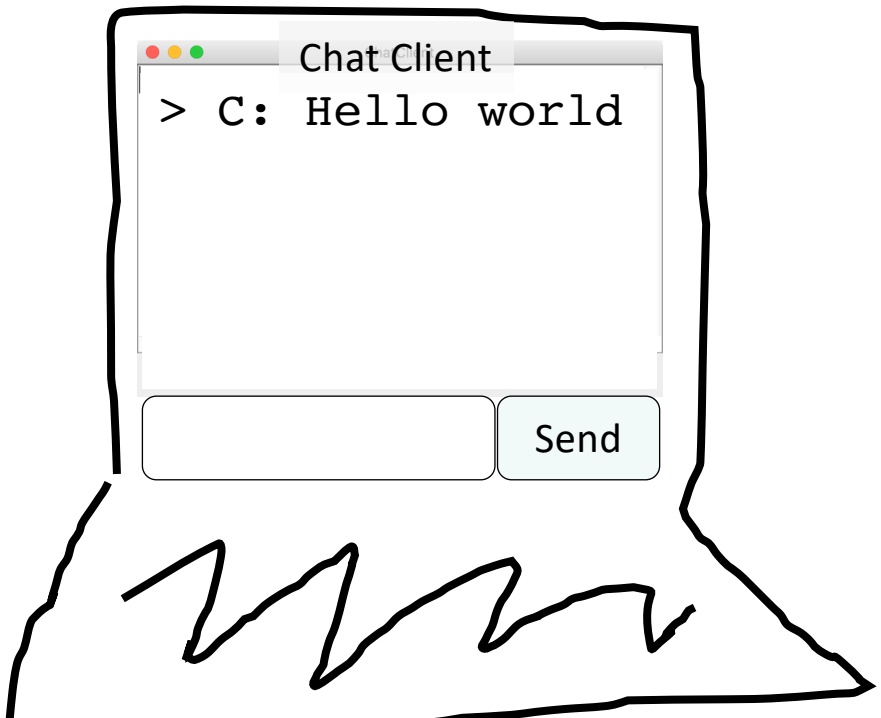
```
getMsgs
index = 0
```

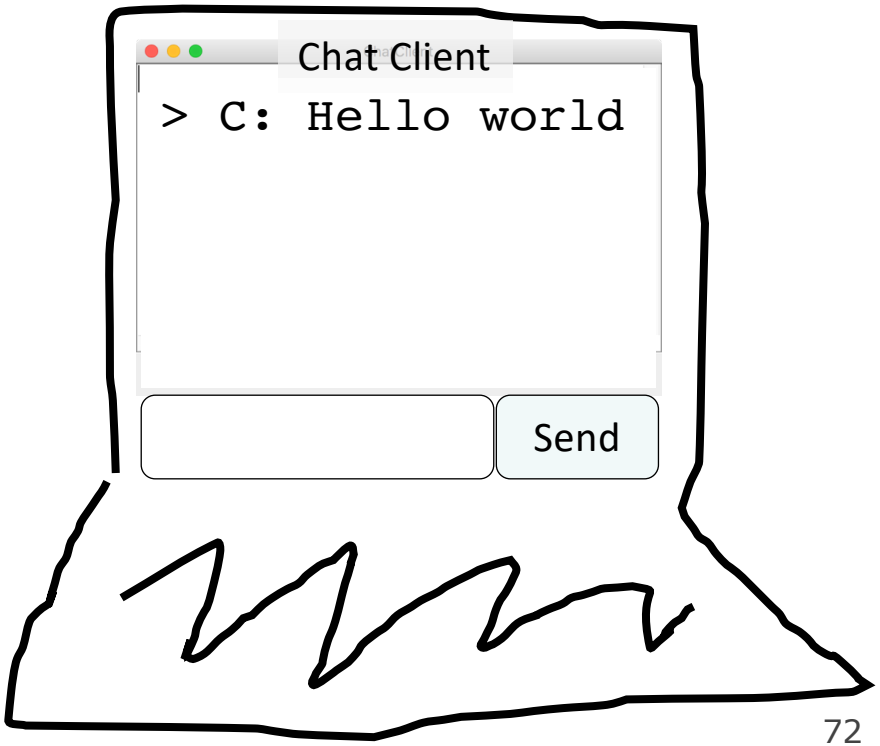





```
history = [
    C: Hello world
]
```

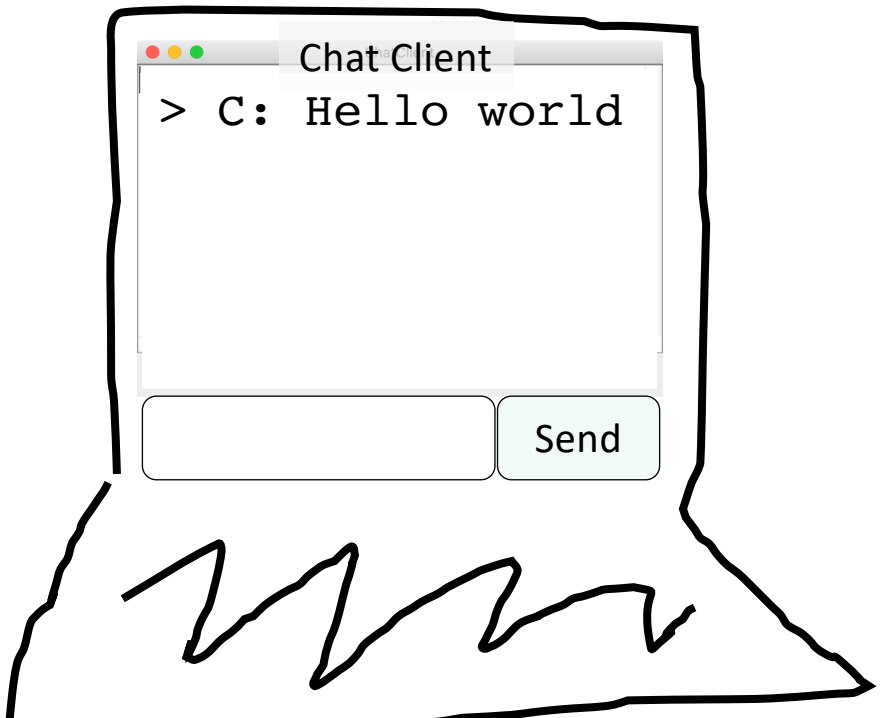
[C: Hello world]





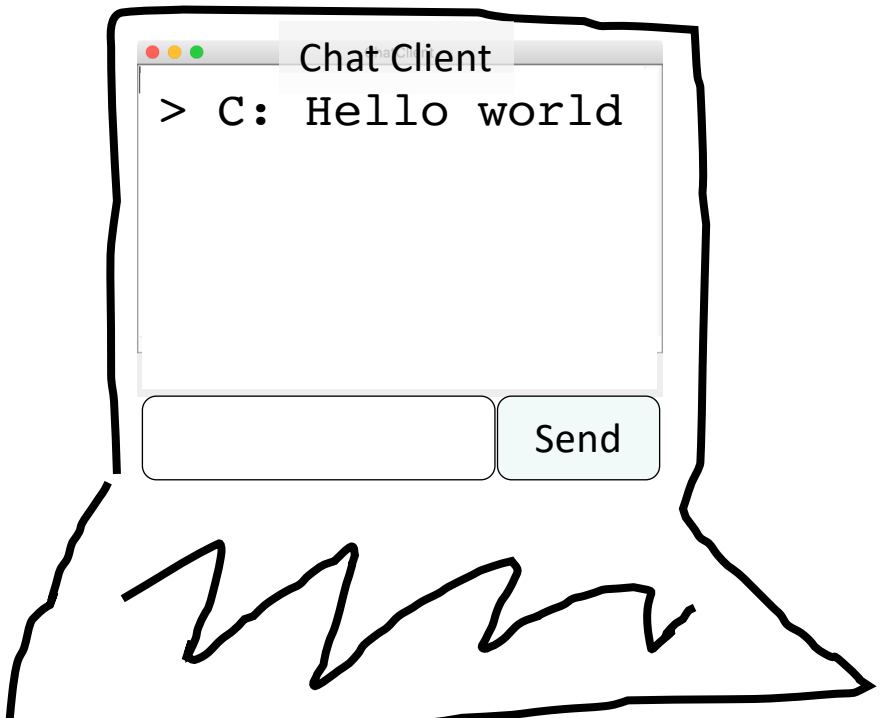

```
history = [
    C: Hello world
]
```

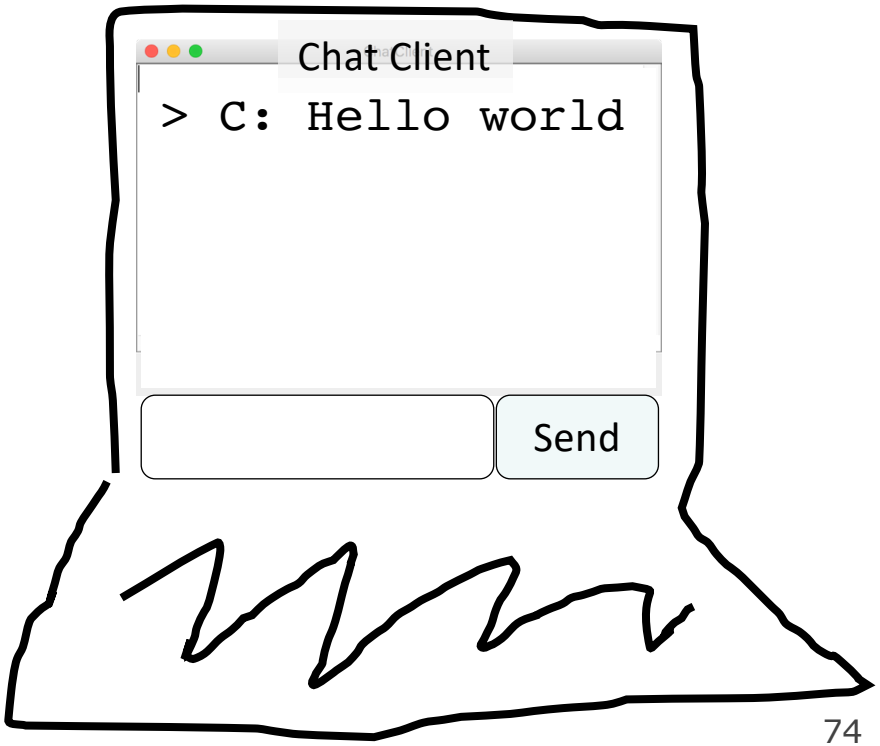
```
addMsg
msg = B: Im here too
```




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

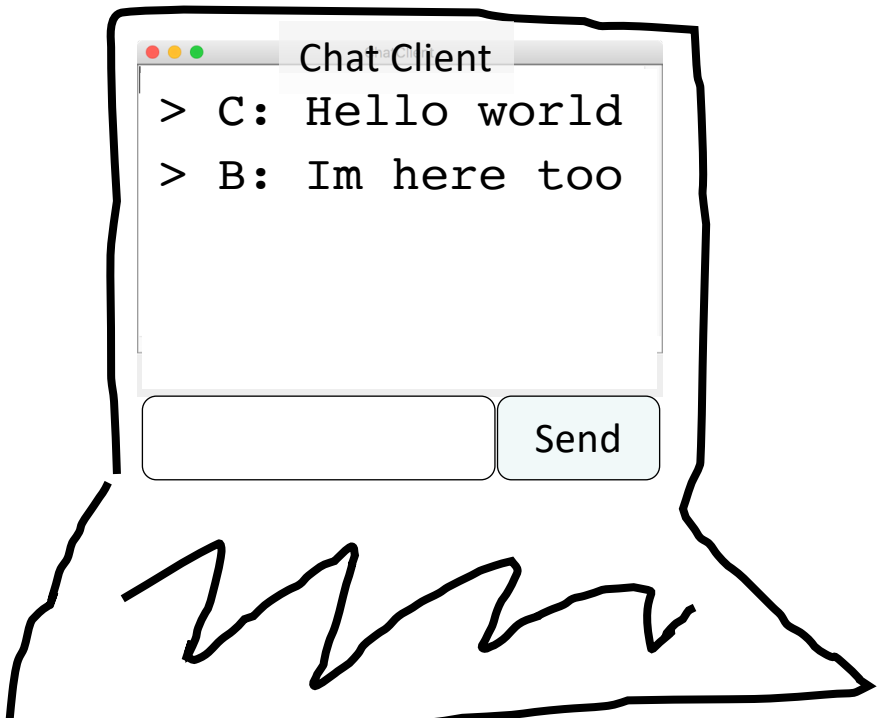
```
getMsgs
index = 1
```

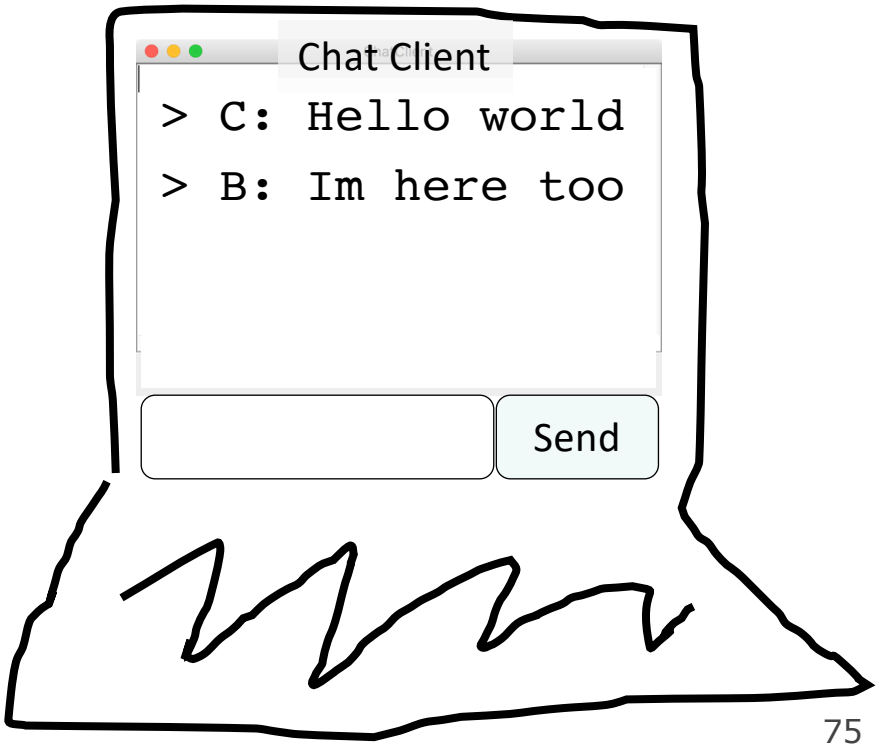





```
history = [
    C: Hello world,
    B: Im here too
]
```

[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