

# The Internet

Chris Piech

CS106A, Stanford University

I came here to make friends  
and program the internet...

... And I already made  
friends.



For the first time ever in  
CS106A:

# Learning Goals

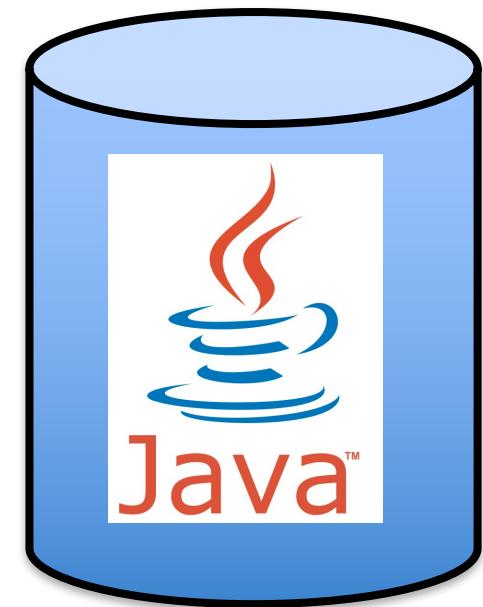
1. Write a program that can make internet requests
2. Write a program that can respond to internet requests



How does your phone  
communicate with facebook?

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

Face Book Server

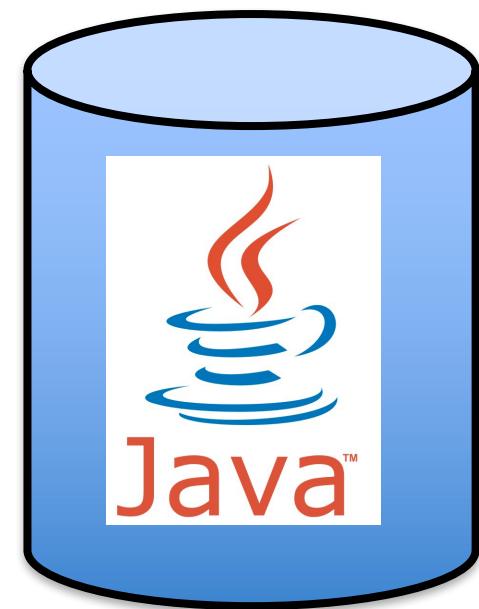
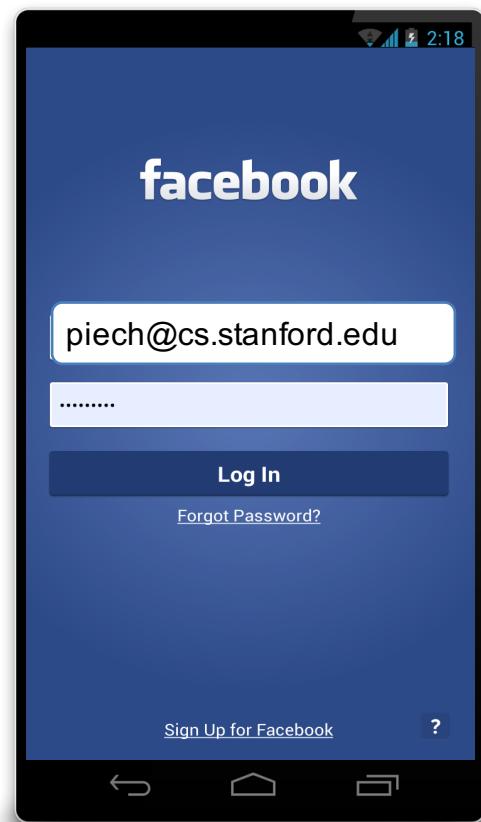


\* Android phones run Java. So do facebook servers



Face Book Server

Is this legit?

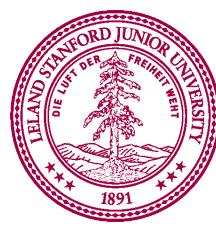
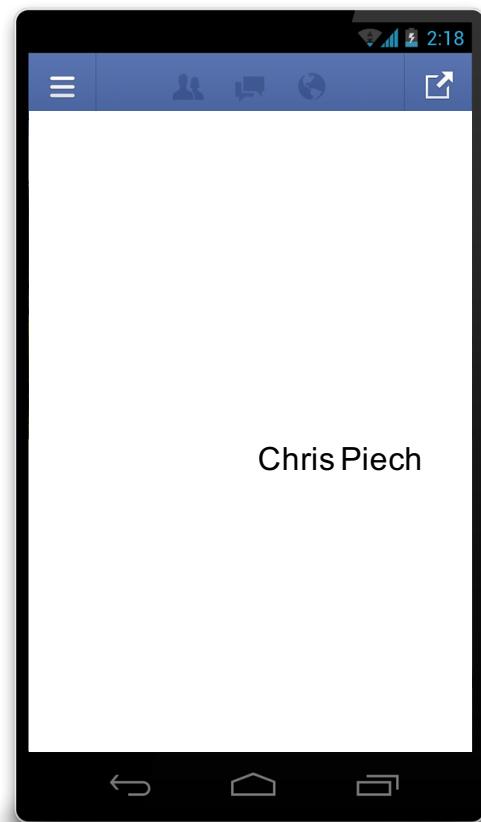


piech@cs.stanford.edu  
is now logged in



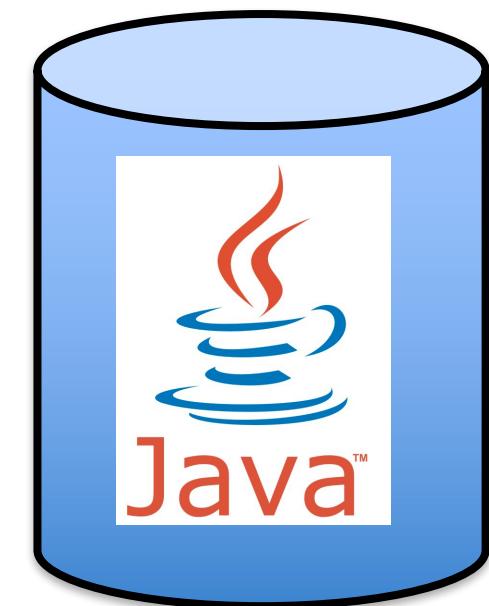
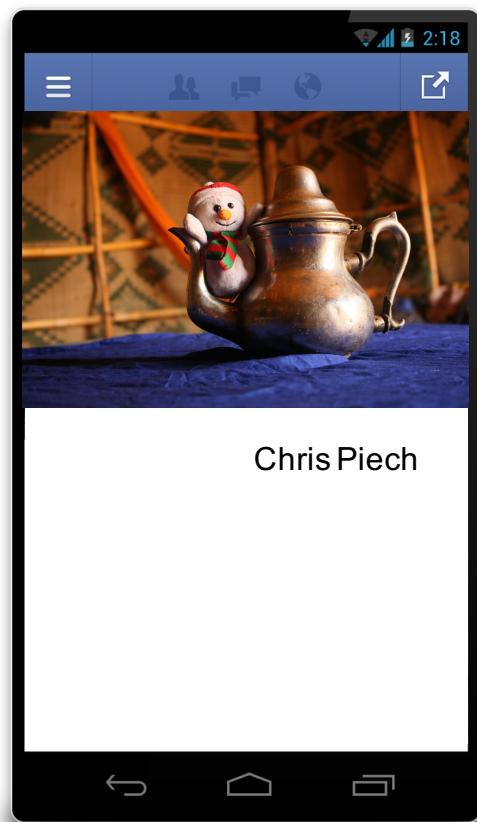
# Face Book Server

Send me the **full name** for  
piech@cs.stanford.edu



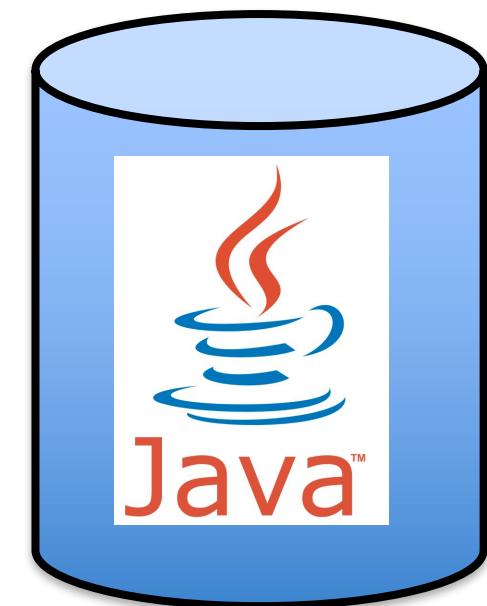
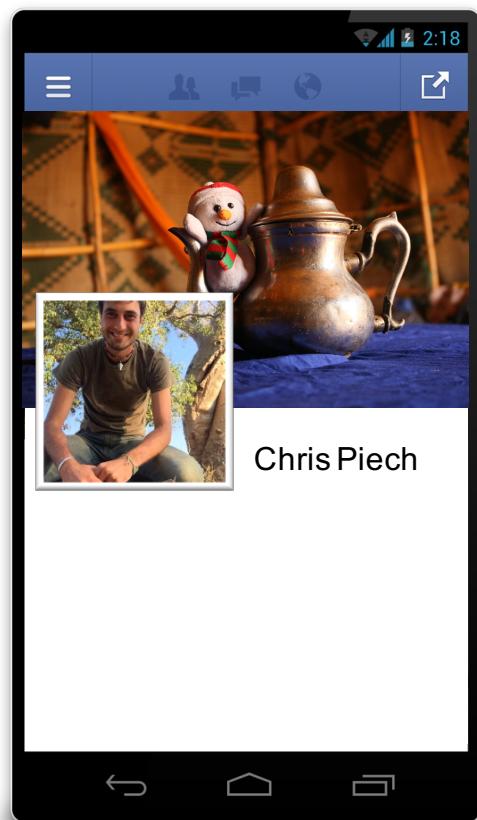
# Face Book Server

Send me the **cover photo** for  
piech@cs.stanford.edu



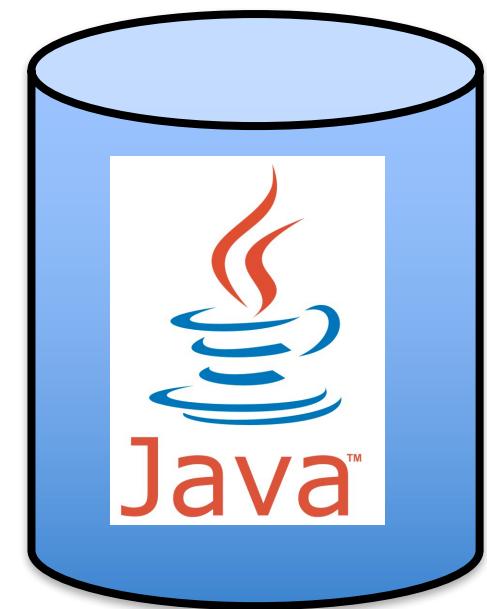
# Face Book Server

Send the **profile photo** for  
piech@cs.stanford.edu



# Face Book Server

Send the **status** for  
piech@cs.stanford.edu

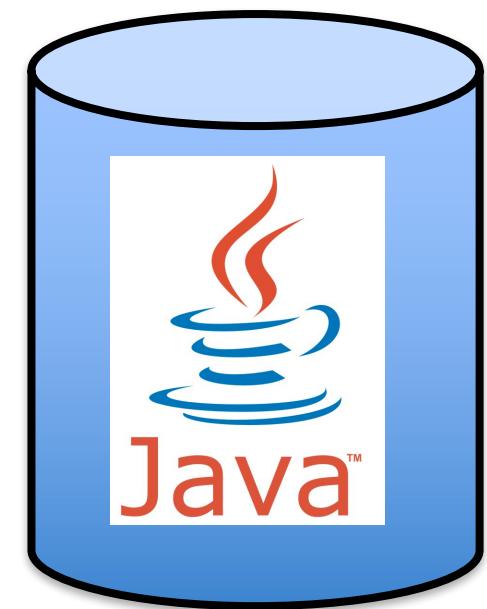


“chillin”

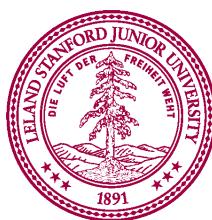


# Face Book Server

Set the **status** for  
piech@cs.stanford.edu  
to be "**lecturing**"

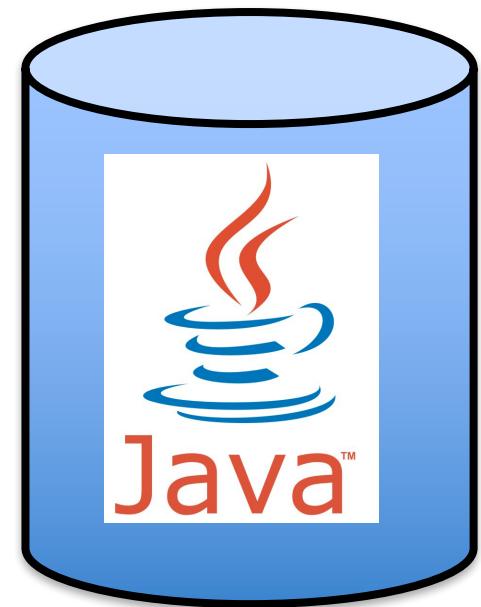


"success"

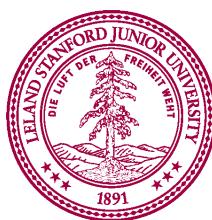


# Face Book Server

Send me the **status** for  
piech@cs.stanford.edu

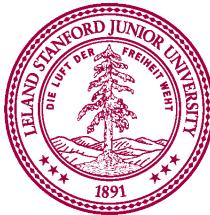


"lecturing"





There are two types of  
internet programs. Servers  
and Clients



# Internet 101

# Servers are computers (running code)

Face Book Server



=



# Facebook's closest datacenter is here

I am here



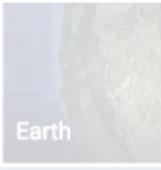
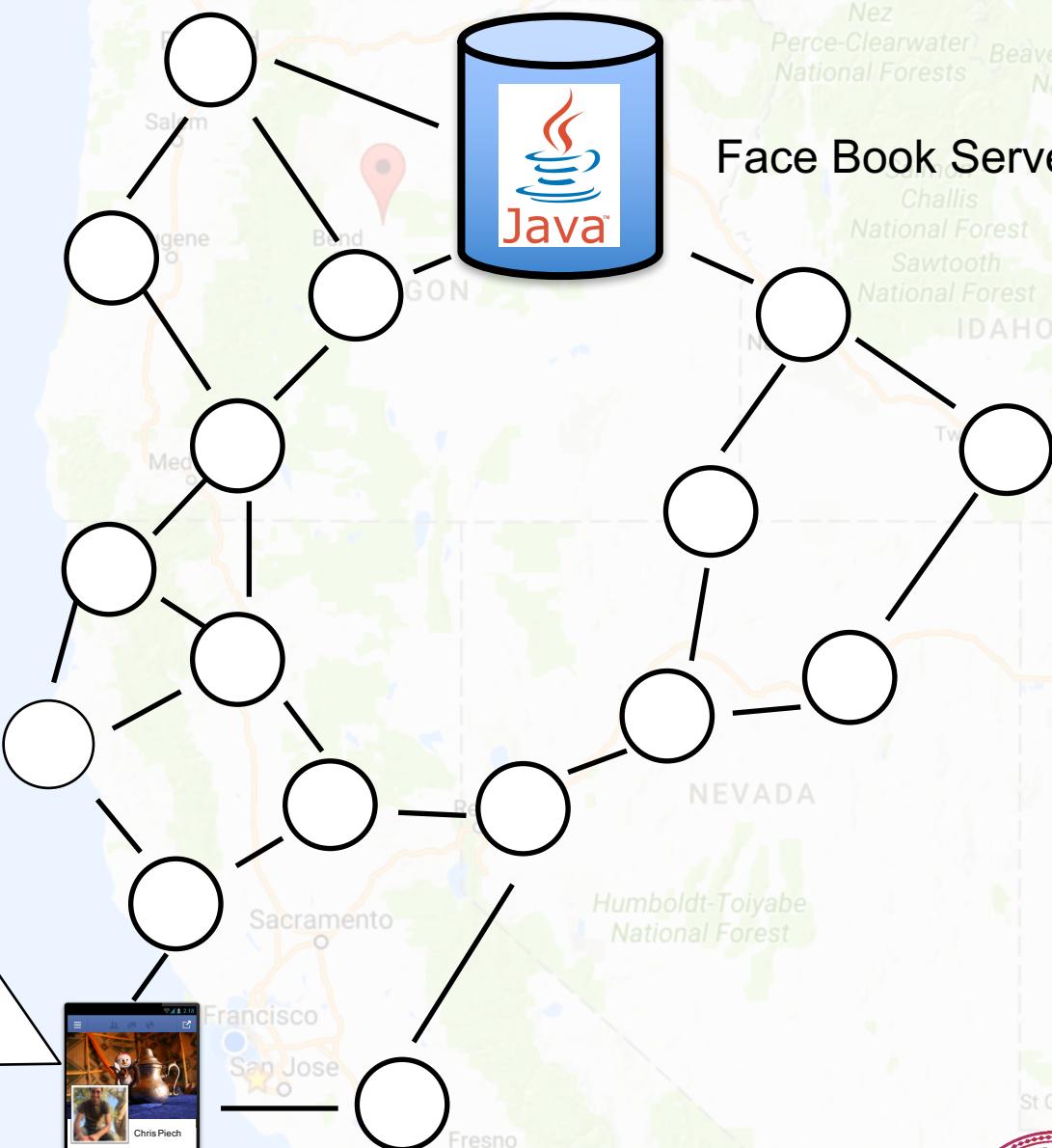
Google

The Internet

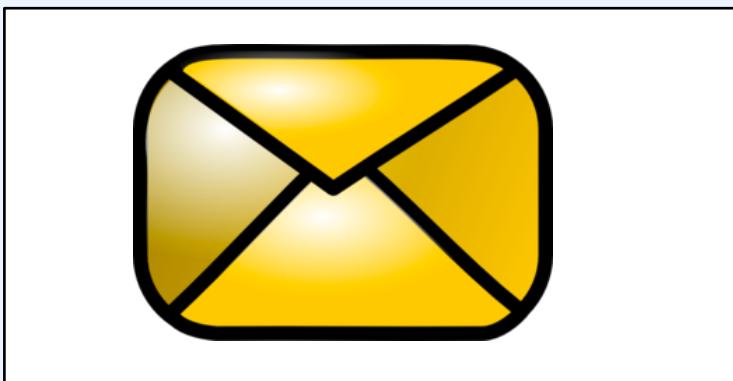


The Internet

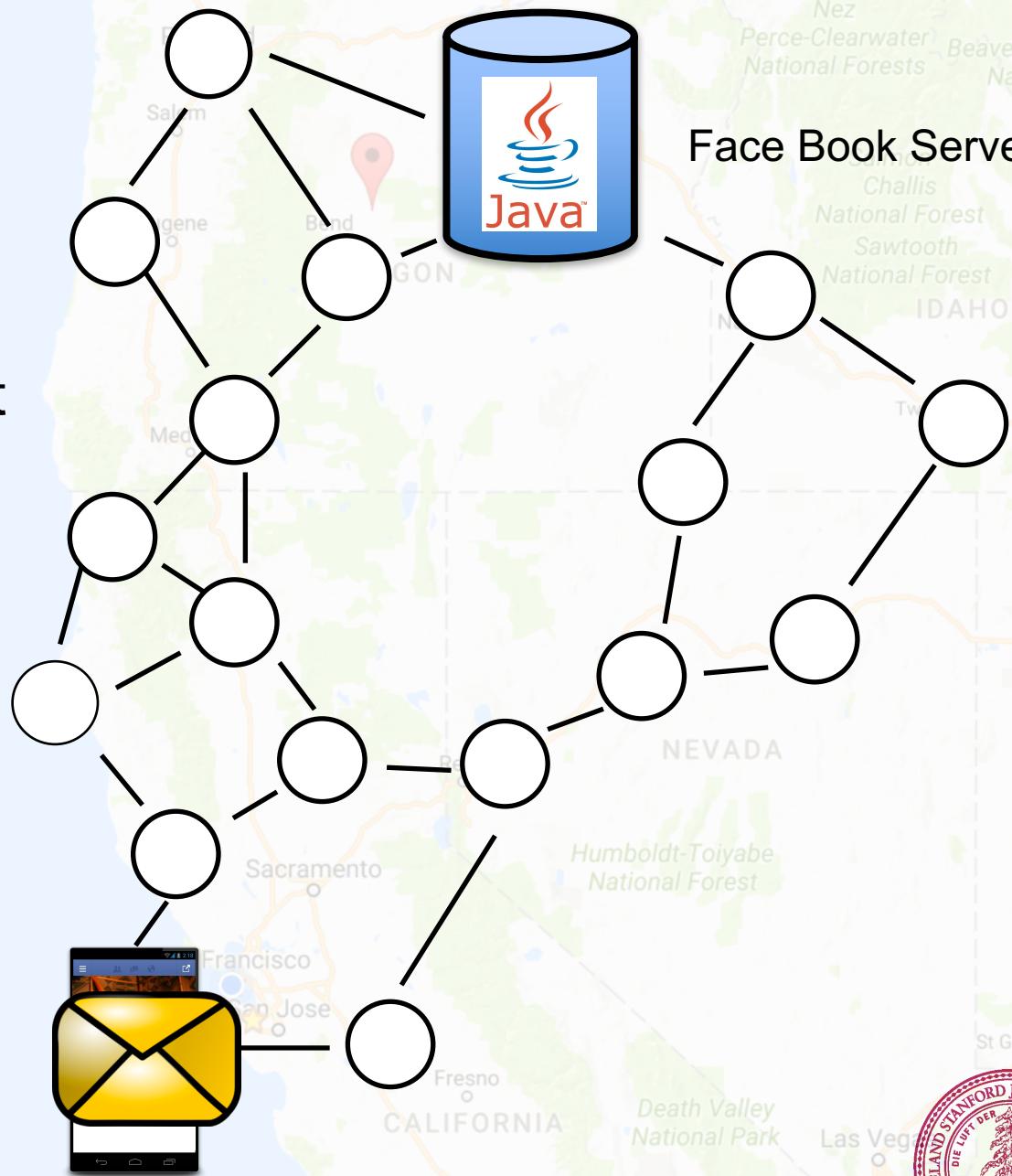
Get status for  
piech@cs.stanford.edu



# The Internet



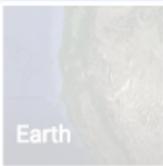
The Internet

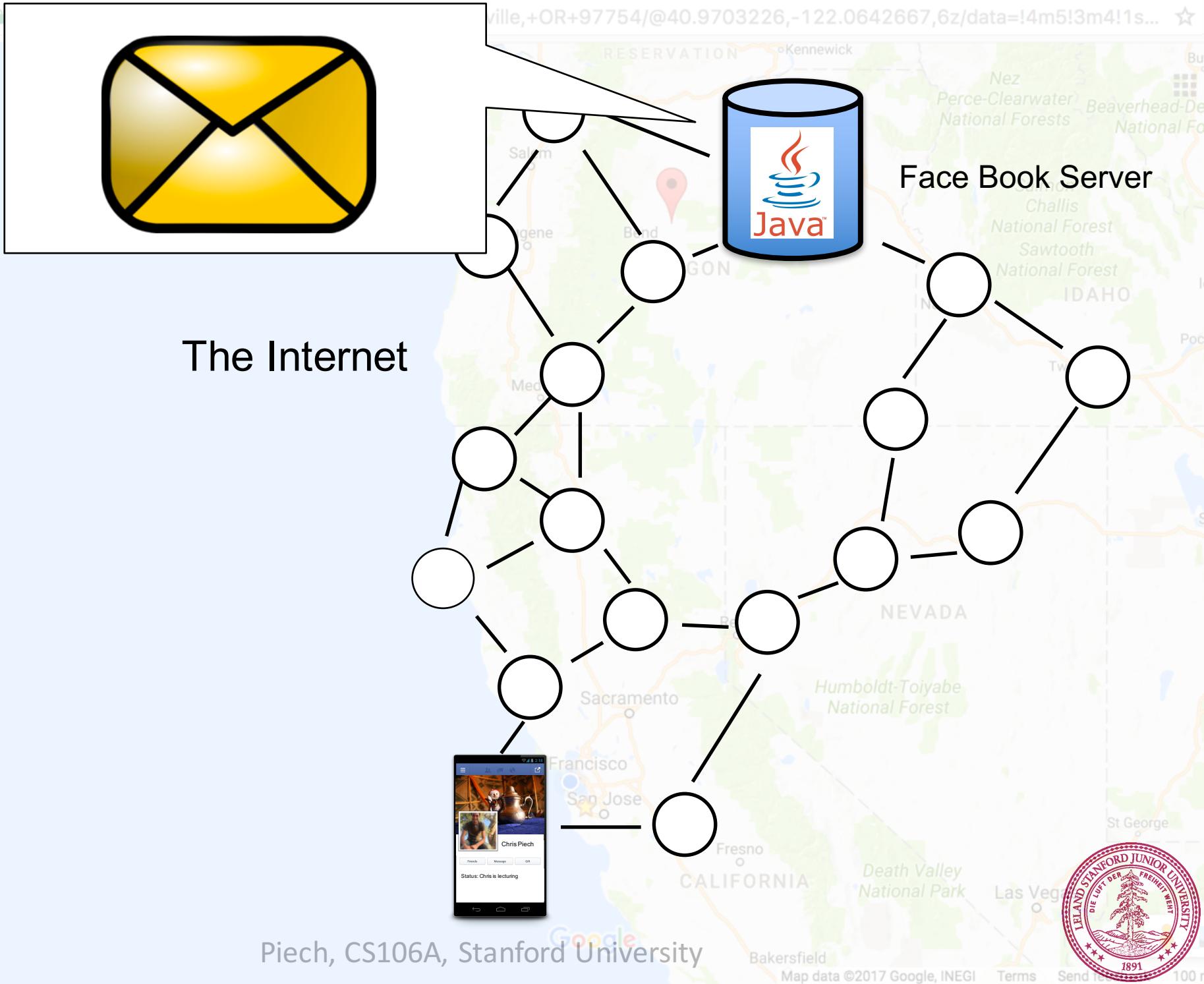


teaching

The Internet

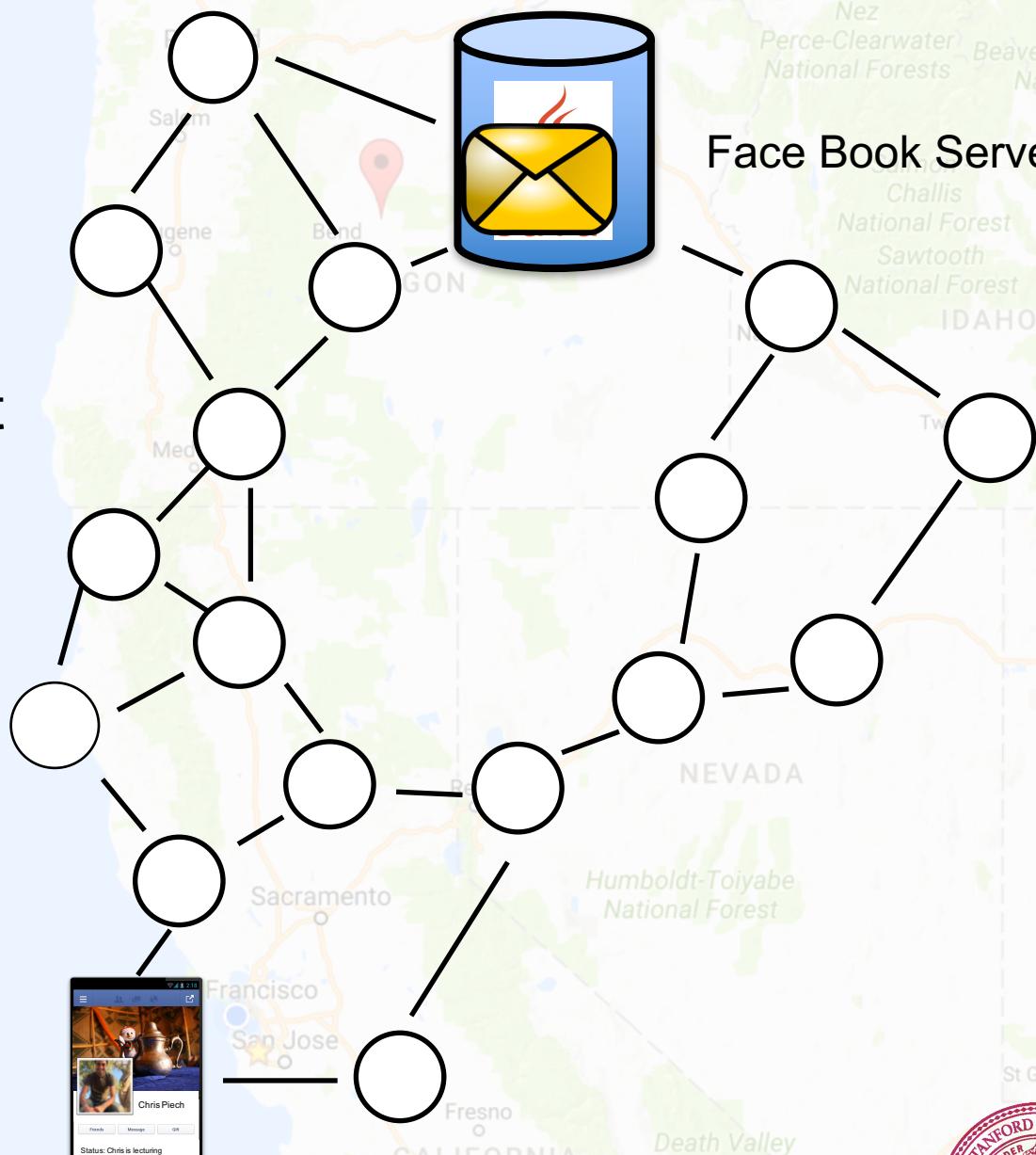
Face Book Server





teaching

The Internet



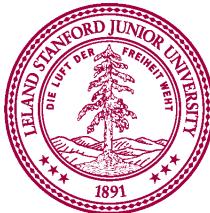
Earth

Piech, CS106A, Stanford University





There are two types of  
internet programs. Servers  
and Clients

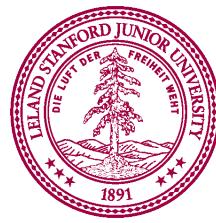
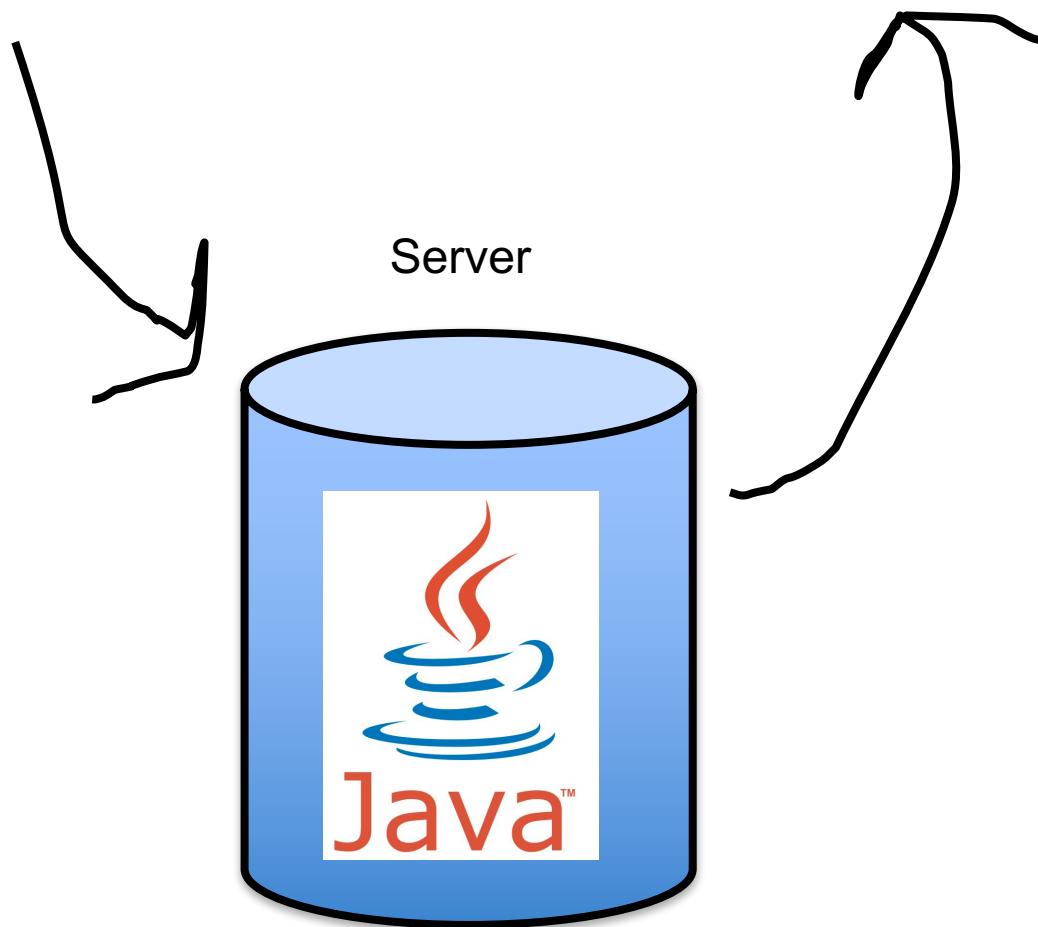


First, the server

# A Server's Simple Purpose

**Request**  
someRequest

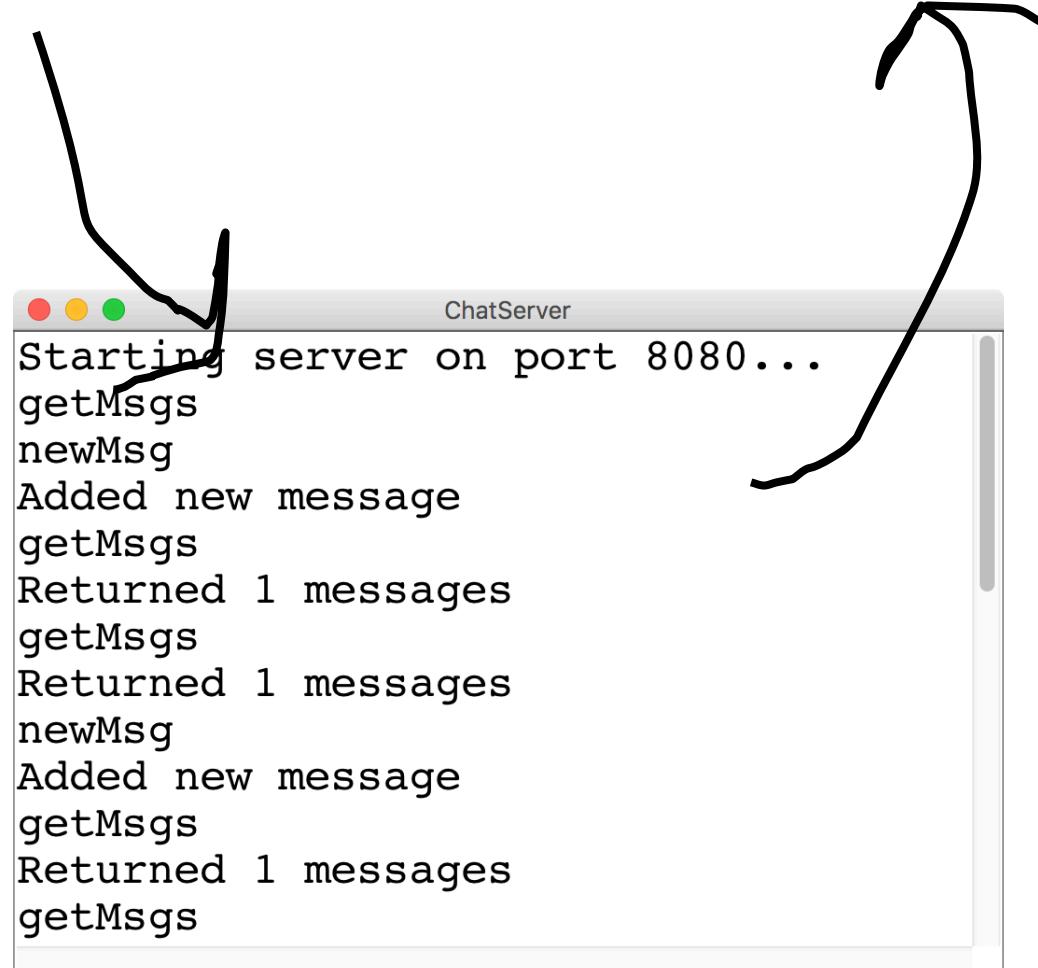
**String**  
serverResponse



# A Server's Simple Purpose

**Request**  
someRequest

**String**  
serverResponse



```
ChatServer
Starting server on port 8080...
getMsgs
newMsg
Added new message
getMsgs
Returned 1 messages
getMsgs
Returned 1 messages
newMsg
Added new message
getMsgs
Returned 1 messages
getMsgs
```



# 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();  
}
```



# What is a Request?



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

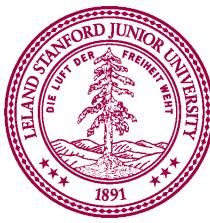
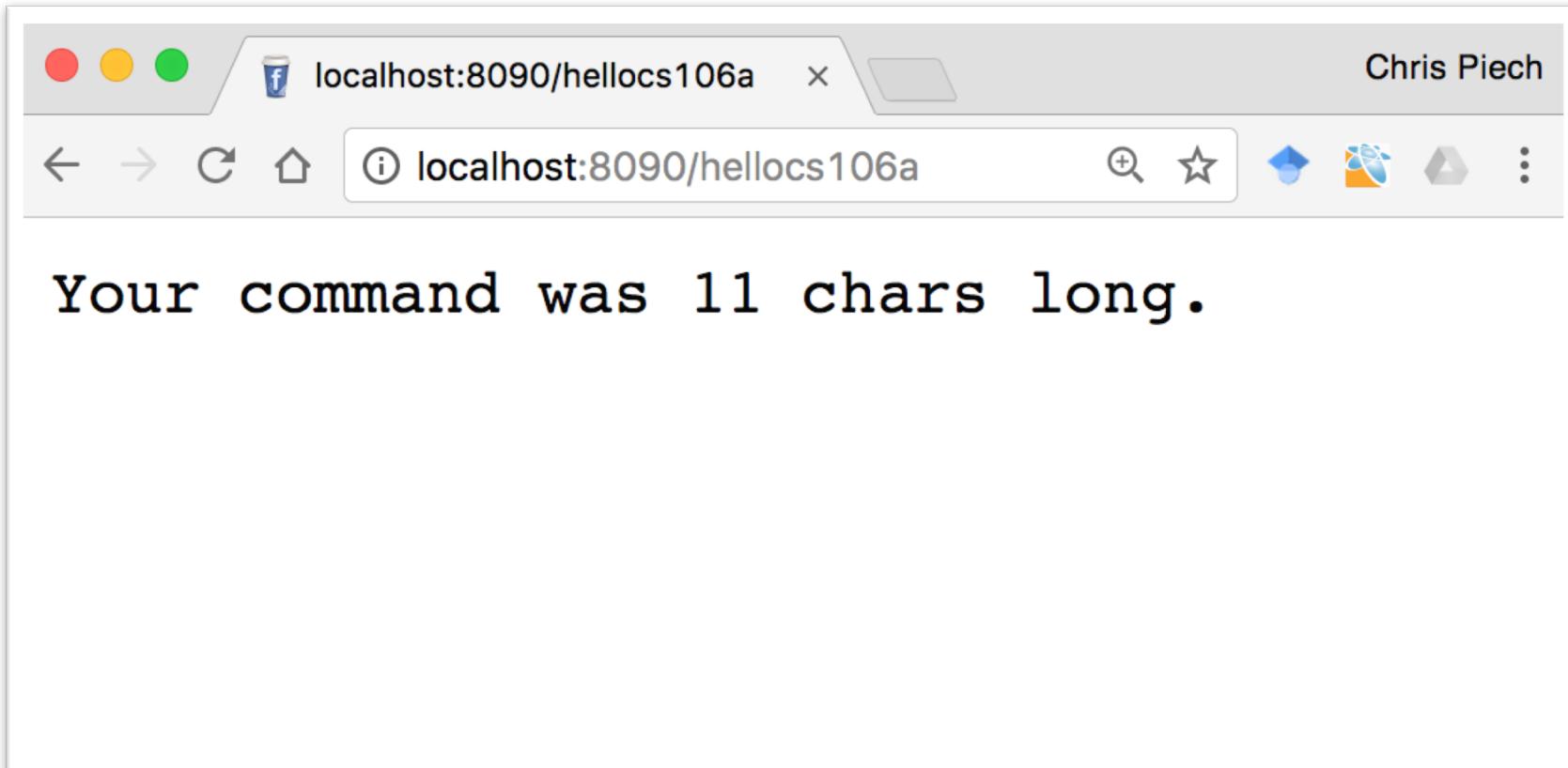
## Request request

---

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



# Echo Server



# Echo Server

**Request**

Any Request

**String**

Length of the cmd



The screenshot shows a terminal window titled "EchoServer". The window contains the following text:

```
Starting server...
Request received hello
Request received this+is+a+test
Request received whatsGood
Request received ping
Request received ping
Request received ping
Request received pong
Request received ping
```



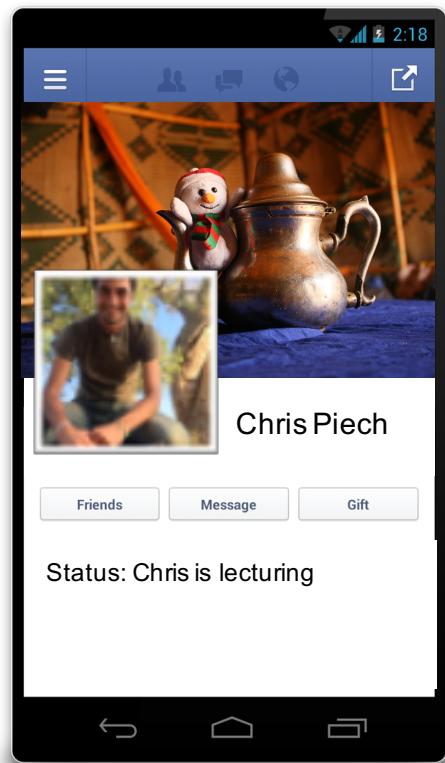


There are two types of  
internet programs. Servers  
and Clients



Then, the client

# A Client's Purpose



1. Interact with the user
2. Get data from its server
3. Save data to its server



# 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  
  
}
```



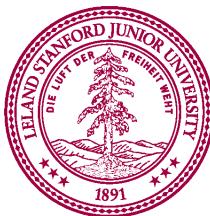
# Clients on one slide

```
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  
}
```



Time for a little chat

# Chat Server and Client

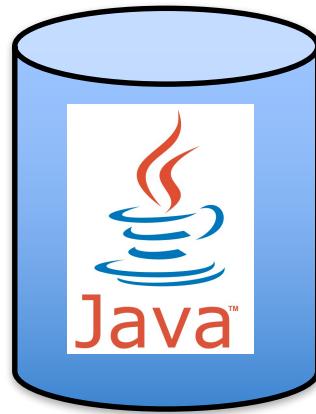
ChatClient

```
> CJP: Testing
> LMS: Hellooooo
> LMS: Im online!
> LMS: This is great
> LMS: And this is going to make it into lecture
```

Message:

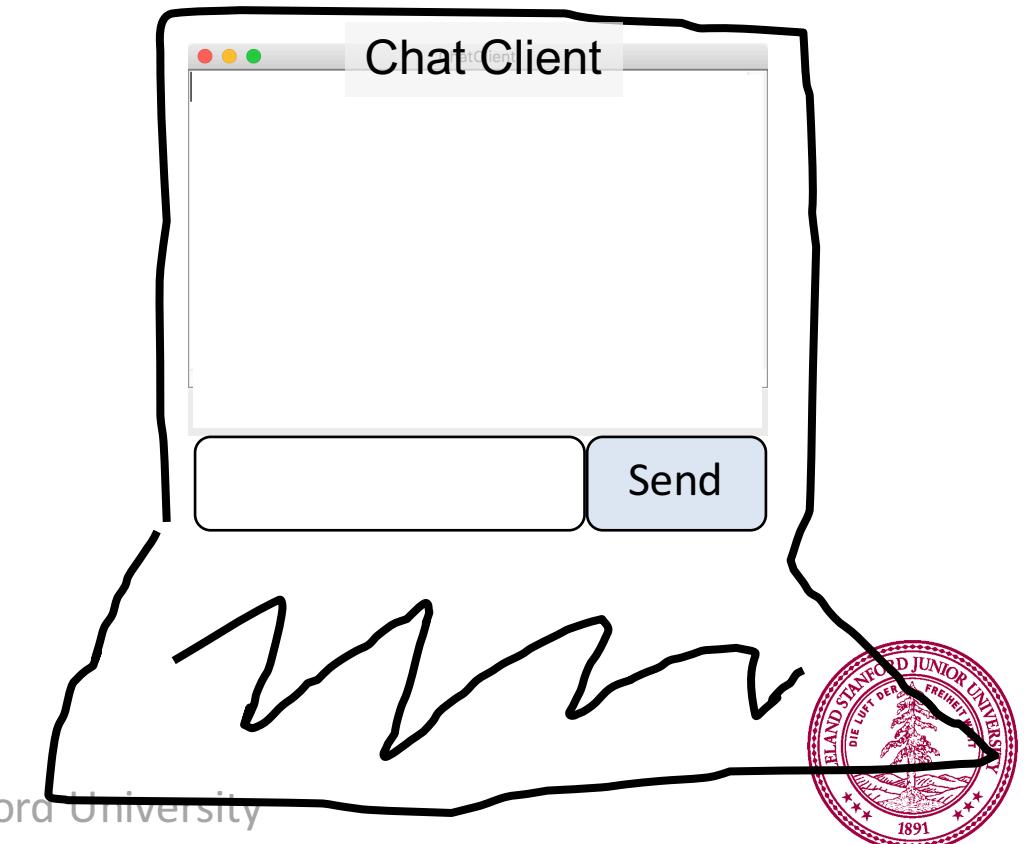
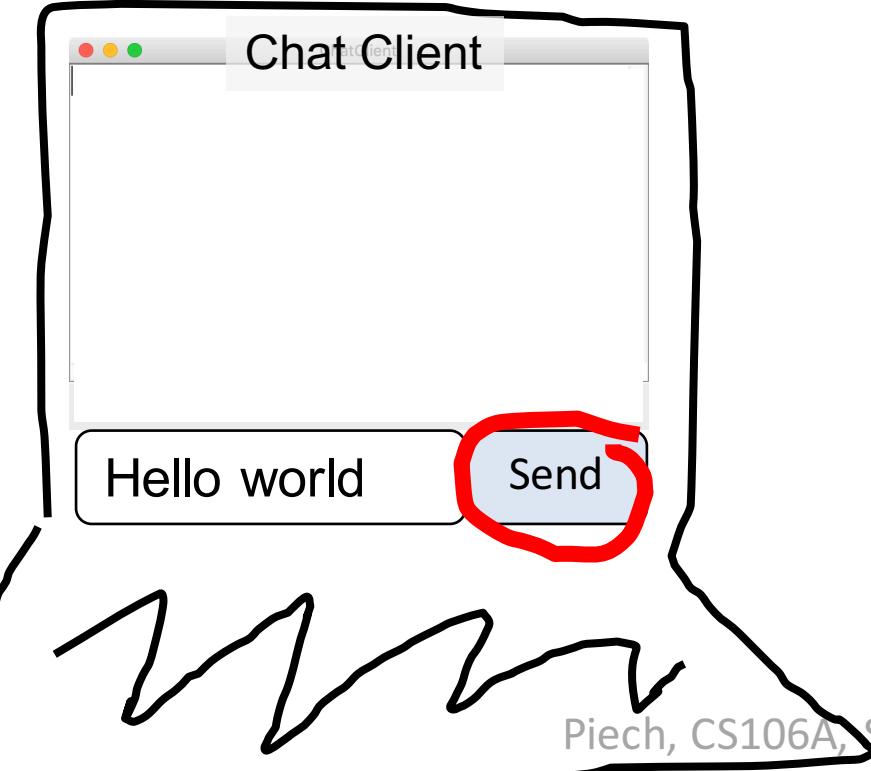
Send Refresh

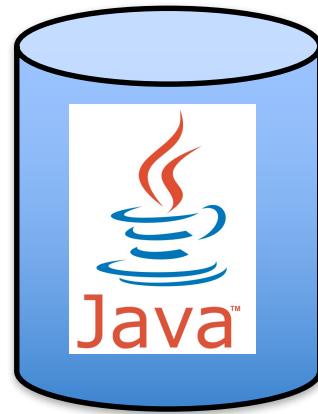




```
history = [  
]
```

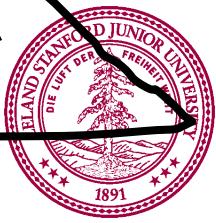
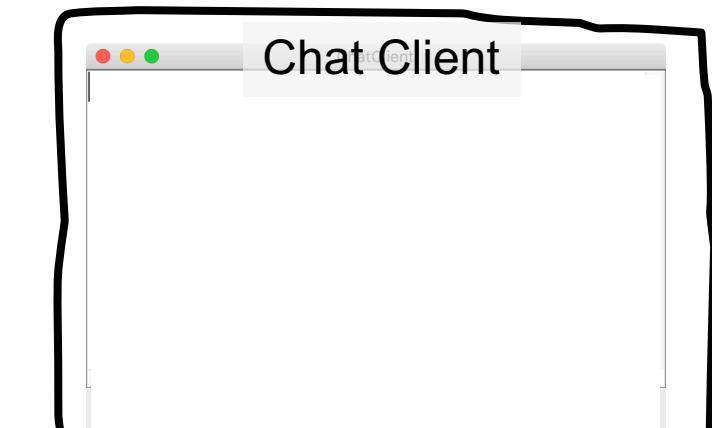
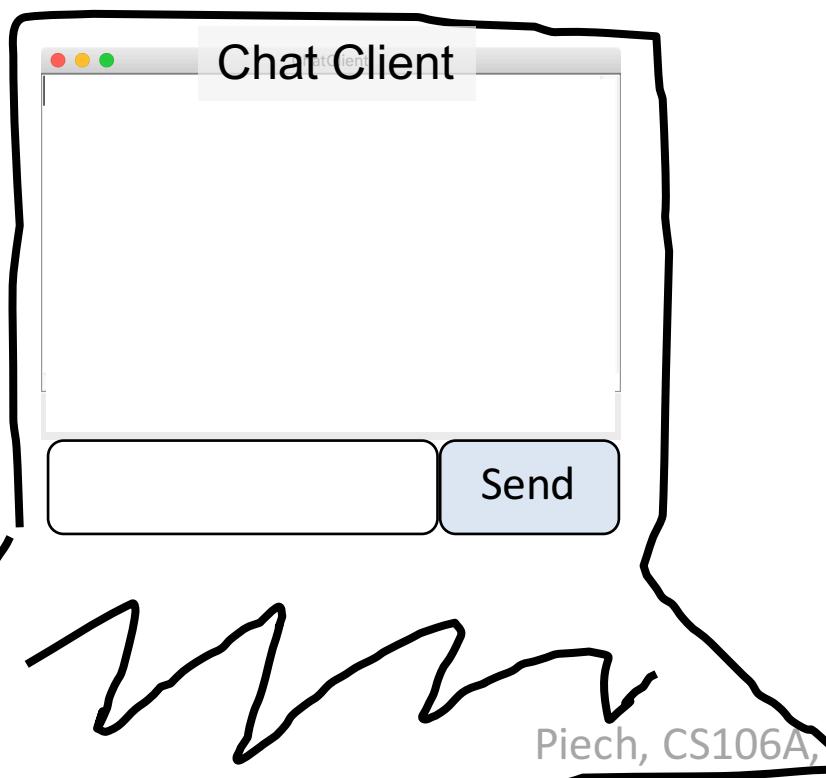
addMsg  
msg = Hello world

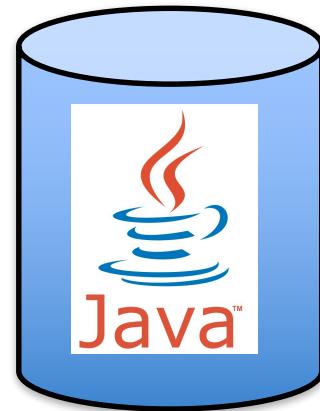




```
history = [  
    Hello world  
]
```

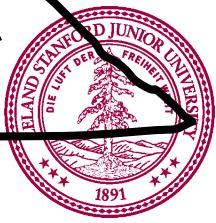
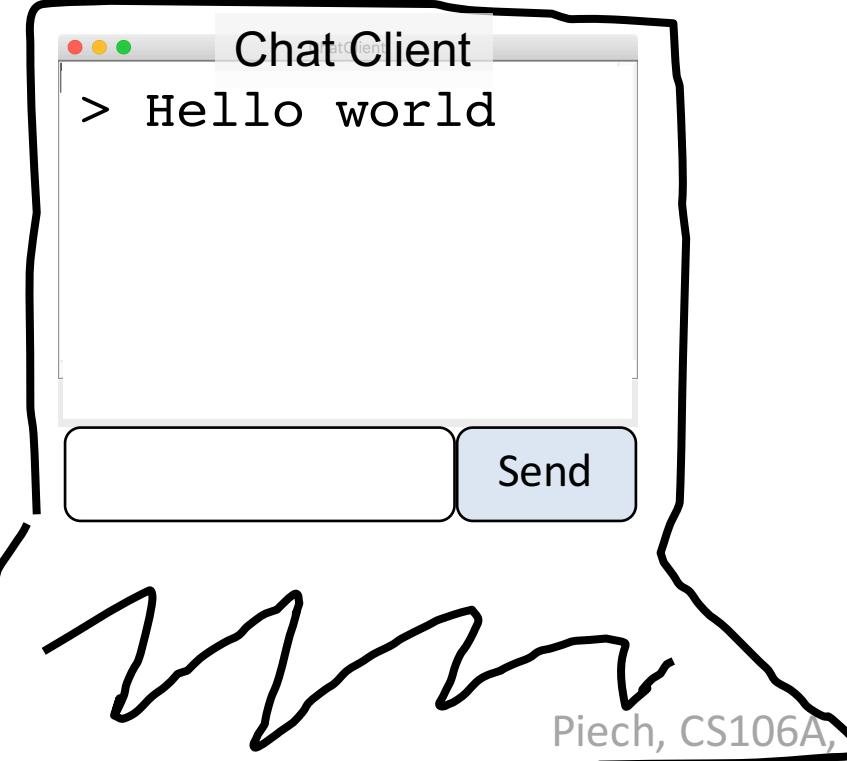
getMsgs  
index = 0

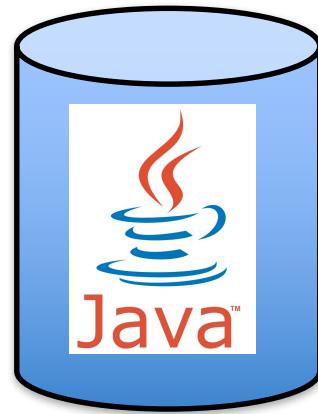




```
history = [  
    Hello world  
]
```

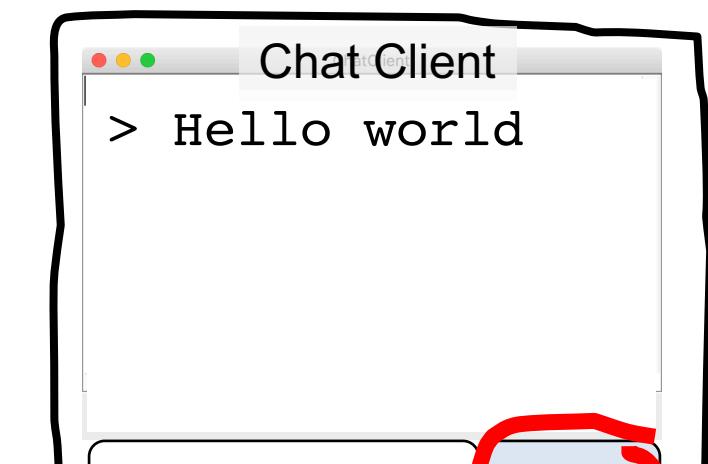
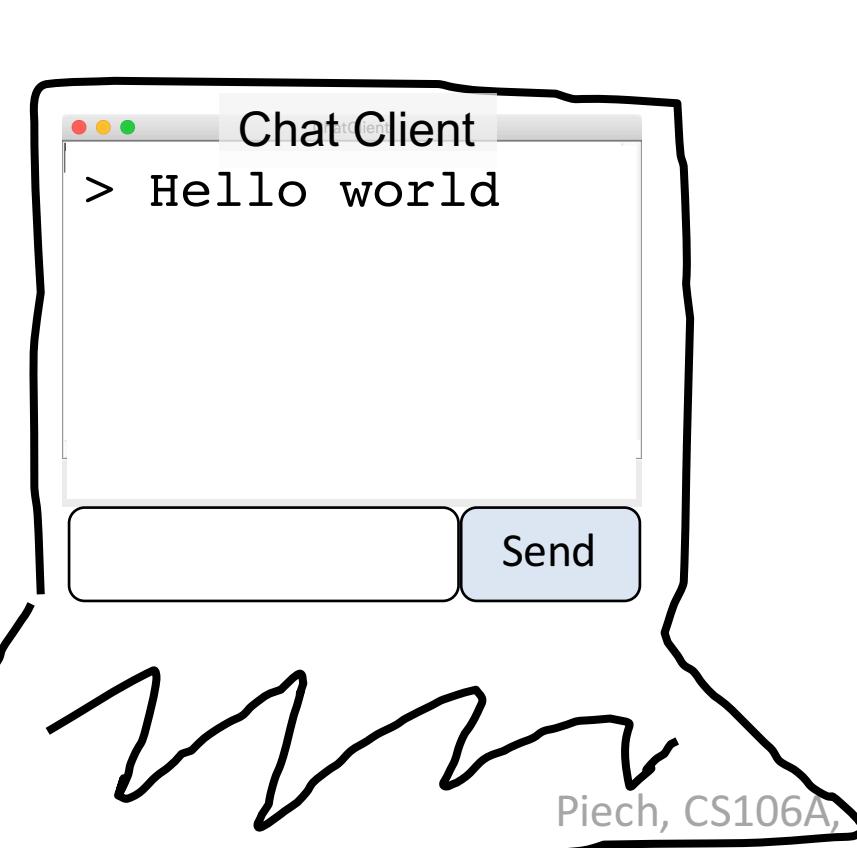
[Hello world]





```
history = [  
    Hello world  
]
```

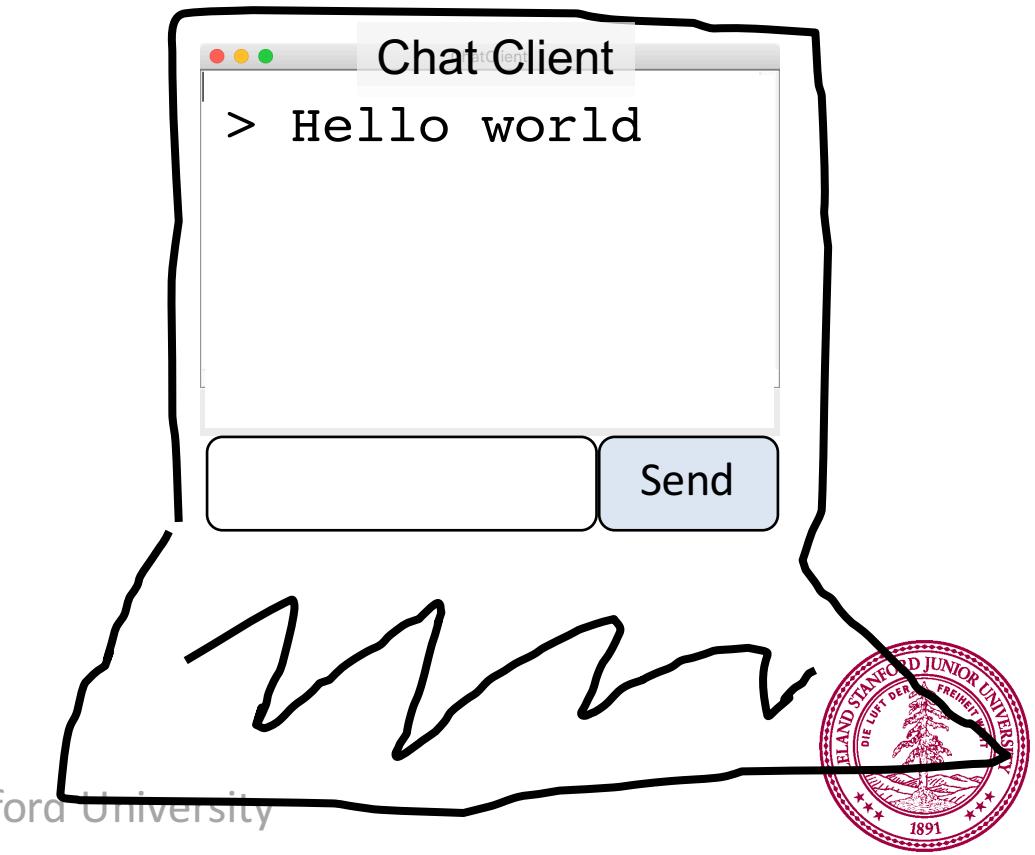
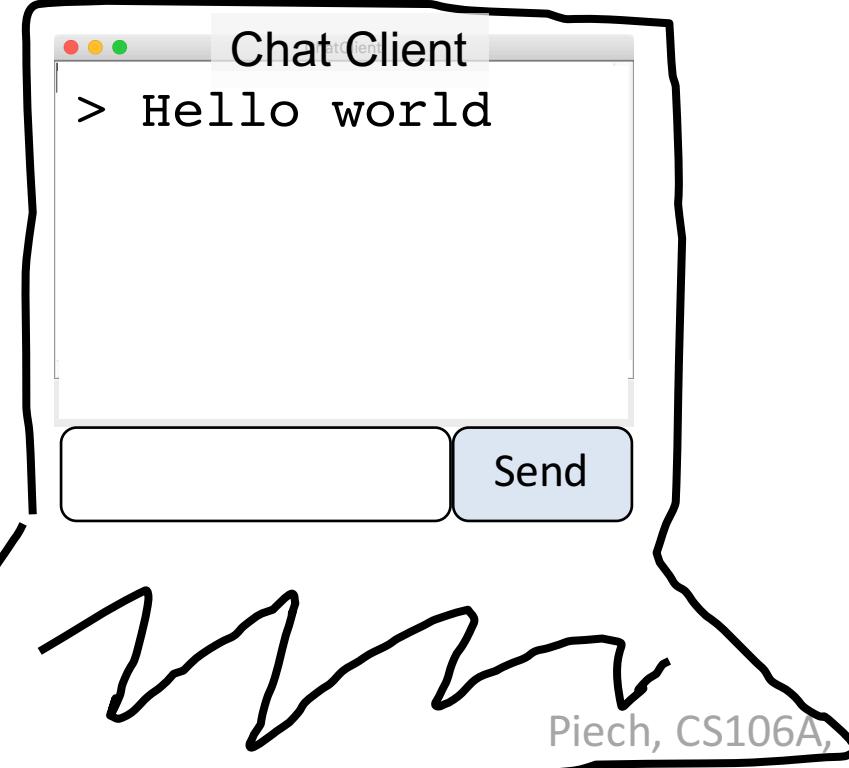
addMsg  
msg = Im here too

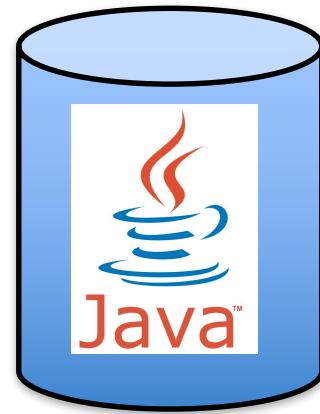




```
history = [  
    Hello world,  
    Im here too  
]
```

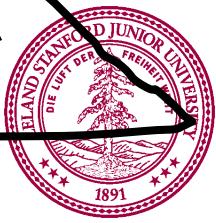
getMsgs  
index = 1





```
history = [  
    Hello world,  
    Im here too  
]
```

[ Im here too ]

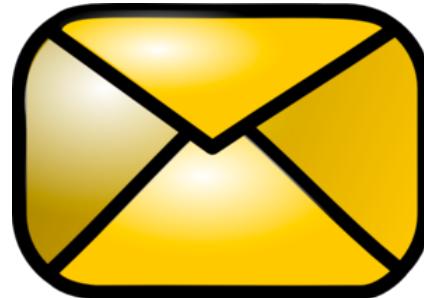


# Chat Server

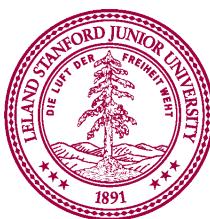
Chat Server



addMsg  
msg = *text*



getMsgs  
index = *startIndex*





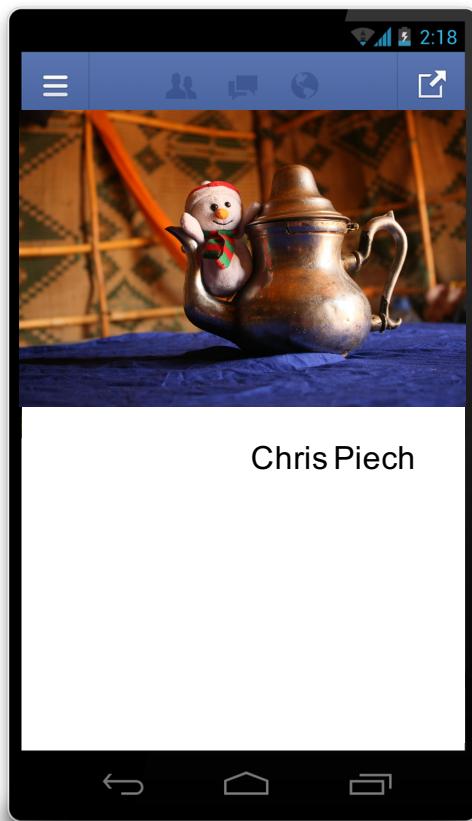
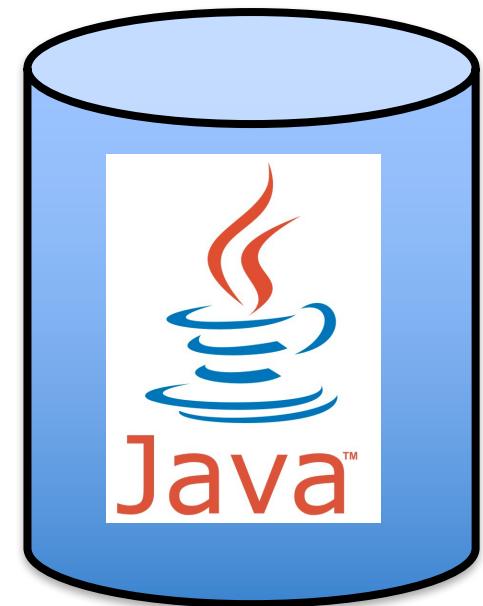
# Facebook revisited

Command: getProfileImg

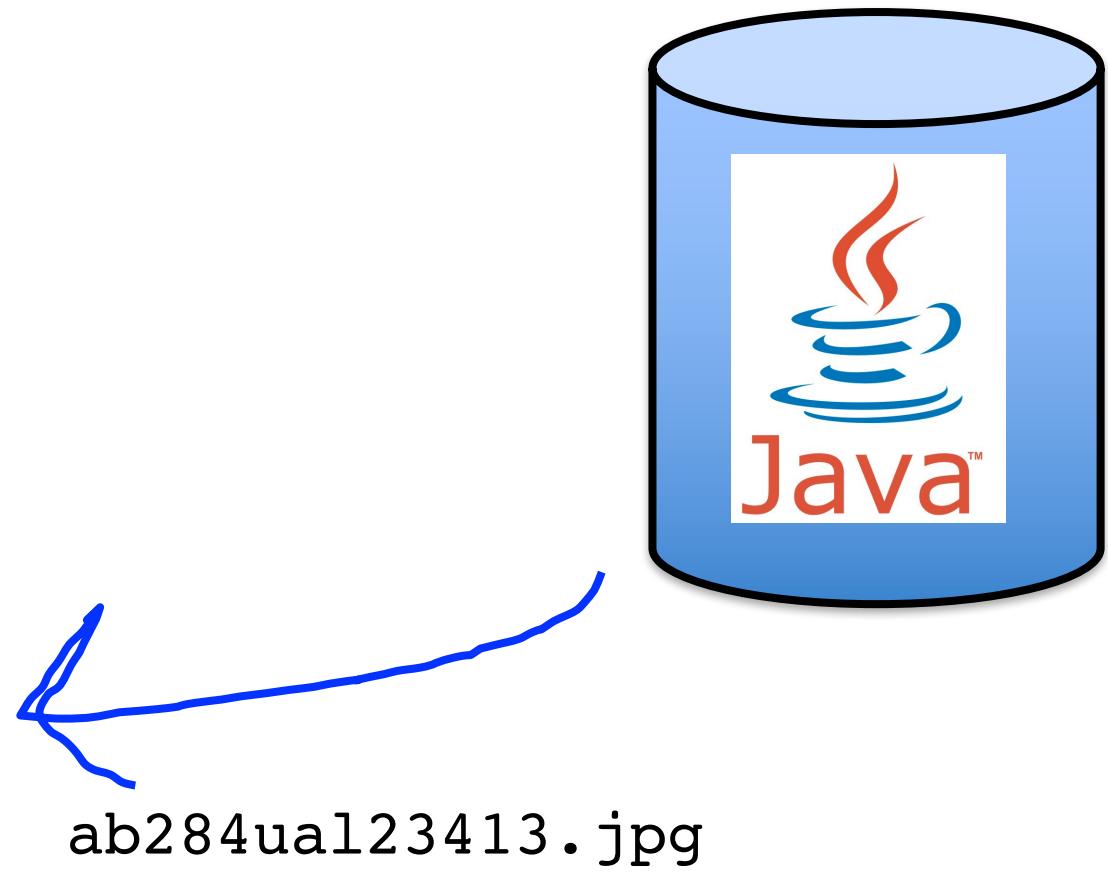
Params:

email=piech@cs.stanford.edu

Face Book Server



# Face Book Server

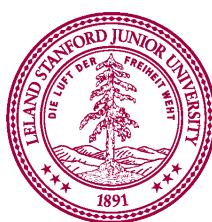
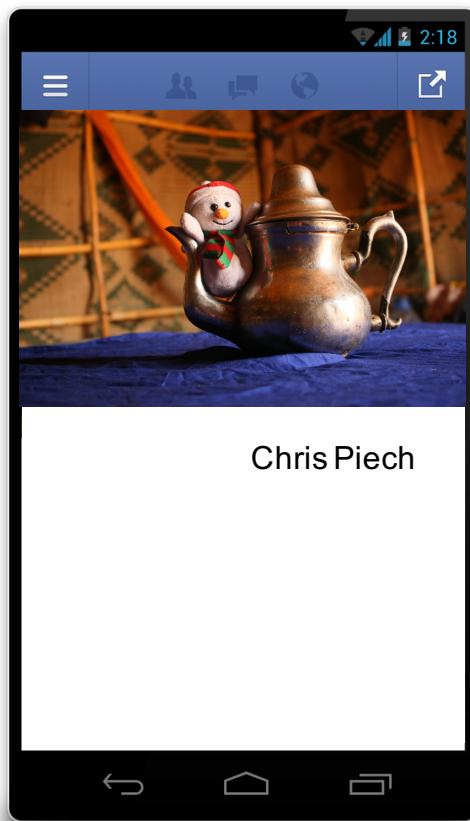
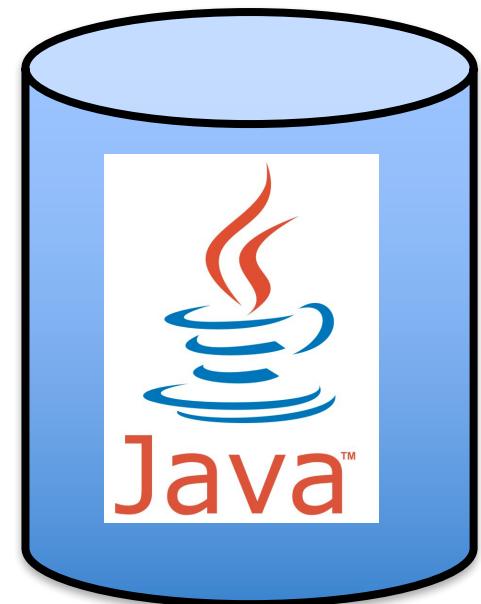


Command: getImage

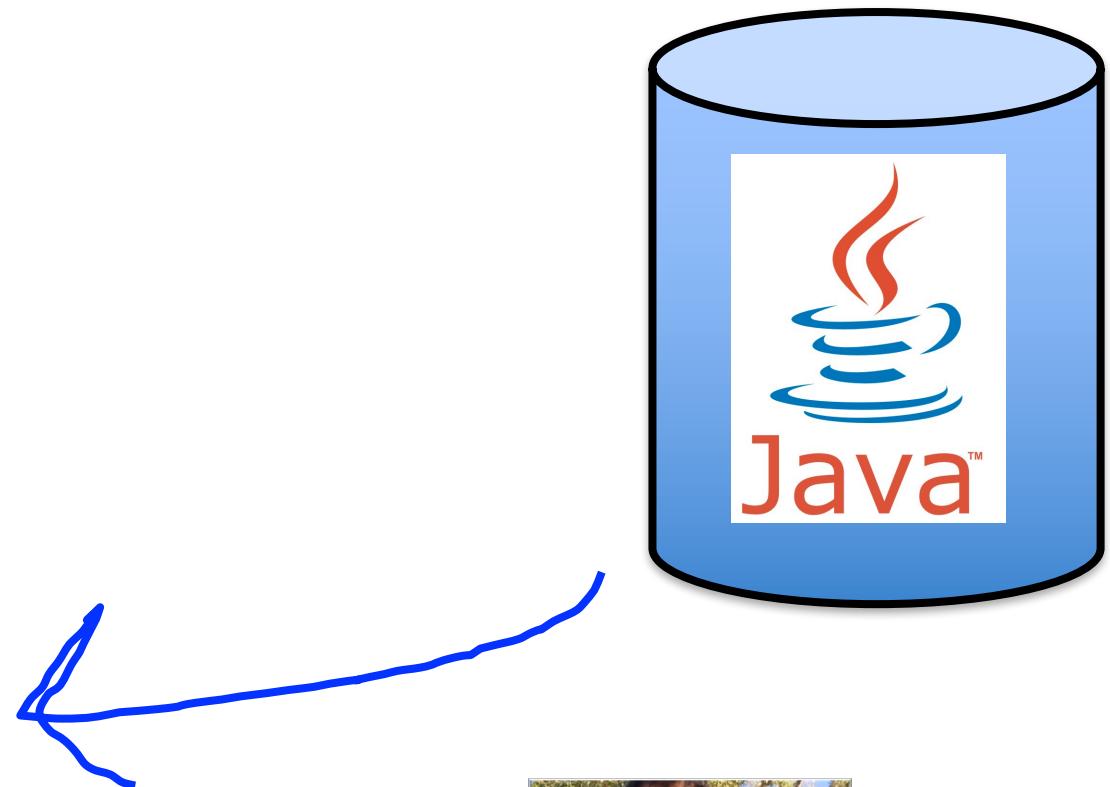
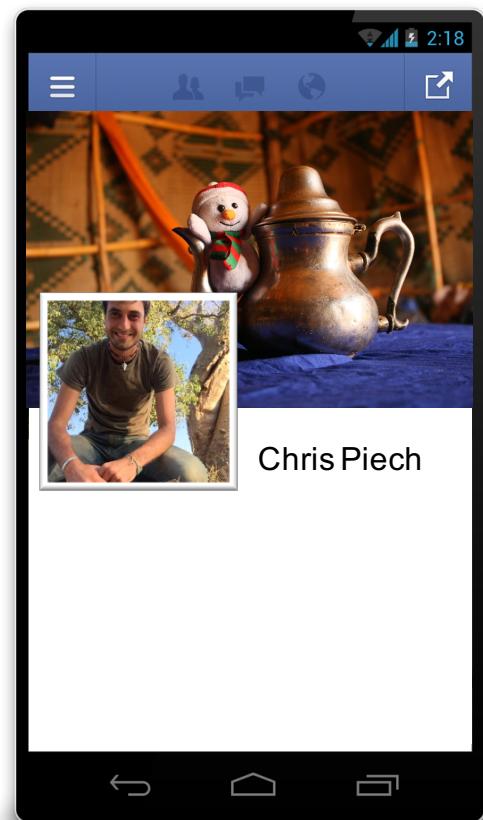
Params:

fileName=ab284ual23413.jpg

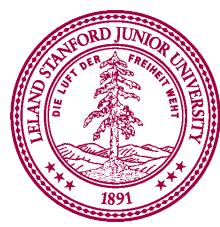
Face Book Server



# Face Book Server



Piech, CS106A, Stanford University



# Face Pamphlet

## FacePamphletServer

```
FacePamphletServer
Starting server on port 8000...
addProfile (name=Mehran)
  => success
addProfile (name=Chris)
  => success
addProfile (name=Chris)
  => Error: Database already contains Chris.
getStatus (name=Chris)
  => none
setStatus (name=Chris, status=teaching)
  => success
getStatus (name=Chris)
  => teaching
addFriend (name2=Mehran, name1=Chris)
  => success
getFriends (name=Chris)
  => [Mehran]
addProfile (name=Julie)
  => success
getImg (name=Julie)
  => none
getStatus (name=Julie)
  => none
getFriends (name=Julie)
  => []
setImg (img=JulieZ.jpg, name=Julie)
  => success
getImg (name=Julie)
  => JulieZ.jpg
getStatus (name=Julie)
  => none
getFriends (name=Julie)
  => []
addFriend (name2=Chris, name1=Julie)
  => success
getImg (name=Julie)
  => JulieZ.jpg
getStatus (name=Julie)
  => none
```

## FacePamphletClient

FacePamphletClient

Name

Chris

  
Chris is teaching

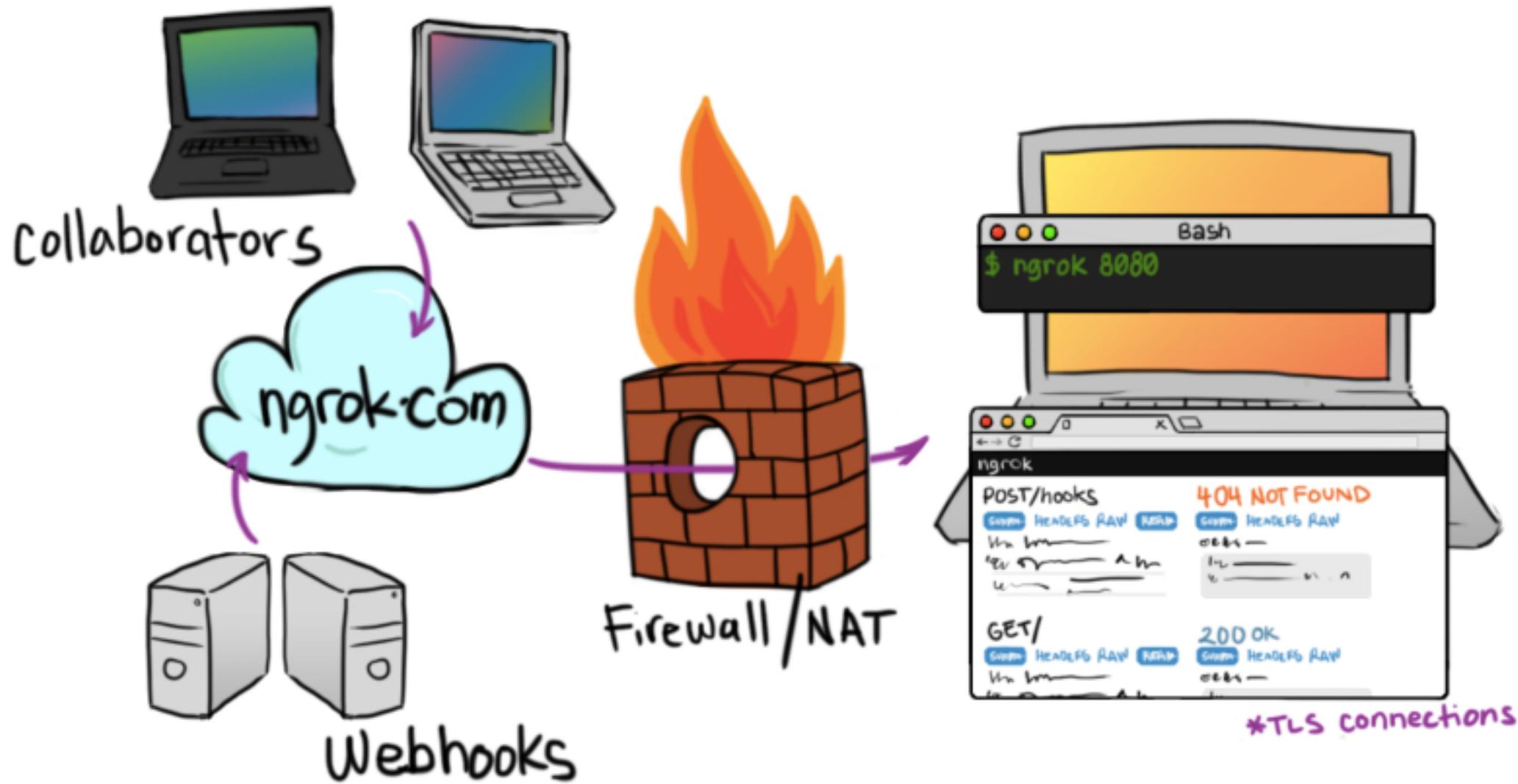
Friends  
Mehran  
Julie

Julie added as a friend.



Localhost forever?

# Use ngrok to get a url



# Want to learn more?

CS144 Computer Networking



Or CS193P

Or CS193A

Or CS108

Piech, CS106A, Stanford University



The end