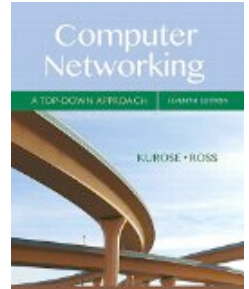# COMP 375: Lecture 04

- **News & Notes:**
  - Congrats to "Team Fluff," winners of Puppy Bowl XIV!!!
  - Lab today @ 2:30PM
  - Quiz #1 in class Wednesday
  - Project #1 due in one week
- **Reading (Due: Wed, Feb. 7)**
  - Section 2.2.{4-5} (HTTP)

Beginning-of-class Discussion

*How would you explain the **relationship** between the **Internet** and the **Web**?*

Section 2.1

NETWORK APPLICATIONS

In a distributed system you have multiple copies of each component.

What happens if one of our components/computers fails?

If a process sends a message, can it tell the difference between a slow link and a delivery failure?

A.	Yes
B.	No

What should we do if we experience a **partial failure**?

Under what circumstances, or what types of distributed applications?

A.	If one process fails or becomes unreachable, switch to a spare.
B.	Pause or shut down the application until all connectivity and processes are available.
C.	Allow the application to keep running, even if not all processes can communicate.
D.	Handle the failure in some other way.

Expected failure leads to the need for easily replaceable components.

Applications with an entity that handles requests modeled as client/server.

Client ⟷ Internet ⟷ Server
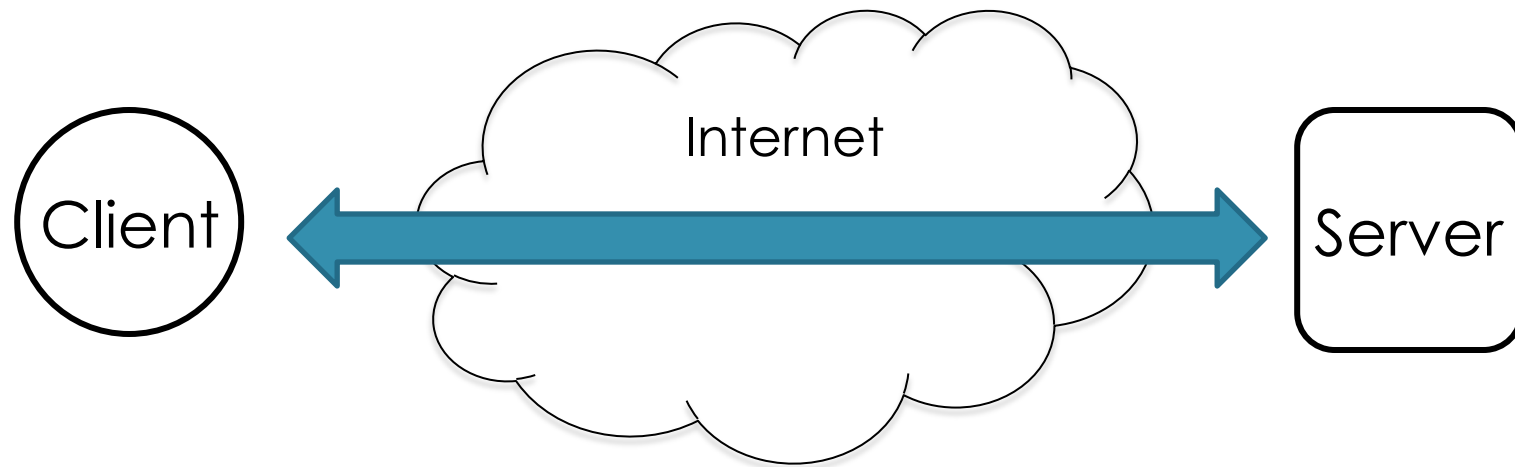
A client's properties are the inverse of a server's properties.

- **Connectivity**: Constant vs. intermittent

- **IP Address**: Static vs. dynamic

- **Ports**: Well known vs. ignored

- **Peer communication**: Yes vs. no

Apps that have no central authority are modeled as peer-to-peer (P2P).

Peer

Internet
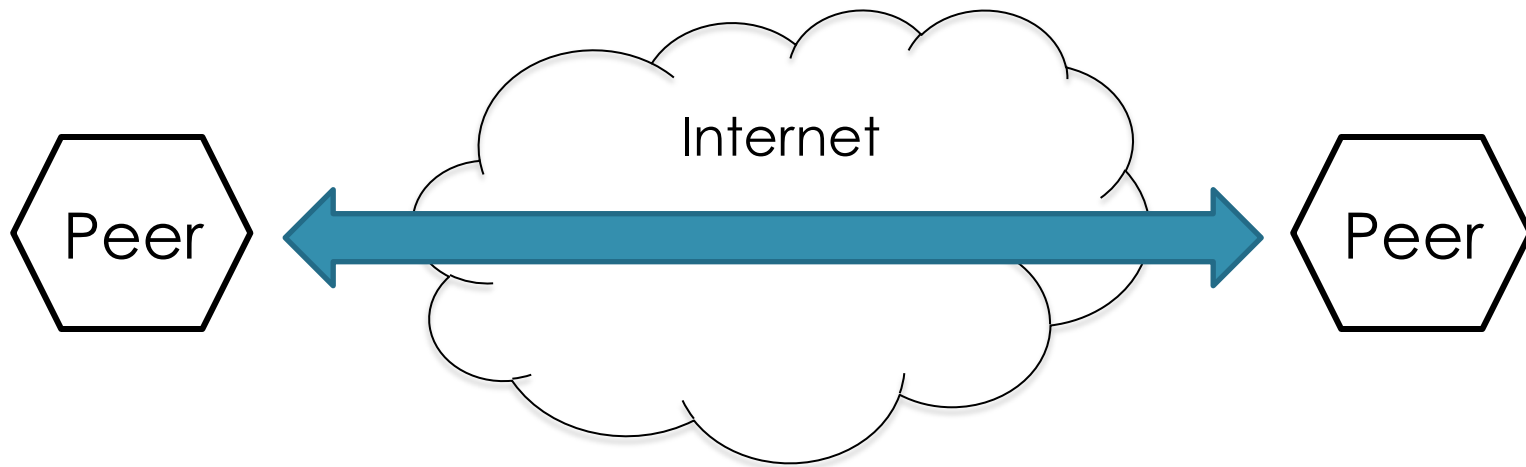
Peer

In a peer-to-peer architecture, are there clients and servers?

A.	Yes
B.	No

Which is more scalable?

A.	Client/Server
B.	Peer-to-Peer
C.	They are equal

P2P trades complexity for *scalability.*

Internet

Peer ⟷ Peer
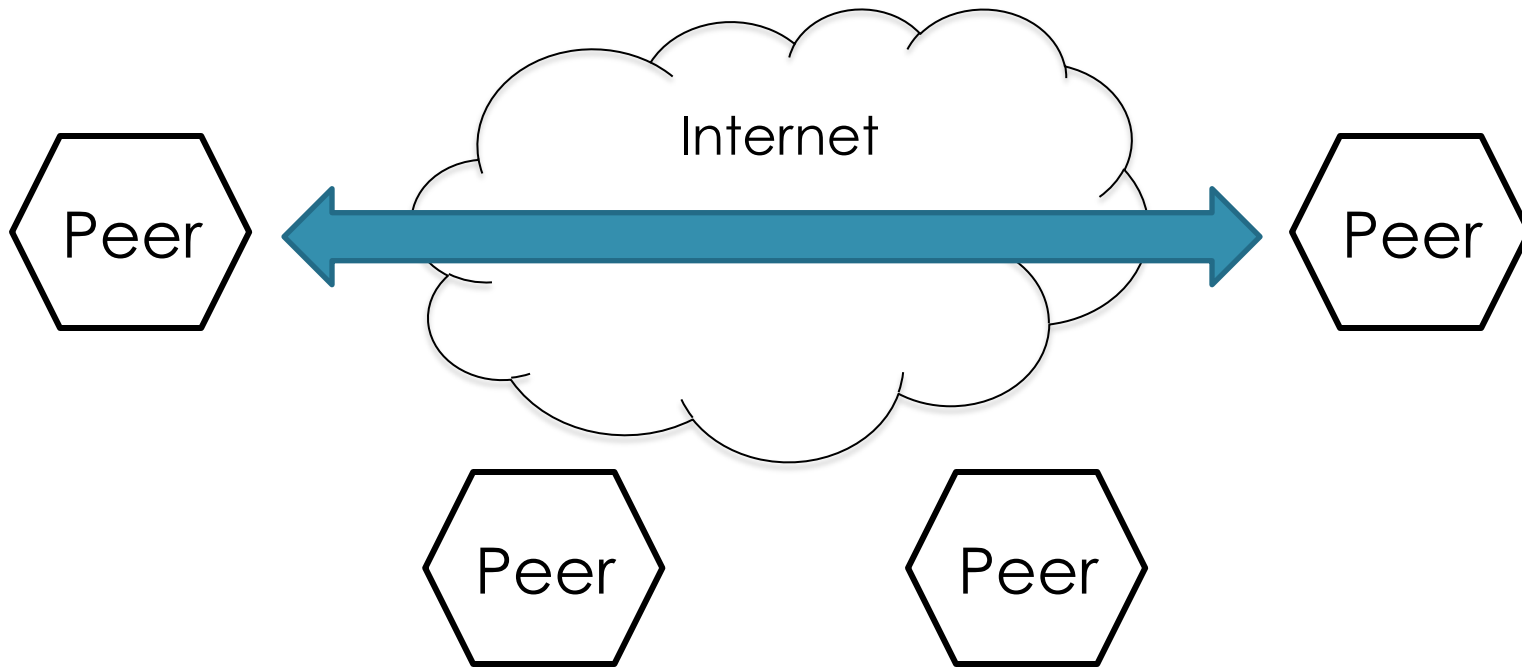
Peer

Peer

What is good about adding new peers?
What is bad?

Think
Pair
Share
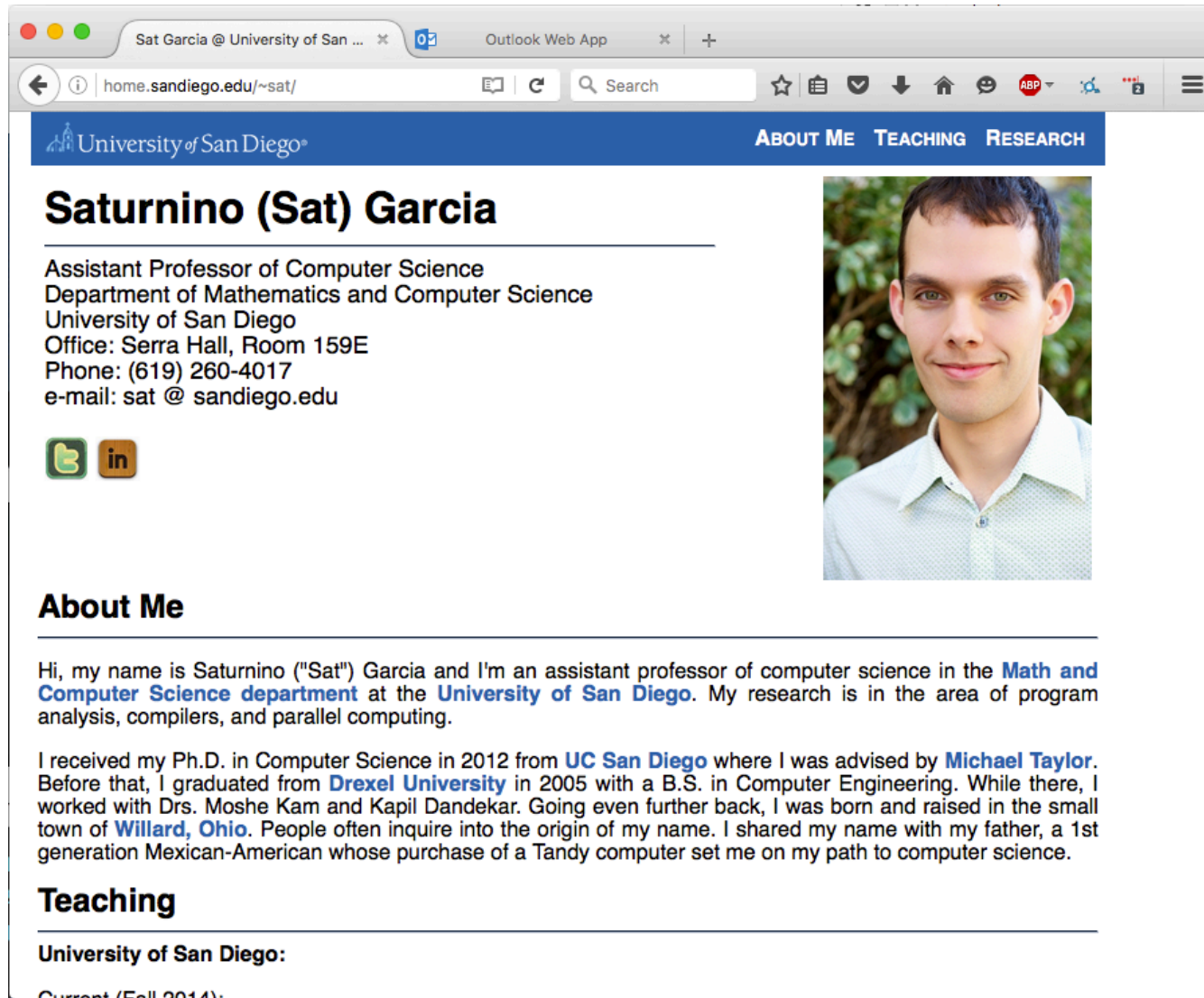
Which application is **LEAST** suited to a P2P architecture?

A.	A service that provides videos
B.	A service that provides news and discussion
C.	A service that provides hostname resolution
D.	A service that implements stock market trades

Section 2.2

THE WEB & HTTP

A Web Browser is many things, including an HTTP client.
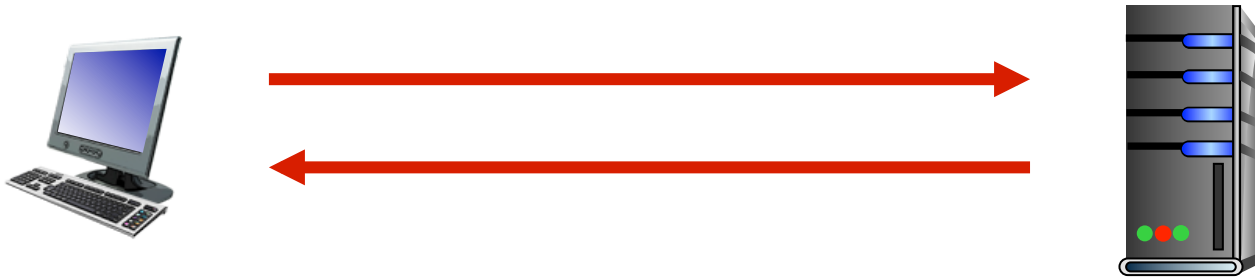
First, the user specifies a URL.

`http://some.host.name.tld/directory/name/file.ext`

Second, the browser initiates a TCP connection with the server.

- Looks up "some.host.name.tld"
- Calls connect()

Third, the browser sends an HTTP request message for the data.


```
GET /~sat/index.html HTTP/1.0
Host: home.sandiego.edu
[other optional fields, for example…]
Mozilla/5.0 (Macintosh; Intel Mac OS X 10.11;
  rv:47.0) Gecko/20100101 Firefox/47.0
Accept-language: en-US,en;q=0.5
```

Fourth, the server sends an HTTP response message.


```
HTTP/1.0 200 OK
Content-Type: text/html
Content-Length: 10477
Date: Fri, 02 Sep 2016 21:26:38 GMT
[Blank line]
(Data data data data…)
```

Finally, the browser renders the data, possibly requesting additional objects.