CS5412: HOW IT WORKS

Lecture II

Ken Birman

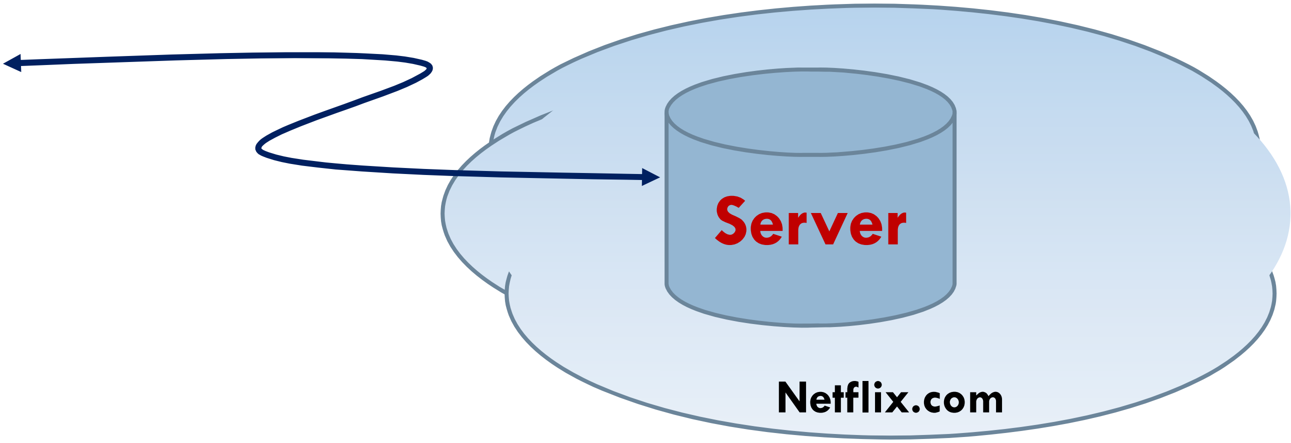
Today: Let's look at some real apps

- We'll focus on two very standard examples
  - Netflix movie player
  - Siri, Apple's new digital companion service

- How are these built?
  - What issues arise on the client platform?
  - What about in the Internet?
  - How is the cloud computing side structures?

Netflix App: Version 0

- Plays movies on demand on a mobile device



Simplest design?

- Web Services standards
- Netflix owns the data center
- Uses a fairly standard server

Version 0: Encounters issues

- □ Hard to compete with companies that already own massive cloud infrastructures (Google, MSN, etc)
- □ Web Services standards were for downloading web pages, must adapt them for video streams
- □ How can we determine that the user is legitimate?

Options for connection

- Based on the Web Services standards:
  - Transmits web pages that contain SOAP requests: Simple Object Access Protocol. Request could be "play movie"
  - The pages and responses are themselves encoded in HTML . Requests and responses are sent using HTTP(S)
  - Data is moved over a TCP connection (can be insecure for HTTP, or use the SSL security layer for HTTPS)
- Dialog with the DNS maps Netflix.com to a list of IP addresses. Client picks one
- On arrival, Netflix load balancing policy routes request to a particular server within the data center

Options for movie streaming

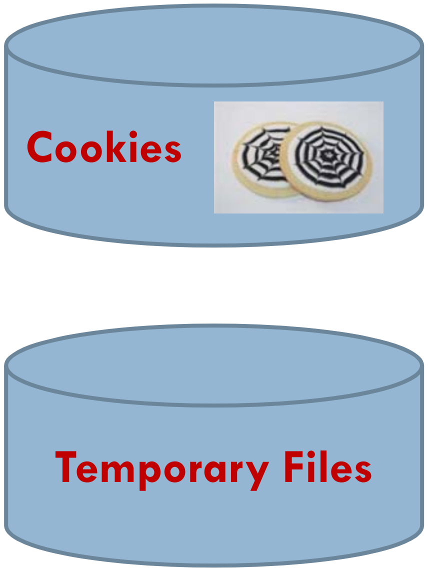
- We could fetch segments of the movie as if they were long web pages and render “incrementally”
  - This is what Netflix actually does
  - Convenient for users who jump around in the film
- We could establish an RSS connection, in which case a series of web pages can be transmitted by the server, page by page
- We could use a specialized streaming protocol called AtomPub designed to improve performance

Building the App

- An "App" is an application that runs in a browser
  - Typically, browser is told to disable its menu options and not display a border
  - Gives the illusion that the App is a dedicated application, yet in fact it can access the full power of the underlying browser framework
- Video player App?
  - A browser plug-in designed to work in many kinds of browsers
  - Would have interactive API ("pages") and also a player component (code) that has plug-in "Codec" modules for the movie format(s) Netflix supports

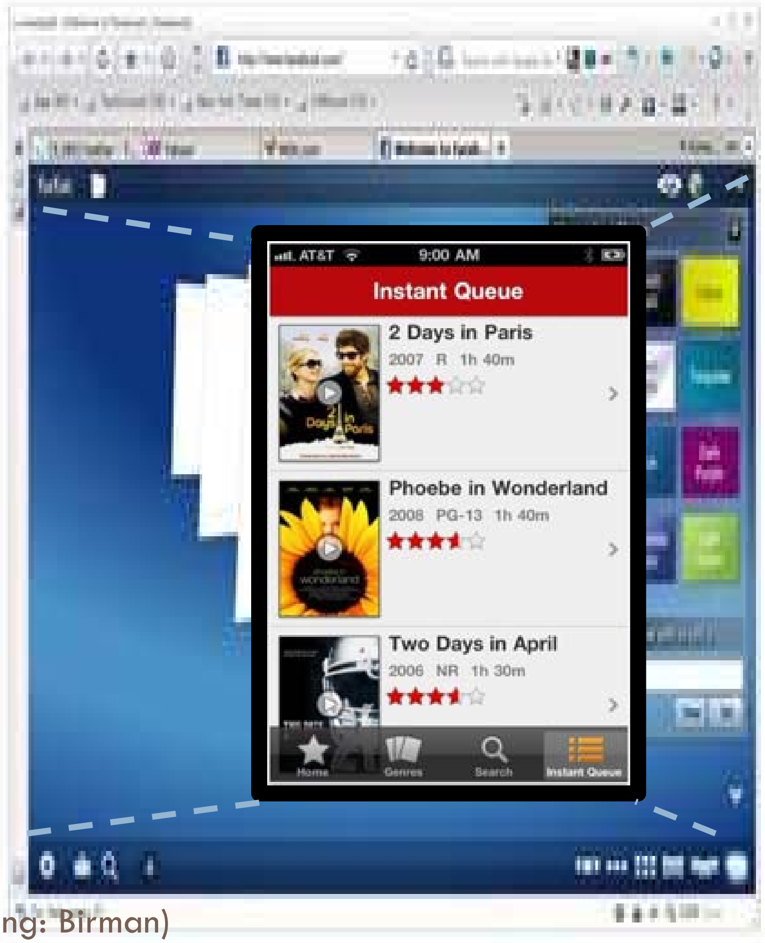
Picture of an App

- □ Browser itself is a complex machine that renders pages but can also run code



Cookies

Temporary Files



Instant Queue

2 Days in Paris
2007 R 1h 40m

Phoebe in Wonderland
2008 PG-13 1h 40m

Two Days in April
2006 NR 1h 30m

A browser is a "virtual machine"

- A kind of mini-operating system
  - Web pages are the programs (and they can contain real executable code)
  - Has various policies for which pages can access or create which cookies (must be from same site), which files, conditions under which user must click "ok", etc
  - Intent is to protect applications from one-another and also user from malfunctioning application

- Browser security: an arms race against functionality
  - Most browsers have vulnerabilities and some sites use them benignly (e.g. circumvent popup block) or maliciously
  - *Why is web full of free porn?* Hint: Many free porn sites have code designed to seize control of your machine

Browser complexity

- Browser is multithreaded and can do many things concurrently
- One page can have many frames, each with its own security context, and each independently active
- Can execute code such as Adobe Flash, Javascript, AJAX, Silverlight, CAJA
  - Some code downloads silently with web page
  - Other code must be installed as a "plug in" and gets access to broader browser functionality
  - A plug-in "extends" the browser with new functions

Popups

- Rendered content can generally render in
  - Frame that created the content
  - Parent frame ("entire page")
  - A new frame that runs as a new tab
  - A new frame that runs as a new page
- Frame has an associated security context (site) and can only download or upload from that site
  - But since downloaded page can have new frames, and one site can map to many places, limitation isn't very meaningful
- If frame also controls web page borders can be hard to understand interaction as being "pages"; looks more like a "live window" GUI

Cookies

- Cookies store history and other data
  - A file in a standard HTML format
  - Many possible fields, and application can add more
- Browser provides the cookie for Netflix.com when connecting to Netflix.com
  - To avoid huge cookies, some sites have multiple cookies with subdomain names
  - Browser prevents BadGuysAreUs.com from seeing the Netflix.com cookie as a security measure

Files

- When content is downloaded from the Internet, the browser "quarantines" it by placing it in a secure area of the file system
- Intent is that only application that downloaded a file can access it, and that files can only be created or read from this safe part of the file system
- User has a degree of control over downloaded content but might be surprised at how much of this there is, and what it could contain!

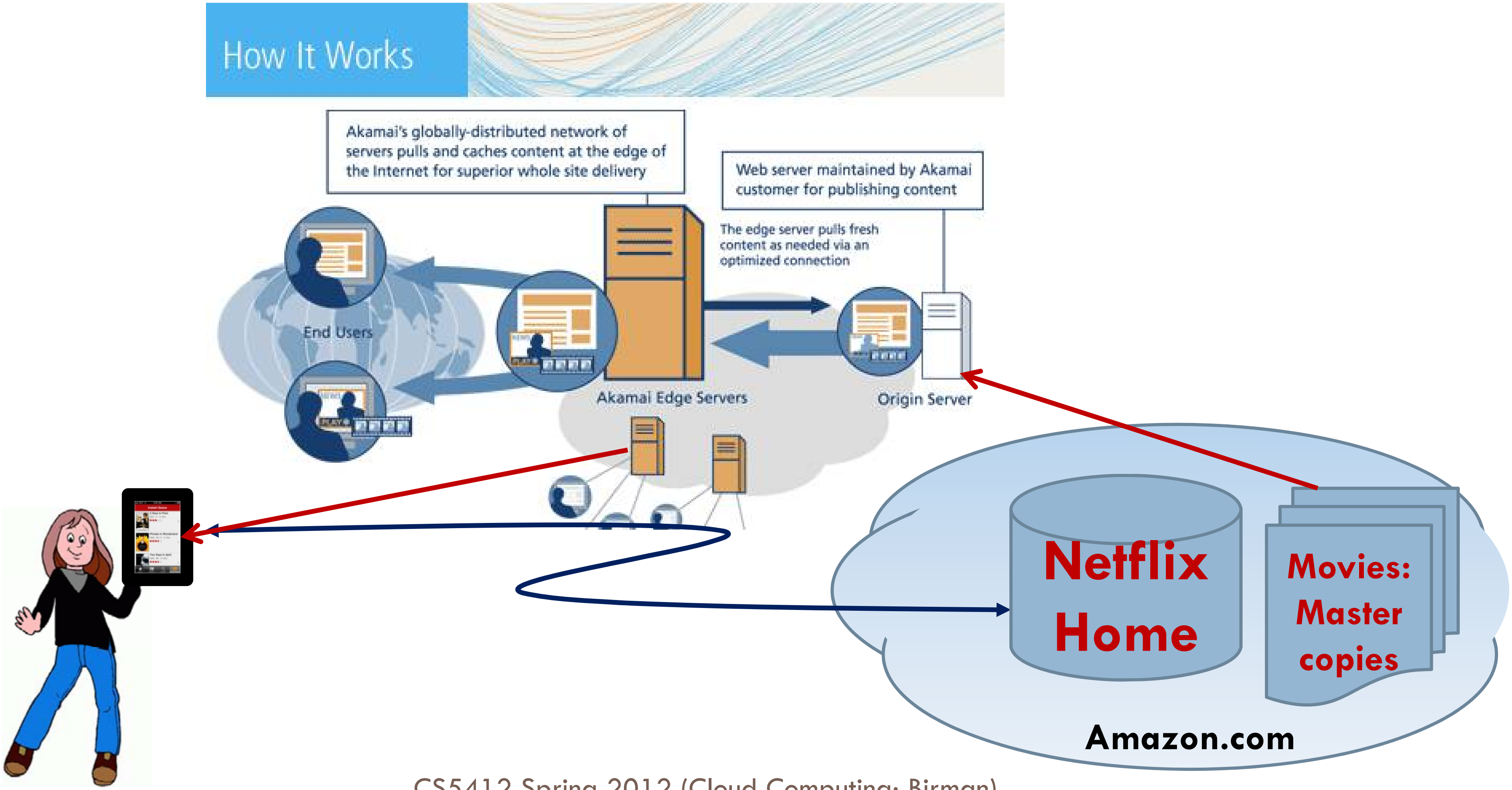
Localization

- Our Netflix.com application wants to stream data from:
  - A nearby data center
  - Within it, a machine with the right content
  - Among those, one that has light load
- But Netflix won't want to build its own nationwide collection of data centers!
- Leads to Netflix "version 1"

Netflix “outsourcing” components

- Think of Netflix in terms of main components
  - The API you see that runs on your client system
  - The routing policy used to connect you to a data center
  - The Netflix “home page” service in that data center
  - The movie you end up downloading
- Netflix 1.0 breaks the solution into parts
  - Builds each of these aspects itself
  - But then pays a hosting company to run each part, and not necessarily just one company!

Netflix Version 1



How It Works

Akamai's globally-distributed network of servers pulls and caches content at the edge of the Internet for superior whole site delivery

Web server maintained by Akamai customer for publishing content

The edge server pulls fresh content as needed via an optimized connection

Features of new version

- Netflix.com is actually a “pseudonym” for Amazon.com
  - An IP address domain within Amazon.com
  - Amazon’s control over the DNS allows it to vector your request to a nearby Amazon.com data center, then on arrival, Amazon gateway routes request to a Netflix tier-one cloud service component
  - The number of these varies elastically based on load Netflix is experiencing
- Amazon AC3 used to host the master copies of Netflix movies

Akamai

- Akamai is an example of a "content distribution service"
  - A company that plays an intermediary role
  - Content is delivered to the service by Netflix.com (from its Amazon.com platform)
  - Akamai makes copies "as needed" and distributes them to end users who present Akamai with appropriate URLs
- Netflix.com (within Amazon.com) returns a web page with "redirection" URLs to tell your browser app what to fetch from Akamai

ARL (Akamized URL)

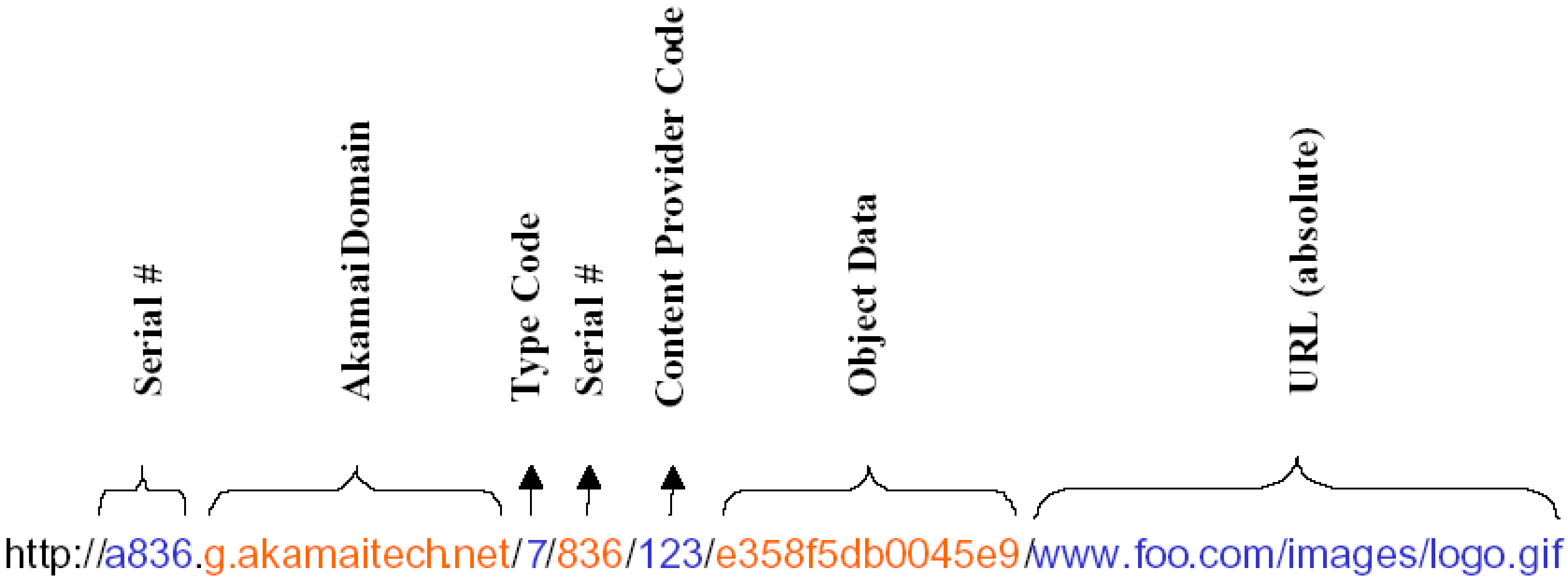


Image from Akamai's white papers

A few options…

- With Akamai, the ARL encodes information about what the user seeks and how to find it

- Netflix.com page would be generated to contain these Akamai ARLs using software Akamai provides

- But there have been several solutions to this problem (we won’t get detailed due to time limits)

Netflix worry: Theft!

- Digital movies cost a lot of money for Netflix

- Can't risk that people might steal them from within Akamai or Amazon by knowing the URLs

- So Netflix uses a cryptographic encoding scheme!
  - Every movie is enciphered using AES 256 coding
  - To decipher a movie, player must have the key

Sending key to user

- We can’t just send it in plain text
  - Anyone on the web might see the page go by

- Could use HTTPS (runs on the SSL standard)
  - The user’s system must log into Netflix.com.
  - We identify (“authenticate”) the user and verify that this user is allowed (“authorized”) to access this movie
  - Web pages sent over SSL use negotiated end-to-end security certificates (again, AES 256) hence are safe against intrusion

- So: we send the key in the web page with the ARL!

Notion of "closest" matters

- We want to direct the user's request to the closest Amazon.com (Netflix.com) data center
- Later want to stream the movie from the best choice of Akamai data center
- But what should *closest* mean within the web?

Options for identifying location

- We could use GPS location
  - Most mobile devices now support GPS radio
  - They can synchronize time to within ~12ms and location to within ~100 meters, improves over time
- We can sense local wireless networks
  - There are companies that maintain databases
  - Can triangulate locations using wireless SSIDs even if the wireless network itself is a secure one
- We can ping “landmark” sites and triangulate

Pros and cons…

- Laptop computers might lack GPS information
- Anyhow GPS location may not predict network access routes: closest "within" network could be physically far away

- In Ithaca, the route from my home to work (2 blocks) used to route through New Jersey!
- 12 hop routes are very common

Pros and cons…

- □ Triangulation schemes work well but not for all parts of the world. Basically, ping each option
- □ Need to be reasonably "centered" relative to landmark sites
- □ So good for finding best Akamai data center, but bad for predicting performance between two computers both in Ithaca

Pros and cons…

- Sensing wireless networks and IP route "upstream"
  - A great way to augment GPS data, widely used by mapping applications
  - Can often figure out which side of a hotel I'm in
  - Permits "precomputing" of best data center to use

- This is a very popular approach today because if we can precompute the data center we can *control* which data center a particular user will route to!

Identification options

Netflix identifies you

- IP address: not a good option (mobility)
- Your name, password automatically sent (old style Web security, but very easy to compromise)
- Login info (annoying)
- Biometrics, portable keychain "dongle"
- Cookies remember login

You identify Netflix

- Netflix.com publishes the "credentials" services it uses
- You go to such a service via a secured chain of certificate services
- It provides a public key for Netflix
- Key is public… Netflix has the matching private key. Thus Netflix can prove that you've got right site

Shazam: Flipping it around

- Shazam samples music and then tells you what the song is. It does the "reverse" of what Netflix does:
  - Captures fragment of music, sends it to the cloud
  - *Precomputed* search indicies are used to speed up the search, with a fine-grained match to improve accuracy

- Less common now, but coming soon: massive parallel search in real-time within a scaled-out cloud service
  - Today's cloud is good at rapid replies but multiple machines rarely cooperate to compute an answer in real-time

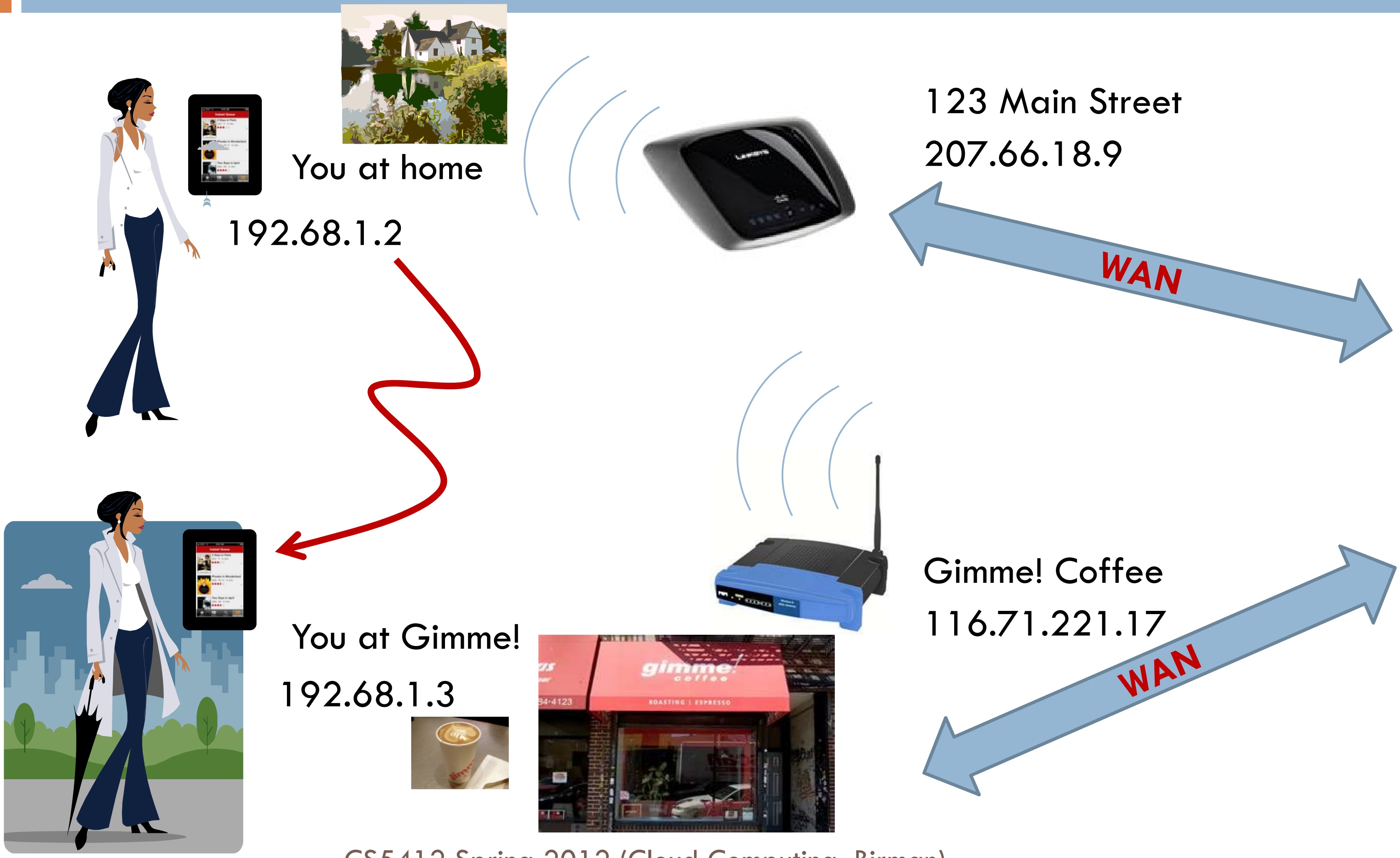
Mobility

- Suppose you grab your laptop and head down to the local coffee shop

- Reopen laptop. Can your movie still play?

- Problem: IP addresses change and in fact the best data center choice could differ!

Network address translation



You at home

192.68.1.2

123 Main Street

207.66.18.9

You at Gimme!

192.68.1.3

Gimme! Coffee

116.71.221.17

What about Netflix.com

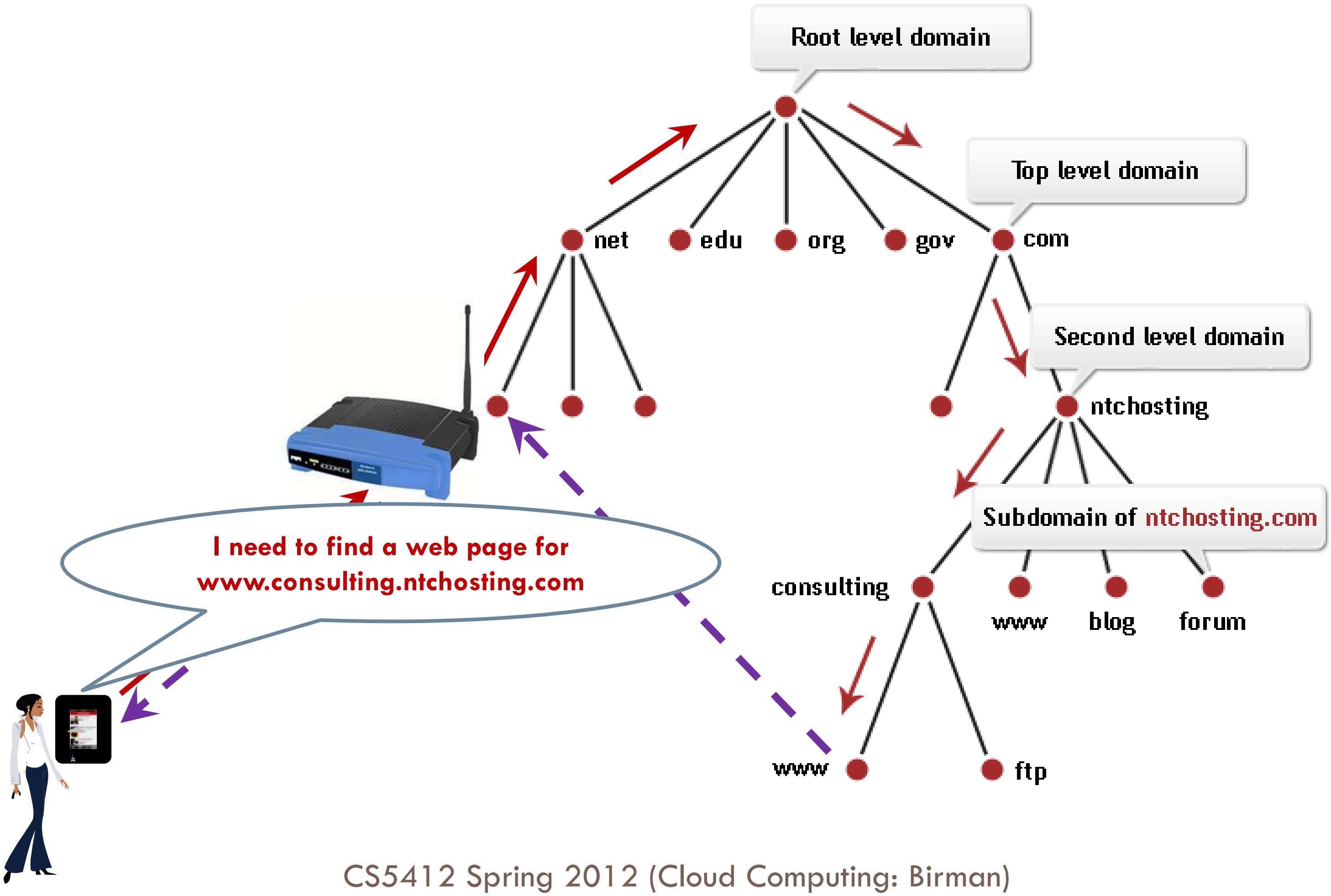
- Netflix.com is really hosted by Amazon.com
- And Amazon.com operates many (dozens) of major data centers
  - Each of which has at least two IP addresses (multihoming) to protect against ISP failures
- So as you move about we see changes:
  - IP address / port change (Network address translation)
  - Best data center to connect you to

Will your connection break?

- Perhaps so
  - If you fetch minutes 20-26 of “Sleepless in Seattle” you won’t care if the source changes!
  - In this sense you don’t have a connection

- Perhaps not
  - We can hide the breakage of the TCP connection and give the illusion of continuity
  - Can even “tunnel” TCP over a changing connection

Resolving a web address



Routing control

- Netflix, Akamai and Amazon all have opinions about which data center is best for you!
  - Netflix: concerned about "affinity" (some server may have best data for your account)
  - Amazon: concerned about load, nearness of data center
  - Akamai: same issues as Amazon but also cares about which of its servers have the movie online

Ways of doing that control

- A site name like “Netflix.com” is mapped by the DNS (Domain Name Service) to an IP address

- This mapping is under control of Netflix.com itself
  - The “authoritative” mapping service for the domain
  - Each mapping has an associated lifetime
  - DNS uses mapping until it expires, then refreshes
  - DNS itself is a hierarchical structure

- Thus Netflix can decide which mapping to use

In fact, Amazon plays this role

- Part of the Amazon web services architecture is to provide a service to Netflix.com that lets
  - Netflix advise Amazon on preferred routing
  - Amazon decides what routing customers “actually” see

- Akamai, in contrast, runs its own show because Akamai isn’t pretending to be Netflix.com

- Tradeoff: How frequent are these updates? Quality of routing versus load on the DNS

Netflix Version 2... 3... 4...

- Each factor of 10x brings new challenges
- Netflix ends up
  - Owning some infrastructure (like their own high-speed links to Latin and South America)
  - Renting some too
  - Building some of their own stuff... using some existing stuff...
- Constantly forced to rethink everything

High assurance

- Could mean many things!
  - Netflix assurance that videos won't be stolen and that only authorized people can play them (more generally, that intruders can't steal private stuff)
  - The additional assurance that App won't turn user's machine into a bot in some massive botnet
  - Technical assurance that playback won't be interrupted
  - Overall goal of picking best binding for each request
  - Fault-tolerance: need to automatically handle things that might happen while system is operational

What might our user desire?

- She doesn’t want her movie history to be public
- She isn’t at all eager for Borat to track her around the city or to use her mobile devices to spy on her
- She worries about whether her bank account is safe, and whether she is at risk of “identity theft” when she makes an online purchase
- She depends on email yet email doesn’t always get through reliably. And she thinks of it as private yet people on the network could read it
- She feels overwhelmed by inappropriate pop-up ads, spam, junk web pages, and the slightest typo can take her to a web site that “mimics” the real thing

Does she have a right to more?

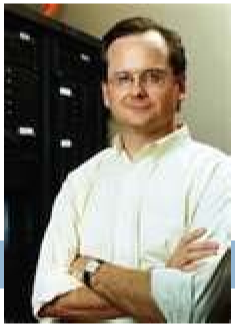
- Do we have a right to privacy in the Internet?

- If we're harmed by information disclosed by a third party, can we be repaid for the damage?

- If she's pulled over, can a policeman demand to see her GPS data for the past half hour?

- Would it be stalking if her ex-husband installed a keyboard tracker on her PC?

Lawrence Lessig



- Writes about "East Code" versus "West Code"
  - East code is about laws and regulations
  - West code is about new cyber technology ideas
- He argues that we're seeing technology get ahead of the law and that this is having a powerful impact on society in many dimensions

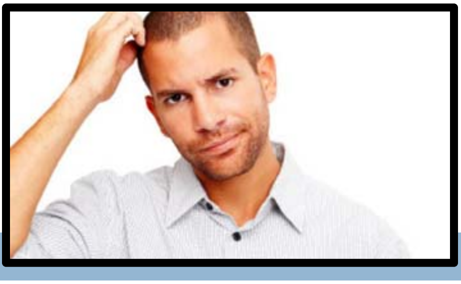
Show me the law and I'll enforce it!

Am very much liking Internet! Is like bedroom window with no curtain!

CyberWar

- Richard Clarke, past national security advisor, worries about the risk of a "Cyber War"
  - An adversary might wait until we depend upon all sorts of technical systems, like a smart power grid
  - Then attack us by disabling those systems
  - Could cause physical damage: he envisages destruction of the power grid, plane crashes, bank failures
- "We can do it. They can too."
- Also says that Chinese cyberespionage has broken into huge range of government/military/high-tech systems

How do they get in?



- We can't even trust the hardware!
- Modern client platforms are exposed at <u>every level</u>
  - If the hardware had "extra" built-in computers that watch the ones we run on, the network, etc, we couldn't tell.
  - The network could be "virtualized" (think of the inter-frame data in Sagan's book/film "Contact")
  - Operating systems often have vunerabilities. USB viruses exploit the way that operating systems display disk drives.
  - The applications that run on them invariably have gaping holes.
  - Systems are misconfigured. Users pick terrible passwords, then leave them lying around. We send sensitive data in plain text
- How could we even dream of entrusting the Internet with our most sensitive and important stuff?

Cloud: Helps, and hurts

- Moving data into data centers could help
  - More standard, easier to manage
  - Could use “synthetic diversity” to repell attacks

- But also can hurt
  - Today’s cloud systems are very weakly secured
  - Insider has almost unlimited potential
  - Operators monitor main system… but what about backups and data “digesting” applications?

CEO of Sun Microsystems on this issue?

"You have zero privacy anyway. Get over it."

-- Sun CEO Scott McNealy at the launch of Jini (1/16/2009)

Google CEO Eric Schmidt

"If you have something that you don't want anyone to know, maybe you shouldn't be doing it in the first place.... The reality is that search engines do retain information... It could become available later..."

What did Steve Jobs think?

Google:

Don't be evil.

Of course, that was before Siri

- Officer: “Do you why I pulled you over today?”
- John: “I’m sorry, sir. I don’t know how fast I was going.”
- John’s Siri: “John, your speed was 82mph, and your top speed today was 91mph. Resume “text Sally”?”

We’re exaggerating… a little…

- Siri shouldn’t respond to “Sir” but mistake is plausible
  - Siri captures voice snippets on your iPhone, GPS data…
  - Ships the data to an Apple data center
  - Uses a mixture of cutting edge AI/NLP with a vast database of utterances to make sense of what you said
  - Remembers information to improve responses

- Most mobile phones have similar capabilities!

Self-censorship

- Applications like Siri force cloud computing systems to censor themselves
  - They capture personal information (lots of it)
  - And they should use it on your behalf

- But aren't supposed to "mix" personal data (e.g. John shouldn't be able to query Sally's Siri data)

- This is hard because we are still far from true "AI" that can actually understand the meaning of things

Search for your SS#

- It won't pull up any pages
  - But the reason isn't that there is nothing in the cloud that has your SS# on it
- Search engines "hide" such pages to protect us!
  - In fact huge amounts of information about us flows into the cloud today. Google does have it. So does MSN, Yahoo!, and many other "aggregators"
  - Tomorrow it will only get more extreme!
  - Danger is that the system won't know what to hide

Exposes various roads we could explore

- Could think hard about meaning of “assurance” in practice, for a world that increasingly depends on technologies that lack assurance properties

- Could ask what the law really should guarantee (and what about in other countries with other laws?)

- Could just ask if we can fix the technology at least have better properties

CS5412 approach?

- We don't have time to explore every aspect
- So we'll focus on technology
- Tomorrow, perhaps better laws will protect cyberspace. But can it be done?
  - If we had to, could we build high assurance cloud-hosted systems?
  - Stick around and find out!