CSCC09F Programming on the Web

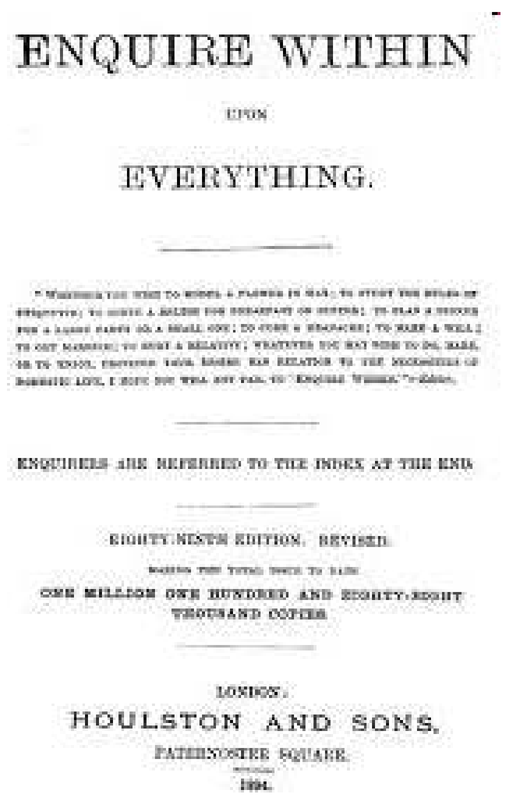
Introduction to the World Wide Web

(aka WWW, the Web)

Web Origins

- In 1989, Tim Berners-Lee working at CERN proposed an Intranet supporting document access within a heterogeneous environment

- TBL had an interest in hypertext, dating back to childhood, when he was exposed to a then-popular book, “Enquire Within”, that contained cross-reference “links” to many different topics – a sort of common-man’s how-to guide



ENQUIRE WITHIN

UPON

EVERYTHING.

"Whether you wish to model a flower in wax; to study the rules of etiquette; to serve a relish for breakfast or supper; to plan a dinner for a large party or a small one; to cure a headache; to make a will; to get married; to bury a relative; whatever you may wish to do, make, or to enjoy, provided your desire has relation to the necessities of domestic life, I hope you will not fail to 'Enquire Within.'"—Editor.

ENQUIRERS ARE REFERRED TO THE INDEX AT THE END.

EIGHTY-NINTH EDITION, REVISED.

MAKING THE TOTAL ISSUE TO DATE

ONE MILLION ONE HUNDRED AND EIGHTY-EIGHT THOUSAND COPIES.

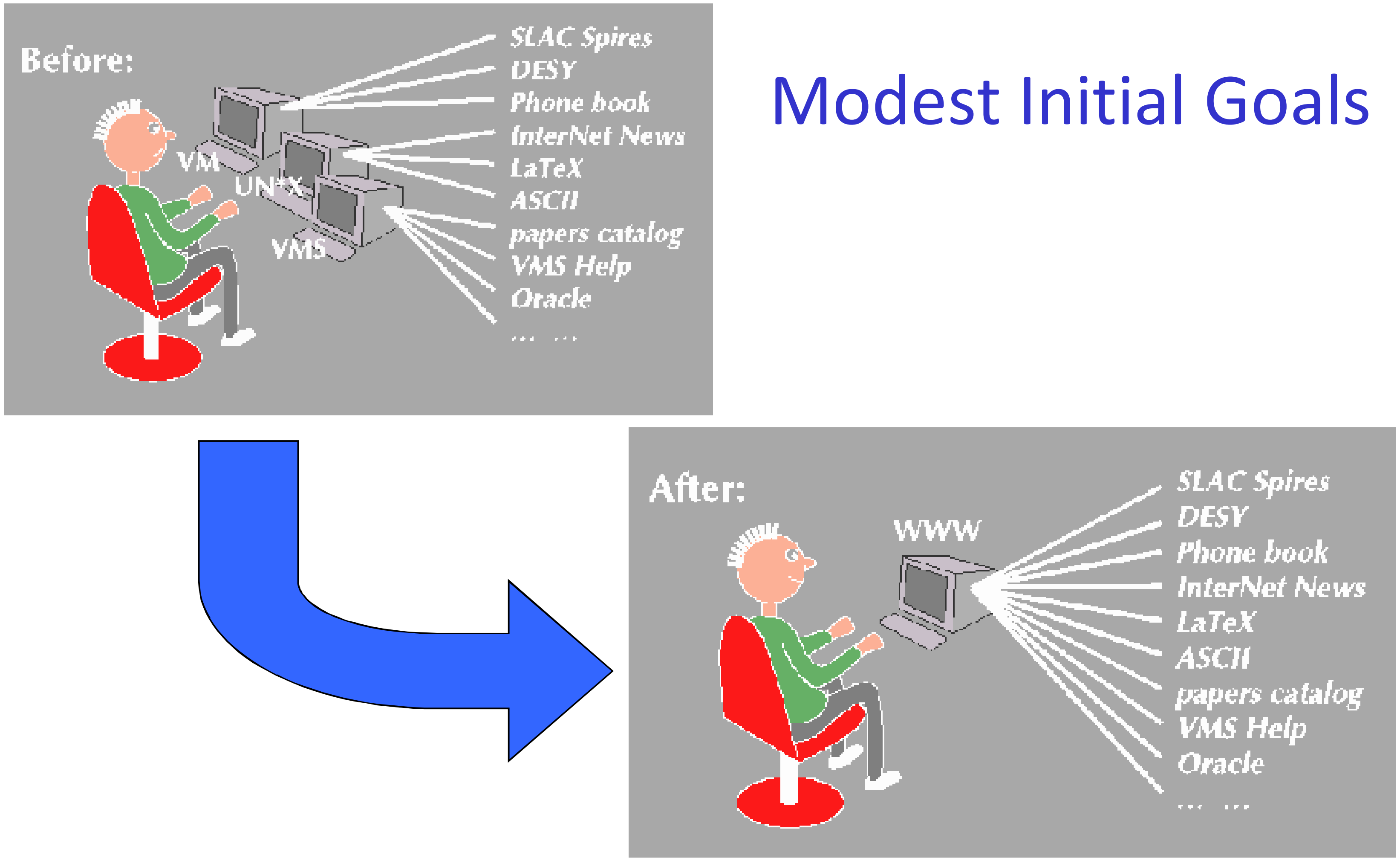
LONDON:

HOULSTON AND SONS,

PATERNOSTER SQUARE.

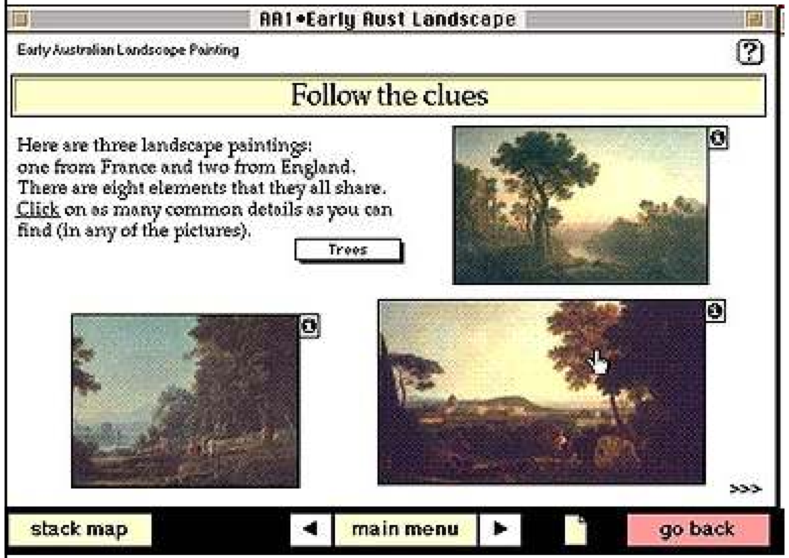
1894.

Modest Initial Goals



01 - World Wide Web CSCC09 3

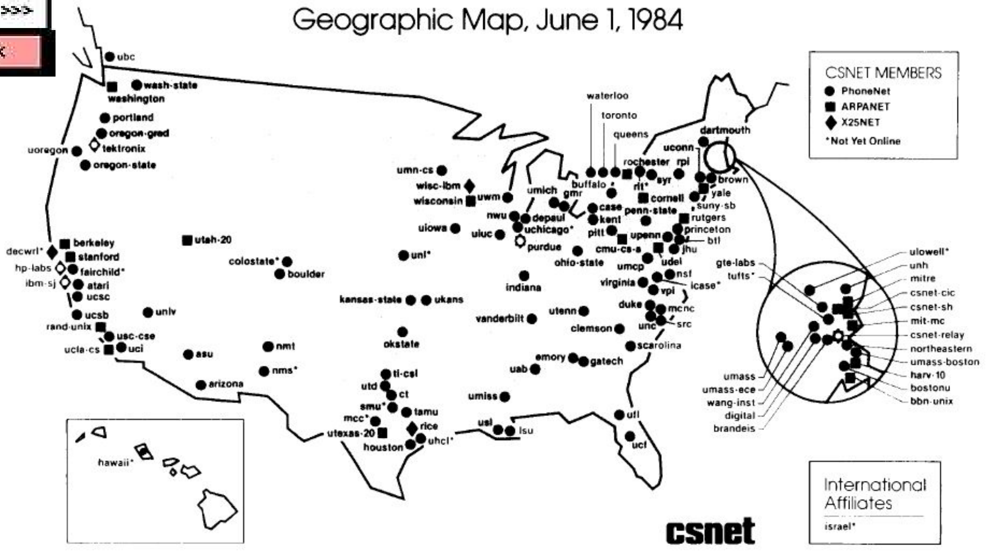
Web Origins: Groundwork



Here are three landscape paintings:
one from France and two from England.
There are eight elements that they all share.
Click on as many common details as you can
find (in any of the pictures).

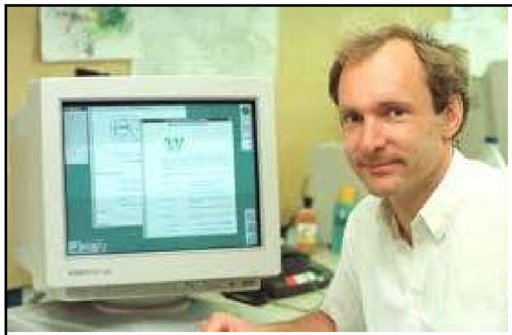
- Hypertext on computers dates back to '69, but especially work in '85 at Xerox (NoteCards) and '87 at Apple (HyperCard) which popularized the technique. Links were not networked.

- Meanwhile, the Internet was still a sleeper research & education network, unknown to the general population.

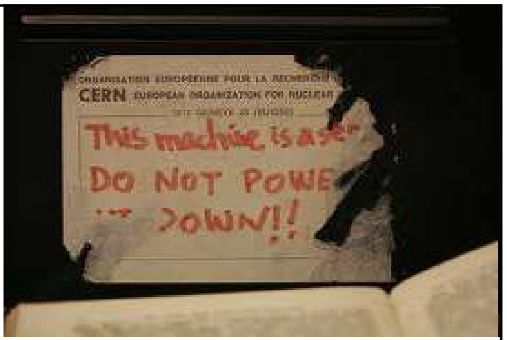


Web Origins: Groundwork

- TBL created a paradigm-shifting innovation, with considerable groundwork laid by earlier technologies:
  - SGML – a standardized "markup language" for describing document content, in use at CERN
  - hypertext – a way to link documents
  - the Internet – a standardized network framework with many existing protocols, e.g. email, ftp, telnet
  - Unix – with its regular, hierarchical file structure
  - and a large supporting cast, such as: client-server architecture, remote procedure calls (which TBL worked on at CERN), scripting languages, relational databases, etc.



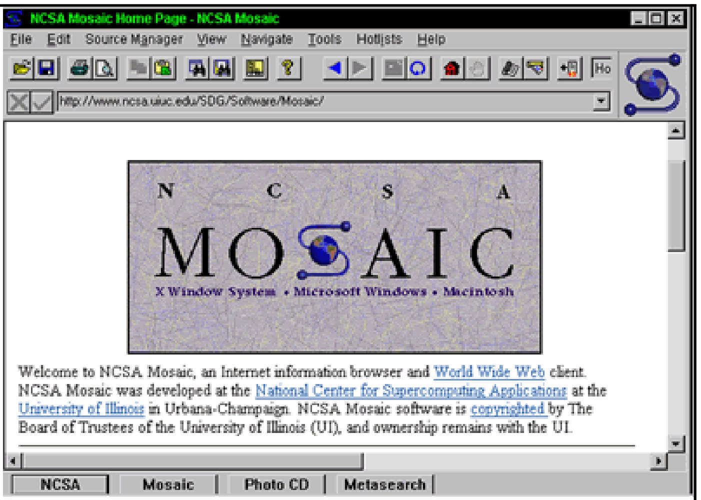
Web Origins: Tim’s Synthesis



- So, what exactly did TBL invent?
  - defined and implemented a small markup language, based on an SGML application in use at CERN, which he named HTML
  - added network-traversible hyperlinks
  - operating over a new Internet application-protocol named HTTP
  - with a uniform naming structure (URLs) based on domain names from the Internet DNS and document locations specified as in Unix file paths
  - (he also implemented his ideas as a working system on a NeXT computer running a version of Unix)
- In hindsight it may all seem obvious, but it took a stroke of insight to pull all the existing pieces together into a coherent and compelling package

01 - World Wide Web CSCC09 6

Web Origins: Out of the Lab


NCSA Mosaic Home Page - NCSA Mosaic
File Edit Source Manager View Navigate Tools Hotlists Help
http://www.ncsa.uiuc.edu/SDG/Software/Mosaic/
N C S A
MOSAIC
X Window System • Microsoft Windows • Macintosh
NCSA Mosaic Photo CD Metasearch

Welcome to NCSA Mosaic, an Internet information browser and World Wide Web client. NCSA Mosaic was developed at the National Center for Supercomputing Applications at the University of Illinois in Urbana-Champaign. NCSA Mosaic software is copyrighted by The Board of Trustees of the University of Illinois (UI), and ownership remains with the UI.

- quantum leap in Web use began with NCSA Mosaic browser:
  - image (img) tag; a multi-platform, easy-to-install and free GUI implementation -- the Web for non-geeks!
  - implemented first for X11 on UNIX(!), ports to Mac, Windows followed shortly
- evolved into a vehicle for solving a large number of important IT problems
- hey, we could make money on this … Netscape Communications kicks off the Internet gold rush aka the ".com" boom!

Web Components

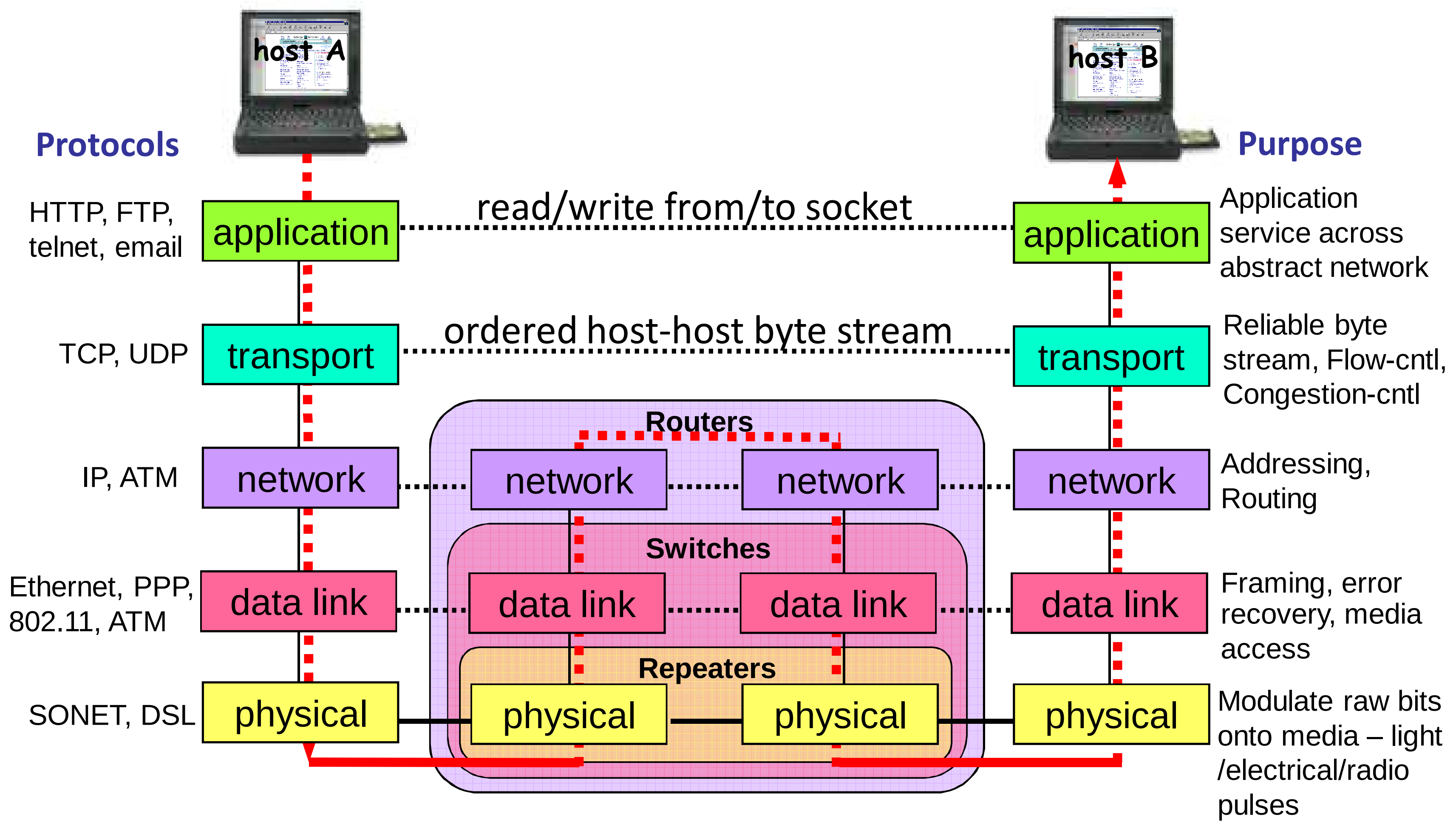
- A collection of markup languages for describing the structure & meaning of document-content together with style descriptors to control document presentation
- A collection of protocols for moving content around the Net
  - especially HTTP (Hyper-Text Transfer Protocol), but also XHR, SSL/HTTPS, SOAP, RTSP, etc.
- Programming languages and frameworks for implementing dynamic content & behavior on client and server (e.g. JavaScript, jQuery, Backbone, XSLT, Java, JSP, PHP, RoR, SQL)
- W3C (World-Wide-Web Consortium, http://www.w3.org) established to promulgate Web technology standards (do standards really matter?)

Web-Internet Relationship

- From the Internet's point of view, the Web's HTTP protocol is just one among many <u>application protocols</u>, along with SMTP (email), FTP, SIP, DNS, etc.
- Internet is designed as a <u>layered architecture</u> whose layers include:

- LAN (Local Area Network) connects hosts, e.g. with Ethernet (link-layer address, e.g.: **`00-B0-D0-3E-51-BC`**)
- IP (Internet Protocol) for interconnecting separate physical networks (LANs) uses IP addresses, e.g.: **`142.150.160.47`**)
- TCP (Transmission Control Protocol) for reliably sending streams of bytes to TCP ports on an host on an IP network
- Applications, such as the Web, running on network-attached hosts, exchange information over the above layers
- DNS (Domain Name Service) is a special application protocol for translating names and domains into IP addresses (e.g. **`www.utsc.utoronto.ca == 142.1.96.30`**)

Internet Protocol stack


host A
host B
Protocols
Purpose
HTTP, FTP, telnet, email
application
read/write from/to socket
application
Application service across abstract network
TCP, UDP
transport
ordered host-host byte stream
transport
Reliable byte stream, Flow-cntl, Congestion-cntl
Routers
IP, ATM
network
network
network
network
Addressing, Routing
Switches
Ethernet, PPP, 802.11, ATM
data link
data link
data link
data link
Framing, error recovery, media access
Repeaters
SONET, DSL
physical
physical
physical
physical
Modulate raw bits onto media – light /electrical/radio pulses

01 World Wide Web CSCC09 10

Internet’s Role in Web Operation

- The Internet is a fascinating system in its own right, but we have time only for a snapshot overview:
  - The Internet Protocol (IP) provides unreliable best-effort packet-delivery service between host interfaces
  - The Transmission Control Protocol (TCP) provides a reliable ordered-byte-stream service on top of IP
  - Hypertext Transfer Protocol (HTTP) uses TCP to exchange messages between Web clients & servers
  - Domain Name System (DNS) uses the unreliable UDP protocol to provide an IP-address lookup service
- The gory details are covered in CSCD58

Internet Engineering & Governance

- Just as the Web has the W3C to manage its standardization process, the Internet has two standardization/governance bodies:
- Internet Engineering Task Force (IETF): as the name suggests this group is responsible for engineering the Internet protocols, including HTTP (Web), TCP (generic Internet reliable connection service), IP (the unreliable Internet packet protocol) and many others
- Internet Corporation for Assigned Names and Numbers (ICANN) has a more politicized role, in managing the distribution of IP address blocks to regions, countries and ISP's, as well as the process of managing the Internet's domain name system

Client-Server Architecture

- Web Browsers (clients)
  - Chrome (the C09 reference browser), Firefox, IE, Safari, Opera, Blackberry, Lynx, ...
- Web Servers
  - Apache, njinx, Tomcat, Microsoft IIS, Node.js ...
- Naming via URLs (Uniform Resource Locators)
  - **`<scheme>:<scheme-specific-part>, e.g.`**
  - **`http://<fully-qualified-domain-name>/<path>, e.g.`**
  - **`http://www.utsc.utoronto.ca/~rosselet/cscc09/`**
- HTTP (HyperText Transfer Protocol) and MIME (Multipurpose Internet Mail Extensions – why need MIME?)
  - **`<type>/<subtype>  e.g.: text/html, text/xml, image/png, audio/mp4, video/H261, application/json ...`**

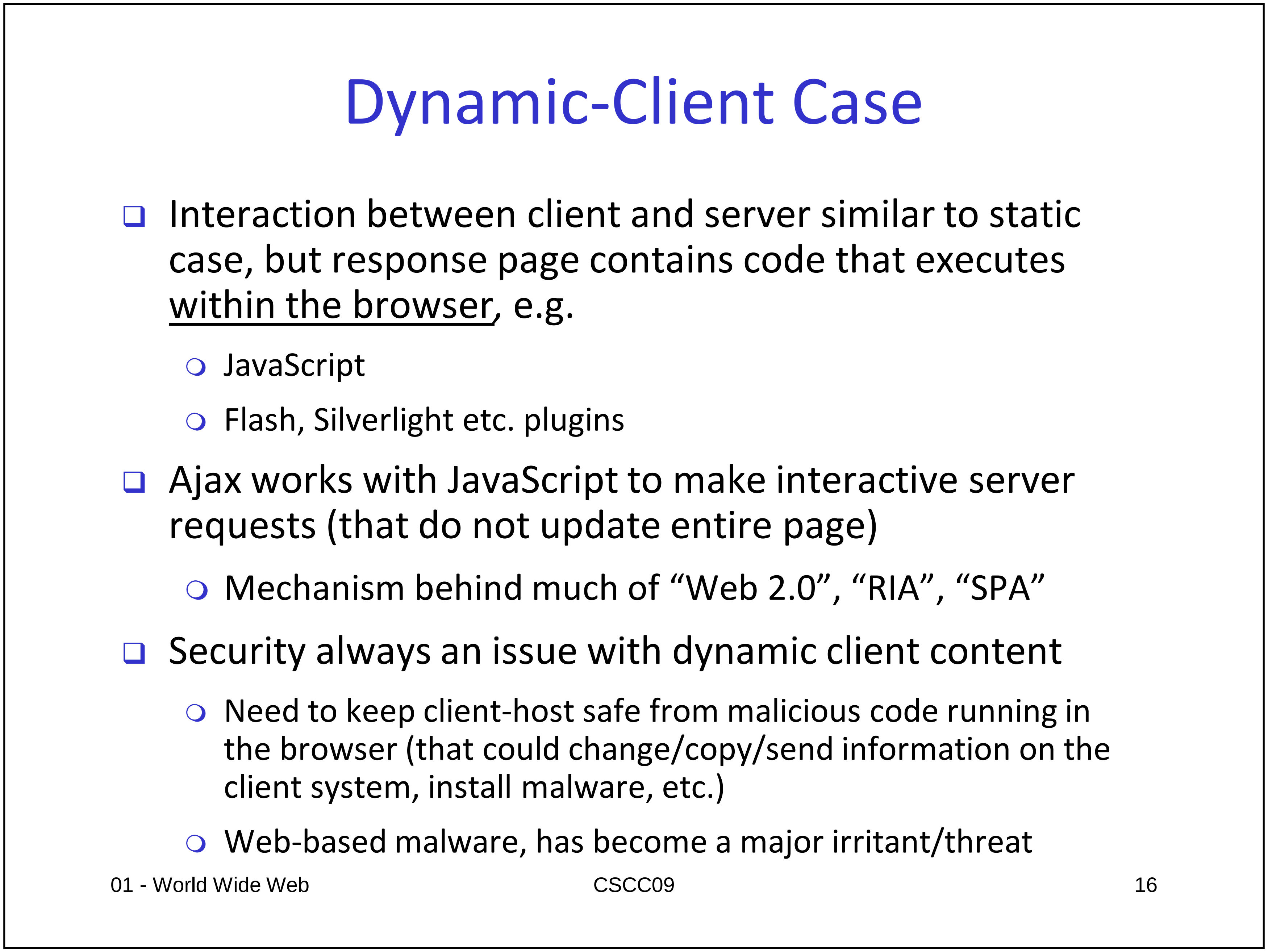
Static-Content Case

- Type a URL into a browser window, e.g.:
  - **`http://www.utsc.utoronto.ca`**
- Since the Internet's plumbing works with numbers not names, DNS is invoked to convert to numeric host IP number: **`142.1.96.30`**
- Browser makes a TCP connection to "port" 80 on the server and sends an HTTP GET request (why 80?)
- Web Server running on **`www.utsc`** fetches the default page for this site and sends it back to the browser via TCP and the HTTP protocol
  - response header provides info to help the browser interpret the page coming from the server: the content type (e.g. MIME text/html), character encoding (e.g. ISO-8859-1), etc.
- Server may elect to forward or redirect the request

```
> telnet www.utsc.utoronto.ca 80
    Trying 142.1.96.30...
    Connected to www.utsc.utoronto.ca.
    Escape character is '^]'.
  GET /. HTTP/1.0 \n\n
    HTTP/1.1 200 OK
    Server: nginx
    Date: Thu, 28 Aug 2014 18:39:38 GMT
    Content-Type: text/html
    Connection: close
    Set-Cookie: Apache=142.1.96.164.1409251178636876; path=/
    Accept-Ranges: bytes
    Cache-Control: no-cache,must-revalidate
    Pragma: no-cache
    <!doctype html>

    <html xmlns="http://www.w3.org/1999/xhtml" lang="en" xmlns:html5="http://www.w3.org/1999/xhtml" >
    <head>
    <meta http-equiv="X-UA-Compatible" content="IE=EmulateIE7, IE=edge, chrome=1" />
    <meta name="language" content="en" />

    <meta charset="utf-8" />

    <meta name="viewport" content="width=device-width; initial-scale=1"/> <!-- very important 4
       mobile devices -->

    <meta name="University of Toronto Scarborough" content="The homepage of the University of Toronto
       Scarborough"/>
    <meta name="description" content="The home page of the University of Toronto Scarborough."/>
    <meta name="keywords" content="University of Toronto at Scarborough"/>
    ...
```

Try it!

(can be useful as a debugging aid, to see unprocessed details of a server's response)



Dynamic-Client Case

- Interaction between client and server similar to static case, but response page contains code that executes within the browser, e.g.
  - JavaScript
  - Flash, Silverlight etc. plugins
- Ajax works with JavaScript to make interactive server requests (that do not update entire page)
  - Mechanism behind much of “Web 2.0”, “RIA”, “SPA”
- Security always an issue with dynamic client content
  - Need to keep client-host safe from malicious code running in the browser (that could change/copy/send information on the client system, install malware, etc.)
  - Web-based malware, has become a major irritant/threat

01 - World Wide Web CSCC09 16

Dynamically-Served Content Case

- Page requests trigger code-execution on server side
  - Usually the output produced by this code is returned to the browser – e.g. code could trigger a search of a server-side persistent data store whose results would be returned in the response
- Widely used technologies:
  - Common Gateway Interface (CGI) can be used with a wide variety of server languages)
  - Server-Side Includes (SSI)
  - JSP, ASP, PHP, RoR, etc. embed server commands w/i HTML docs
  - Java Servlets/JSP execute on Web server
  - RESTful API's – language agnostic, typically used with "thinner" servers
- Security a concern here too, but now the issue shifts to server-side security (e.g. poorly written CGI could open door to malicious actions against server)

Browser-Document Compatibility

- Backward compatible (browser looking backward)
  - A new browser renders an earlier-generation document
    - deprecated features and older syntax must live on in browsers
- Forward compatible (browser looking forward)
  - An older browser renders a later-generation document
    - requires hacks in the document; author includes support for older browsers (or in Google's case, renders old UI for user)
- Sideways compatible
  - Browsers from different companies render documents in a consistent way
    - requires standards (the W3C's role)
    - although standards are widely implemented, still often must use document hacks (e.g. for CSS, JavaScript, DOM, Ajax)

Web Synopsis

- Invented by Tim Berners-Lee, drawing upon a number of existing technologies/systems including: SGML, hypertext, the Internet, Unix, client-server architecture
- TBL designed and implemented HTML, networked hyperlinks , URL's, HTTP
- The Web consists of a collection of markup and style languages, network protocols, programming languages
- Web standards are set by the W3C (and why stds matter)
- Internet serves as Web communication system
- Browsers and servers operate in a client-server relationship:
  - HTTP and MIME protocols
  - Static/dynamic client, server scenarios
- Browser compatibility is an issue for Web apps