Stony Brook University

CSE 361: Web Security

Introduction and History of the Web

Nick Nikiforakis

Who am I, and where can you find me?

- Nick Nikiforakis
  - Associate Professor in Department of Computer Science
  - Research interests:
    - Web security and Privacy
    - DNS security
    - Intrusion detection
  - Office: 361
    - Zoom for this semester
  - Office Hours: Monday/Wednesday (5PM – 6PM)

About the course

- Learn the ins and outs of securing web applications
  - Theory: Lectures, mandatory readings, etc.
  - Practice: Gradually (throughout the semester) secure a vulnerable web application

- Some overlap with CSE 331 (Computer Security Fundamentals)
  - By design, due to CSE 331 being the only security course that many students take
  - New attacks, new defenses, academic papers, hands on assignments, etc.

Logistics

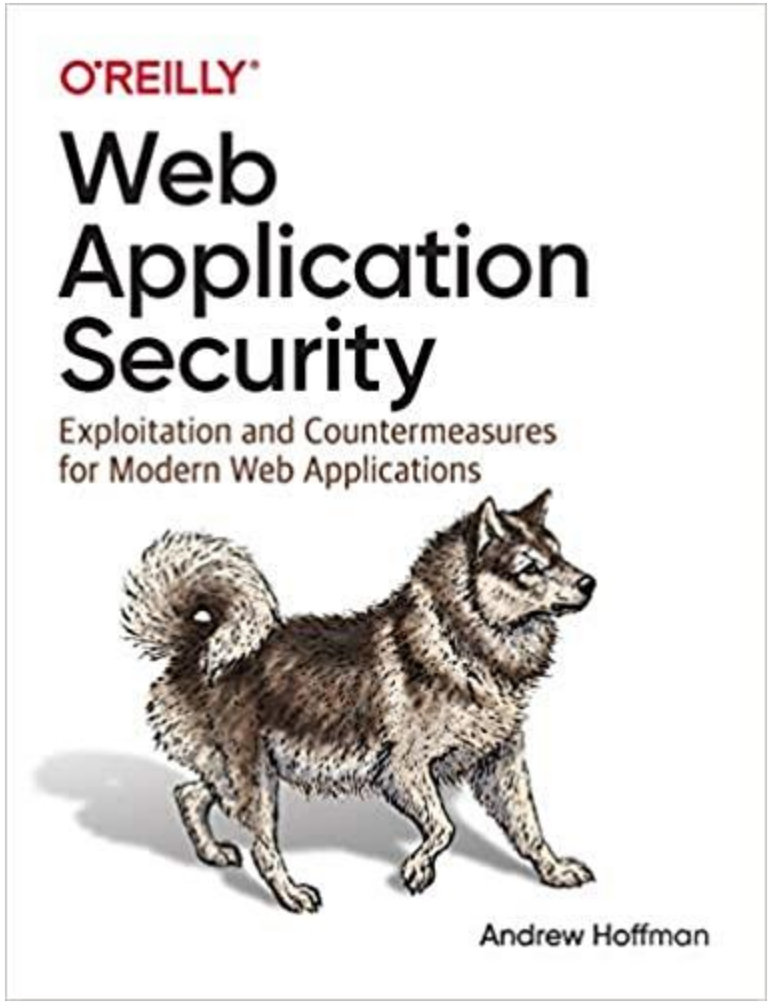
- Class will be on Tuesdays and Thursdays, 4:45 – 6:00 PM
  - Unfortunately, it seems like it will all be over Zoom

- Office hours: Monday, Wednesday, 5pm – 6 pm
  - Dedicated Zoom channel that you'll find on Blackboard

- Grade breakdown
  - Individual assignments (15%)
    - Mostly reading papers, writing summaries, and answering questions
  - Group assignments (25%)
    - Semester-long project-like assignments on securing a web application
  - Midterm (25%)
  - Final (35%)

Logistics

- No official textbook required
  - Slides and mandatory readings should be sufficient
- You should attend lectures
  - Not mandatory but highly encouraged
- Optional book
  - “Web Application Security” book by Andrew Hoffman
  - Currently freely available by NGINX as an ebook
    - Link on the course website


O'REILLY
Web Application Security
Exploitation and Countermeasures for Modern Web Applications
Andrew Hoffman

Late submission policy

- Paper summaries, lab reports, and final project must be delivered by the specified deadlines.
  - Hand them in on time
  - For every day that you are late, there will be a 10% penalty
  - Health-related exemptions will be handled via the appropriate official channels

Code of Conduct

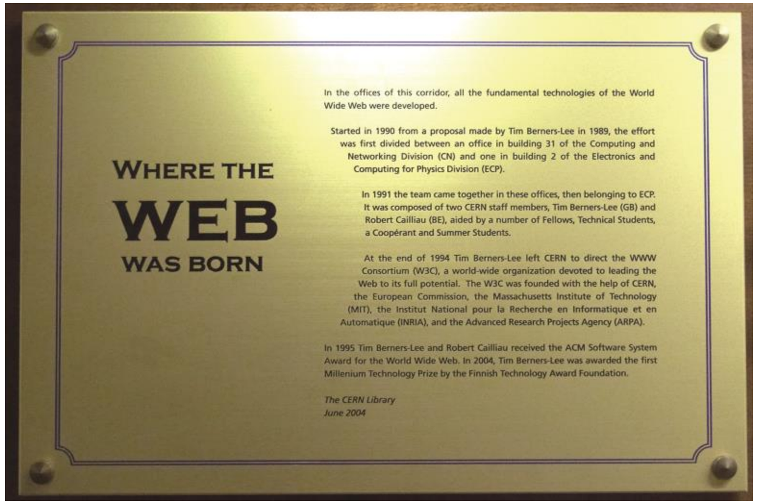
- The work that you present as your own, should be your own
  - Cite the resources that you used (other people's code, documents, etc.)
  - Don't allow your code/paper summaries to be copied
  - Don't copy other people's code or paper summaries
- Anything short of the above, will be grounds for immediate failing of the class and an official report of plagiarism

The Web has won

- Used by billions of people to retrieve information
  - 2B users monthly on Facebook
  - 2.3M searches per second on Google

- Fully-fledged application platform
  - web-based office applications

- Large coverage in mobile applications
  - many mobile apps are just Web views



WHERE THE

WEB

WAS BORN

In the offices of this corridor, all the fundamental technologies of the World Wide Web were developed.

Started in 1990 from a proposal made by Tim Berners-Lee in 1989, the effort was first divided between an office in building 31 of the Computing and Networking Division (CN) and one in building 2 of the Electronics and Computing for Physics Division (ECP).

In 1991 the team came together in these offices, then belonging to ECP. It was composed of two CERN staff members, Tim Berners-Lee (GB) and Robert Cailliau (BE), aided by a number of Fellows, Technical Students, a Coopérant and Summer Students.

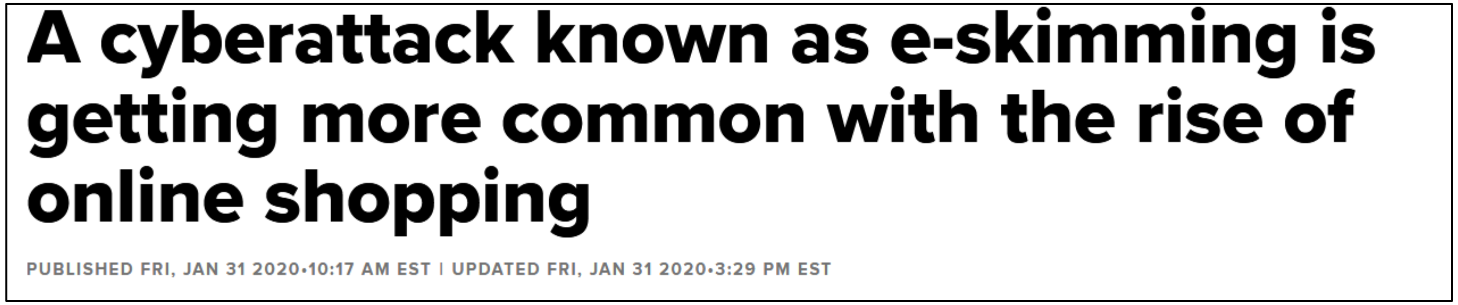
At the end of 1994 Tim Berners-Lee left CERN to direct the WWW Consortium (W3C), a world-wide organization devoted to leading the Web to its full potential. The W3C was founded with the help of CERN, the European Commission, the Massachusetts Institute of Technology (MIT), the Institut National pour la Recherche en Informatique et en Automatique (INRIA), and the Advanced Research Projects Agency (ARPA).

In 1995 Tim Berners-Lee and Robert Cailliau received the ACM Software System Award for the World Wide Web. In 2004, Tim Berners-Lee was awarded the first Millenium Technology Prize by the Finnish Technology Award Foundation.

The CERN Library
June 2004

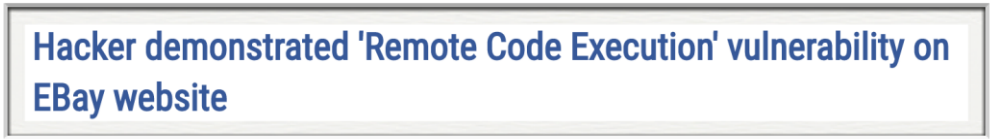
... and the hackers with it

YAHOO MAIL XSS BUG WORTH ANOTHER $10K TO RESEARCHER

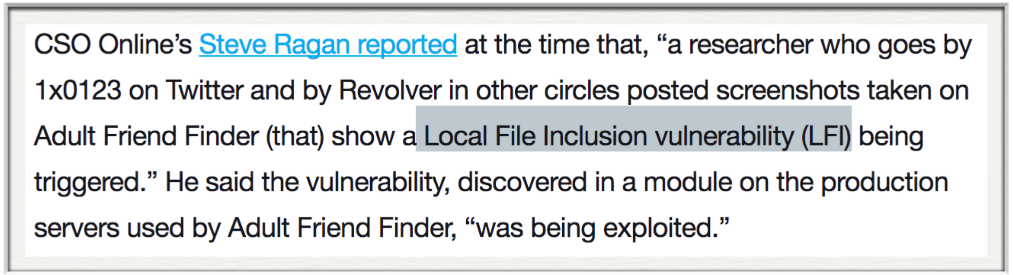


A cyberattack known as e-skimming is getting more common with the rise of online shopping

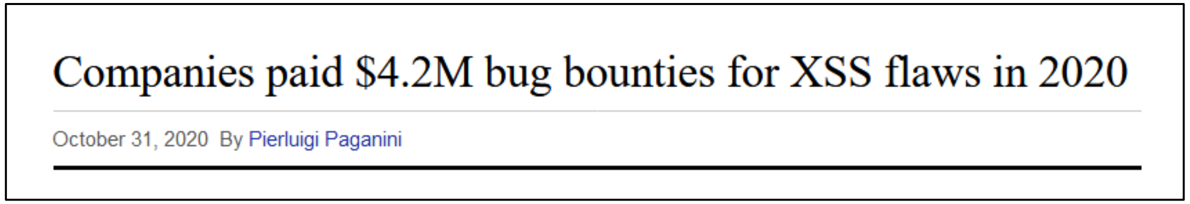
PUBLISHED FRI, JAN 31 2020·10:17 AM EST | UPDATED FRI, JAN 31 2020·3:29 PM EST



Hacker demonstrated 'Remote Code Execution' vulnerability on EBay website



CSO Online's Steve Ragan reported at the time that, "a researcher who goes by 1x0123 on Twitter and by Revolver in other circles posted screenshots taken on Adult Friend Finder (that) show a Local File Inclusion vulnerability (LFI) being triggered." He said the vulnerability, discovered in a module on the production servers used by Adult Friend Finder, "was being exploited."



Companies paid $4.2M bug bounties for XSS flaws in 2020

October 31, 2020 By Pierluigi Paganini

Why Web Security?

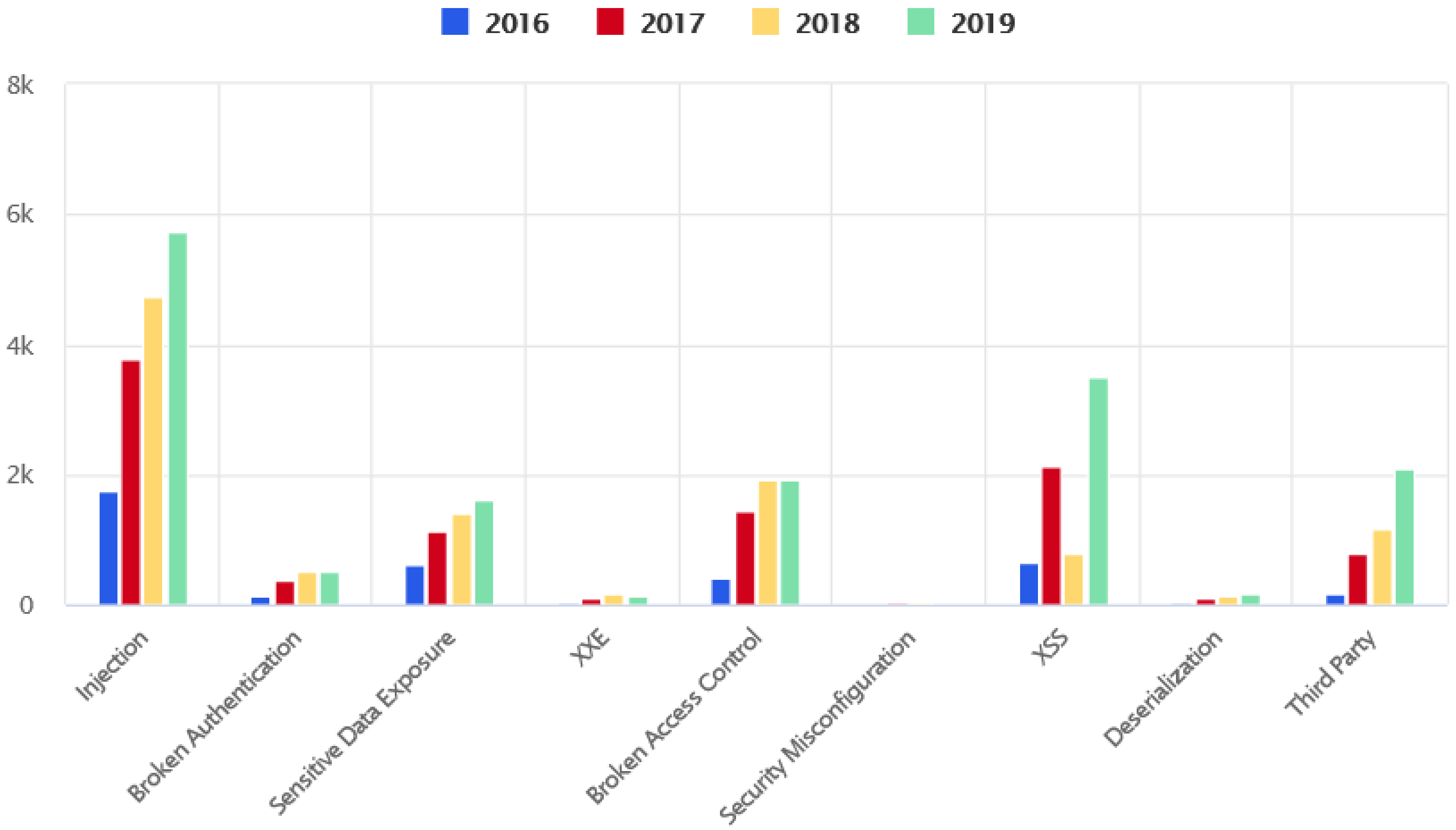
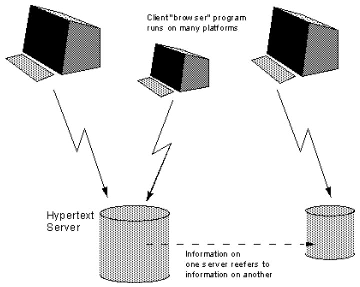


Image source: https://www.imperva.com/blog/the-state-of-vulnerabilities-in-2019/

Short History of the Web

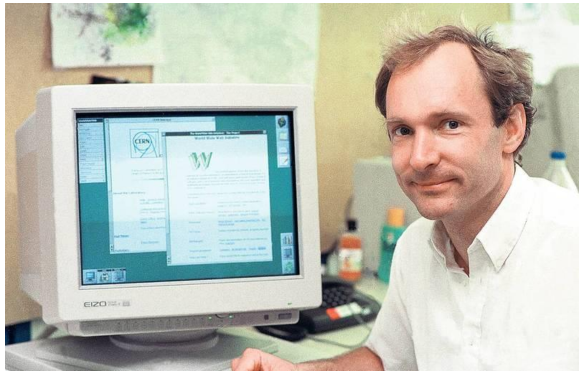
From Hypertext to the World Wide Web

- Hypertext concept first mentioned in 1945
  - Theoretical, Memex (Memory Extender) system by Vannevar Bush
  - No "linear text" anymore, links between documents
- In 1980, Tim Berners-Lee developed ENQUIRE
  - local links between documents only
- In 1989, Berners-Lee wrote "Information Management: A Proposal"
  - extends Hypertext to multiple servers and links between them
  - Basis for the modern Web

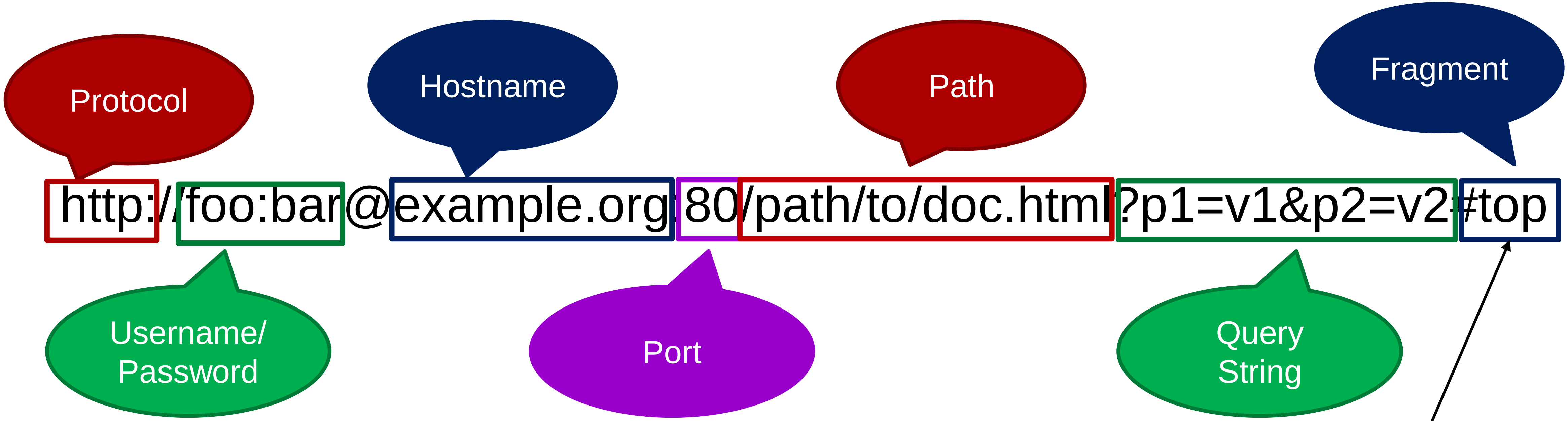


HTTP and HTML

- Web as envisioned by T. Berners-Lee
  - document-centric
  - stateless (just documents linking to one another)
  - structured (based on SGML)
  - tags for semantic interpretation
- HTTP 0.9 introduced in 1991
  - required to answer with an HTML page
  - no headers either way (introduced in 1992 though)
- HTML initially supported 18 tags
  - 11 made it into HTML4 and later versions



Uniform Resource Locator (URL)


Protocol
Hostname
Path
Fragment
http://foo:bar@example.org:80/path/to/doc.html?p1=v1&p2=v2#top
Username/ Password
Port
Query String

Fragments are not sent to the server

HTTP Evolution over Time: HTTP 0.9

- Requirements
  - as simple as possible
  - serve **single** HTML pages

- Result
  - only GET requests
  - no client or server headers
  - server directly answers with HTML body

```
GET /path/to/doc.html
```

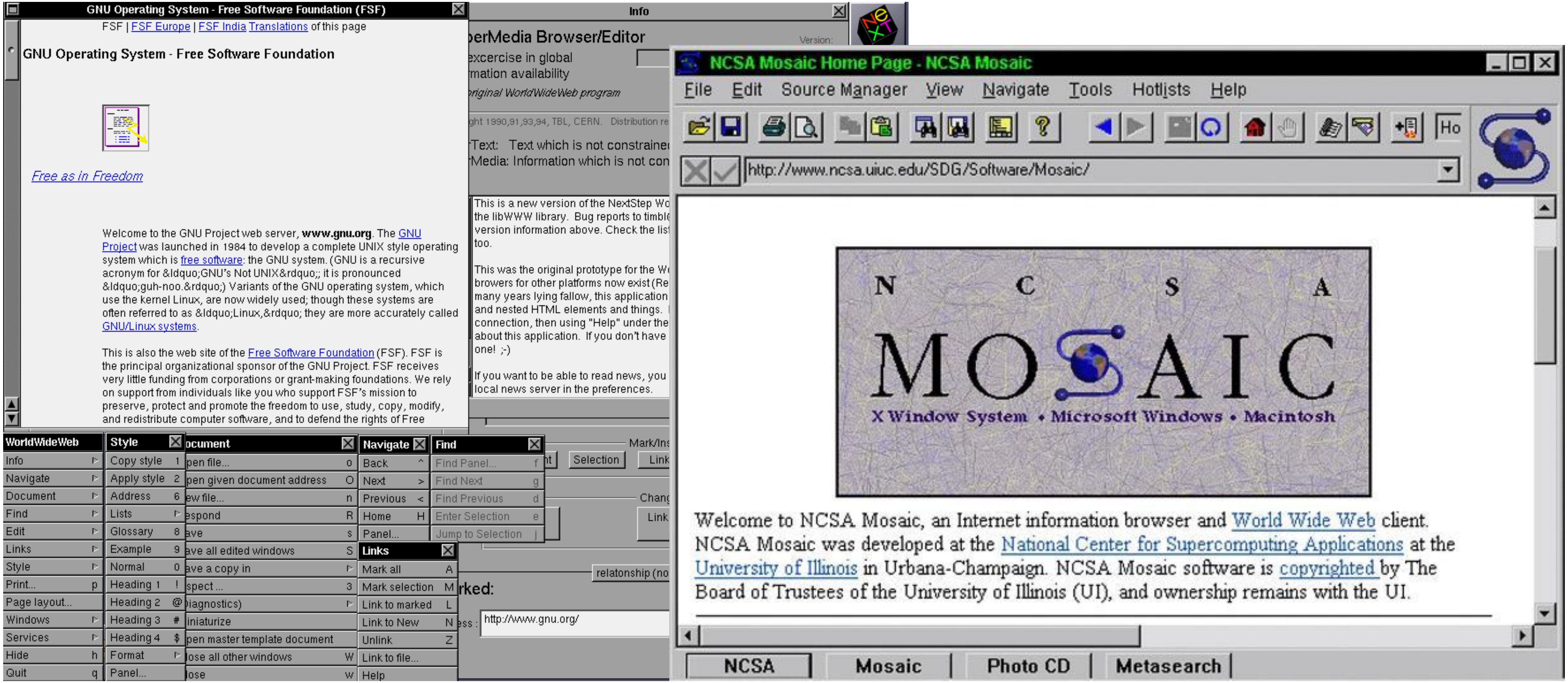
```
<html>…

(connection closed)
```

The first "real" browser: Mosaic (1993)

- Initial version of HTML could only link to images
  - allowed to be on remote server
- Mosaic introduced the `<img>` tag for inline images
  - Implemented by Marc Andreessen
- Images could reside on remote server
  - Birth of the multi-origin Web
  - Followed by many HTML tags later (embed, object, style, script, …)

We’ve come a long way



GNU Operating System - Free Software Foundation (FSF)

FSF | FSF Europe | FSF India Translations of this page

GNU Operating System - Free Software Foundation

Free as in Freedom

Welcome to the GNU Project web server, **www.gnu.org**. The GNU Project was launched in 1984 to develop a complete UNIX style operating system which is free software: the GNU system. (GNU is a recursive acronym for &ldquo;GNU’s Not UNIX&rdquo;; it is pronounced &ldquo;guh-noo.&rdquo;) Variants of the GNU operating system, which use the kernel Linux, are now widely used; though these systems are often referred to as &ldquo;Linux,&rdquo; they are more accurately called GNU/Linux systems.

This is also the web site of the Free Software Foundation (FSF). FSF is the principal organizational sponsor of the GNU Project. FSF receives very little funding from corporations or grant-making foundations. We rely on support from individuals like you who support FSF’s mission to preserve, protect and promote the freedom to use, study, copy, modify, and redistribute computer software, and to defend the rights of Free

Info

perMedia Browser/Editor

Version:

excercise in global
rmation availability

original WorldWideWeb program

ght 1990,91,93,94, TBL, CERN. Distribution re

Text: Text which is not constrained
Media: Information which is not con

This is a new version of the NextStep Wo
the libWWW library. Bug reports to timbl
version information above. Check the list
too.

This was the original prototype for the W
browsers for other platforms now exist (Re
many years lying fallow, this application
and nested HTML elements and things.
connection, then using "Help" under the
about this application. If you don't have
one! ;-)

If you want to be able to read news, you
local news server in the preferences.

rked:

http://www.gnu.org/

NCSA Mosaic Home Page - NCSA Mosaic

http://www.ncsa.uiuc.edu/SDG/Software/Mosaic/

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.

HTTP Evolution over Time: HTTP 1.0 (1991-1995)

- Requirements
  - serve content other than plain text documents
  - allow for authentication
  - allow for transmission of meta information, e.g., age of file
  - transmit data to the server (via forms)
- Result
  - Mandatory HTTP version in request
  - Optional headers in request and response
  - Status Line in response
  - New methods: POST and HEAD

```
GET / HTTP/1.0
Host: example.org
```

```
HTTP/1.0 200 OK
Content-Length: 123

<html>…
(connection closed)
```

HTTP Requests (since HTTP/1.0)

- Consists of several, partially optional components
- Request Line with *Verb*, Path, and Protocol
- List of HTTP headers, as header:value
- Empty line to end headers
- Optional body message (used, e.g., with POST requests)

```
GET /index.html HTTP/1.0
Host: stonybrook.edu
Cookie: hello=1
```

HTTP GET request

- Purpose: retrieve resource from server
- Should not cause side effects on Web server’s state
  - dubbed "idempotent" in W3C standard
  - although it does often cause side effects in practice, due to developers
- Should not carry a message body
- Parameters passed via URL
  - Special characters percent-encoded (hex value of char, e.g., ? = %3F)
  - **Usually logged on server side together with requested file**

```
GET /index.html?name=value%3F HTTP/1.0
Host: stonybrook.edu
```

HTTP POST request

- Purpose: send data to the server
  - for storage or processing
  - should be used for state-changing operations
- Can be combined with GET parameters
- Message body contains data
  - Depending on content-type, percent-encoded or plain

```
POST /index.html?name=value%3F HTTP/1.0
Host: stonybrook.edu
Content-Length: 10
Content-Type: application/json

{"a": "?"}
```

```
POST /index.html?name=value%3F HTTP/1.0
Host: stonybrook.edu
Content-Length: 5
Content-Type: application/x-www-form-urlencoded

a=%3F
```

HTTP Response (since HTTP/1.0)

- Status Line: Protocol, Status Code, and *Status Text*
- List of HTTP headers, as header:value
- Empty line to end headers
- Response Body

```
HTTP/1.0 200 OK
Server: nginx
Content-Type: text/html
Content-Length: 123

<html>…</html>
```

HTTP Response Codes

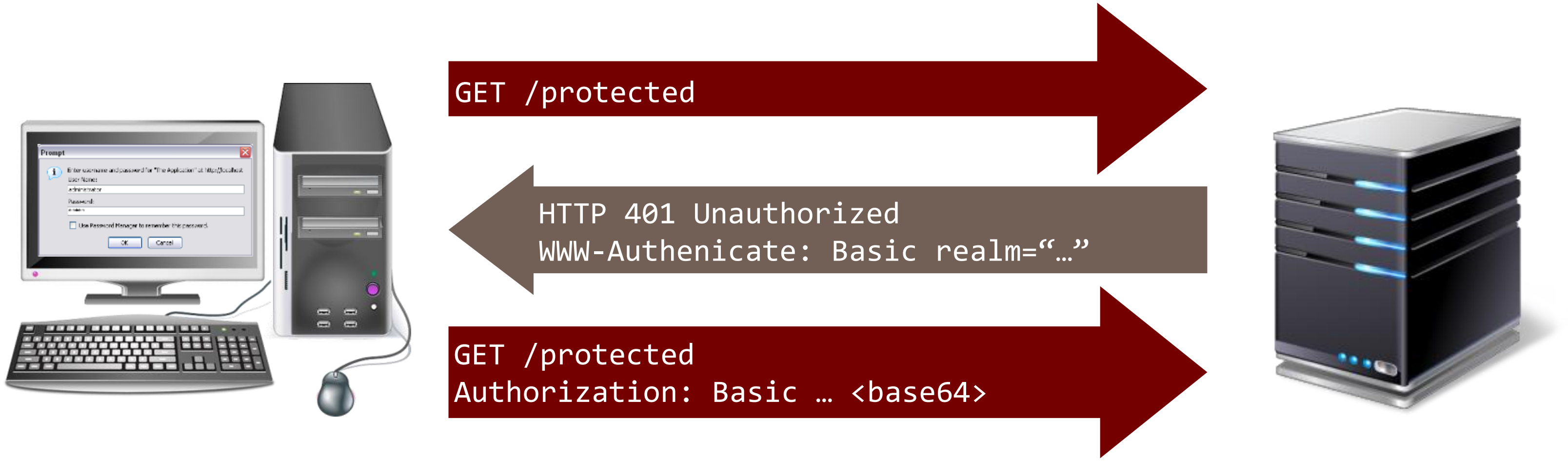
- 2xx Success
  - 200 OK
  - 206 Partial Content (for range requests)
- 3xx Redirection
  - 301 Moved Permanently (always redirect to new URL)
  - 302 Found (redirect once, don’t store redirect)
  - 304 Not Modified (not changed since last client request, not transferred)
  - 307 Moved Temporarily (only redirect to new URL this time)

HTTP Response Codes

- 4xx Client errors
  - 400 Bad Request (e.g., no carriage return in HTTP request)
  - 401 Unauthorized (used for HTTP authentication)
  - 403 Forbidden
  - 404 Not Found
  - 405 Method Not Allowed
  - 418 I’m a teapot (April Fool’s Joke, see RFC 2324)
- 5xx Server errors
  - 500 Internal Server Error
  - 502 Bad Gateway (e.g., timeout in reverse proxies)

First Security Considerations: HTTP Authentication (1993)

- Need for authentication/authorization was recognized early on
- HTTP remained stateless
  - Authentication via HTTP header
- Not too useful for session management though



```
GET /protected
```

```
HTTP 401 Unauthorized
WWW-Authenicate: Basic realm="…"
```

```
GET /protected
Authorization: Basic … <base64>
```

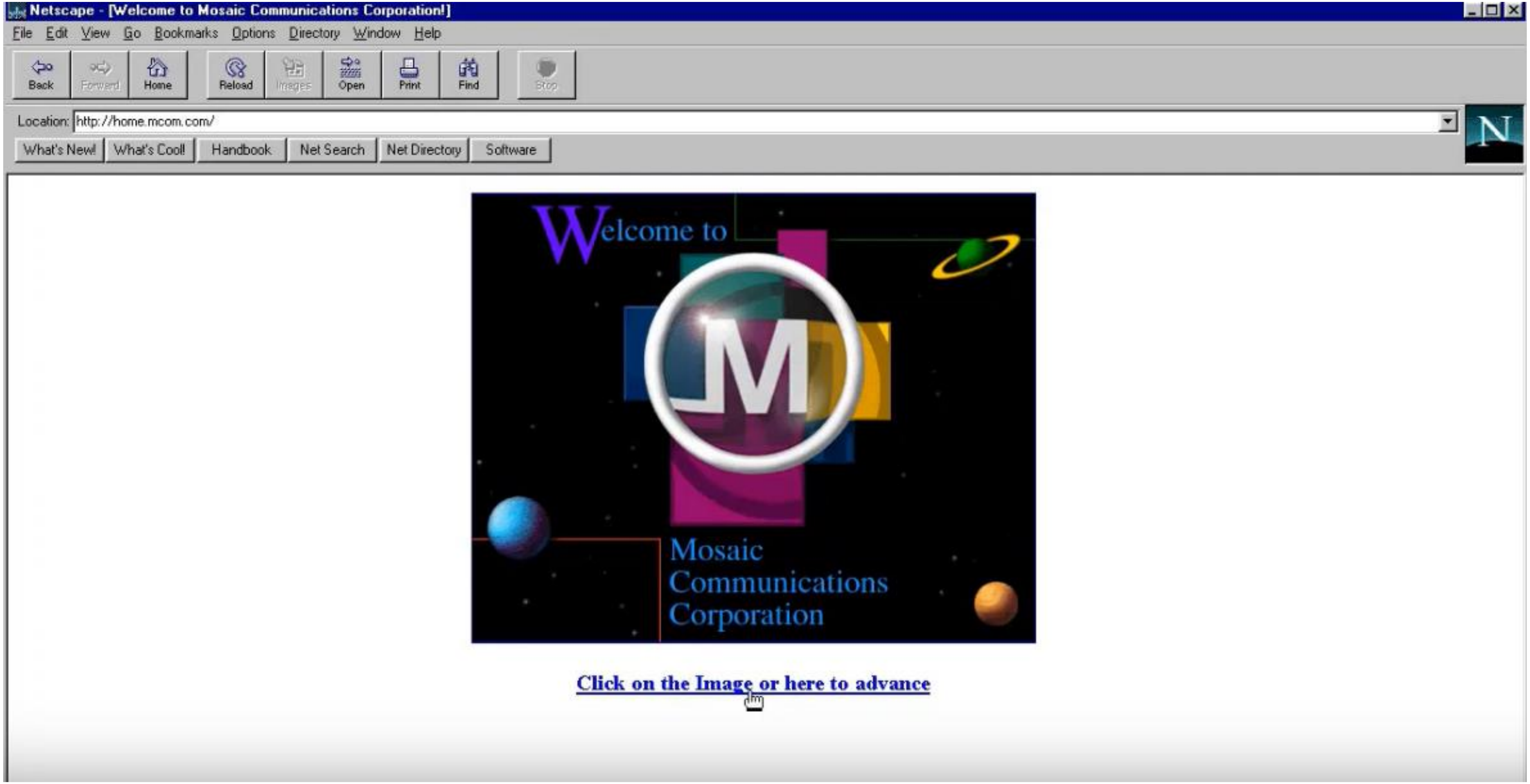
Cookies (1994)

- Adding state to the stateless Web
  - Required to develop applications which should re-identify a user

- Initially added in Netscape Navigator
  - set via HTTP response header
  - sent along with every subsequent request
  - … until lifetime is exceeded, or cookie is deleted

JavaScript (1995)

- Netscape wanted a "glue language" added to HTML
- Brendan Eich was hired by Netscape to "implement Scheme in the browser"
- Instead, he was tasked with developing *Mocha* (initially dubbed *LiveScript)*
  - in the first beta release, already renamed to JavaScript
  - Java was very popular back then
- JavaScript later specified as ECMAScript (ECMA-262)

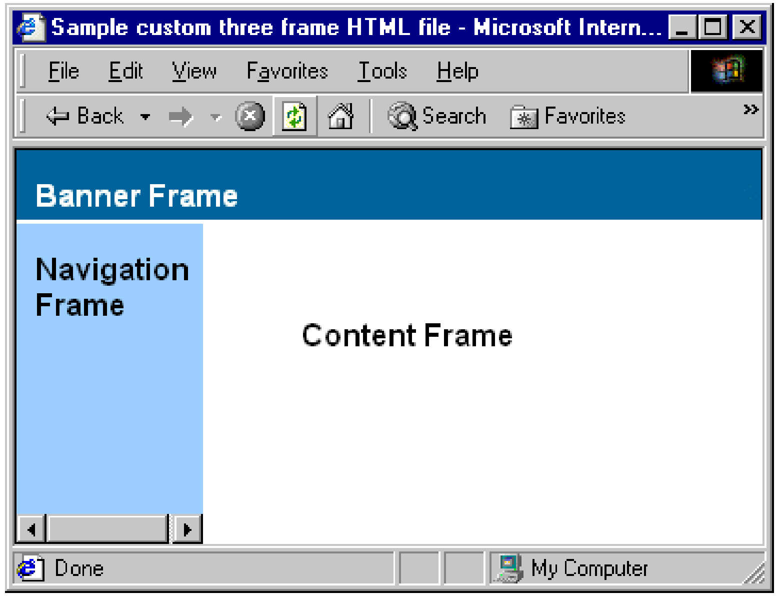


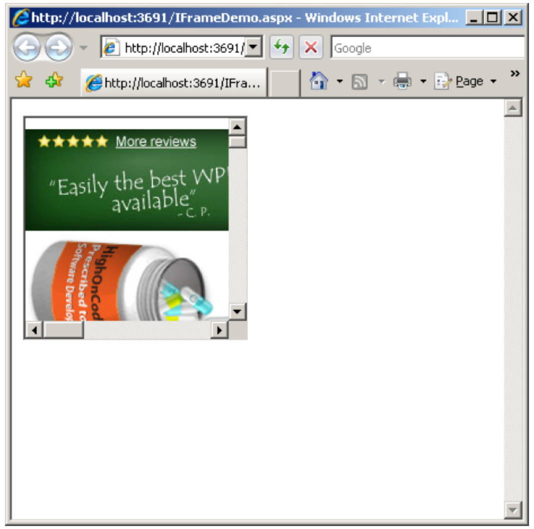
Netscape - [Welcome to Mosaic Communications Corporation!]
File Edit View Go Bookmarks Options Directory Window Help
Back
Forward
Home
Reload
Images
Open
Print
Find
Stop
Location: http://home.mcom.com/
What's New!
What's Cool!
Handbook
Net Search
Net Directory
Software
Welcome to
M
Mosaic Communications Corporation

Click on the Image or here to advance

Frames (1995) and Iframes (1997)

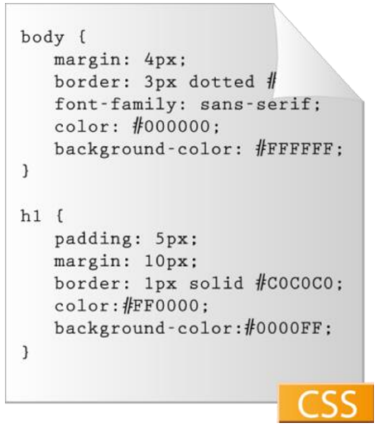
- Concept of frames to display more than one HTML page in a window
  - reduce bandwidth by splitting page
  - fixed navigation elements
- Frames are permitted to come from different origins
- Each frame behaves like a browser window
  - Content rendered and interpreted as if page is loaded regularly
- Reason for introducing the Same-Origin Policy
  - separates frames if they don't share an origin
  - only introduced after first cases of abuse…




http://localhost:3691/IFrameDemo.aspx - Windows Internet Expl…
http://localhost:3691/
Google
http://localhost:3691/IFra…
Page
More reviews
"Easily the best WP available" - C. P.
HighOnCod
Prescribed to
Software Develop

Cascading Style Sheets (1996)

- HTML was initially designed to reflect structure of a document
  - Title, Headings, Sections, Listings, Lists, …
- Web became more popular, should look nicer
  - design tags were add, such as `font`, `b`, `i`
- CSS added to separate *structure* and *presentation*
  - Declarative syntax
  - Could be included remotely or added inline
- Capable of e.g., background images, element placing, opacity

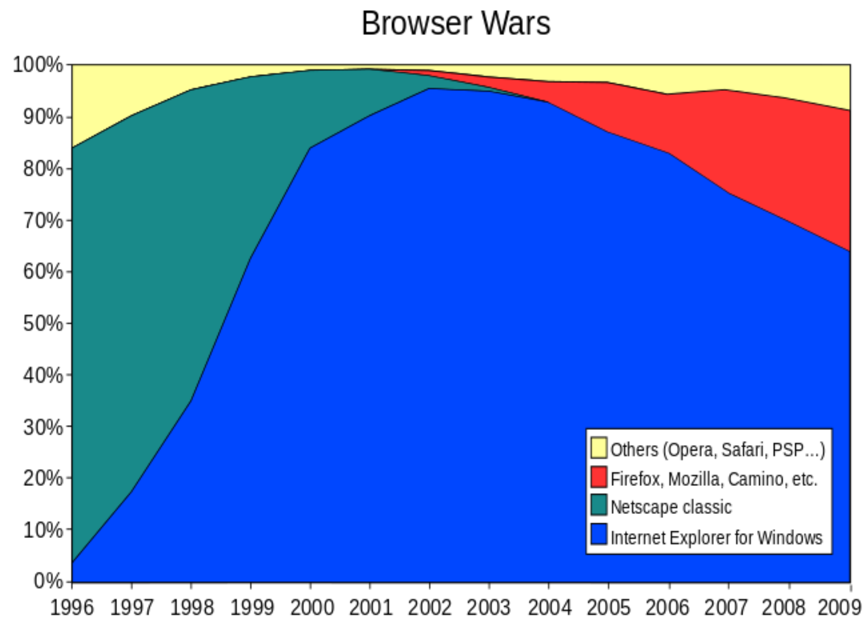


```
body {
    margin: 4px;
    border: 3px dotted #
    font-family: sans-serif;
    color: #000000;
    background-color: #FFFFFF;
}

h1 {
    padding: 5px;
    margin: 10px;
    border: 1px solid #C0C0C0;
    color:#FF0000;
    background-color:#0000FF;
}
```

The First Browser War (1996-1999)

- Market share battle between Netscape and Internet Explorer
- Goal: work with as many sites as possible to win the battle (**compatibility**)
  - everybody was "programming" bad HTML
  - resulted in highly relaxed parsing process
  - error-tolerant to a fault…
- Microsoft’s Internet Explorer won by a landslide
  - also caused by Microsoft’s OS dominance



By Wereon - Image:Browser Wars.png, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=1128061

HTTP Evolution over Time: HTTP 1.1 (finalized 1999)

- Requirements
  - Increased resource size requires other transport and caching strategies
  - Fix some ambiguities in the previous protocol versions
  - Assess server’s capabilities to handle requests
- Result
  - New methods: PUT (similar to POST), DELETE, TRACE, CONNECT (proxies), OPTIONS
  - Keep-Alive connections
  - Accept-Encoding info for the server
  - Chunked transfers, range transfers
  - Standardized in RFC 2616

```
GET / HTTP/1.1
Host: example.org
```

```
HTTP/1.0 200 OK
Transfer-Encoding: chunked

7b
<html>…
0
(connection closed)
```

HTTP Evolution over Time: HTTPS (RFC 2818 finalized 1999)

- Initial discussions about S-HTTP (RFC 2660)
  - unencrypted header, only page data and POST bodies encrypted
- Instead: HTTP over TLS/HTTP over SSL/HTTP Secure (**HTTPS**)
  - encapsulates plain HTTP into TLS tunnel
- Server certificate can be verified via chain of trust
  - Trusted root CAs known to browser
- Until 2011, only one hostname per IP
  - Nowadays, Server Name Indication (SNI) allows multiple vhosts via TLS

https://www.welivesecurity.com/wp-content/uploads/2016/09/HTTPS-TLS-SSL-Sicherheits.png

Years of Stagnation (2000-2003)

- Netscape gave up fighting Internet Explorer
  - Microsoft reduced investment into new client-side technologies
  - IE4: September 1997
  - IE5: March 1999
  - IE6: August 2001
  - IE7: October **2006**
- New features only added through plugins and browser admins
  - Flash: audio, video, vector graphics, cross-domain requests
  - Google Gears: client-side persistence, drag&drop, offline support

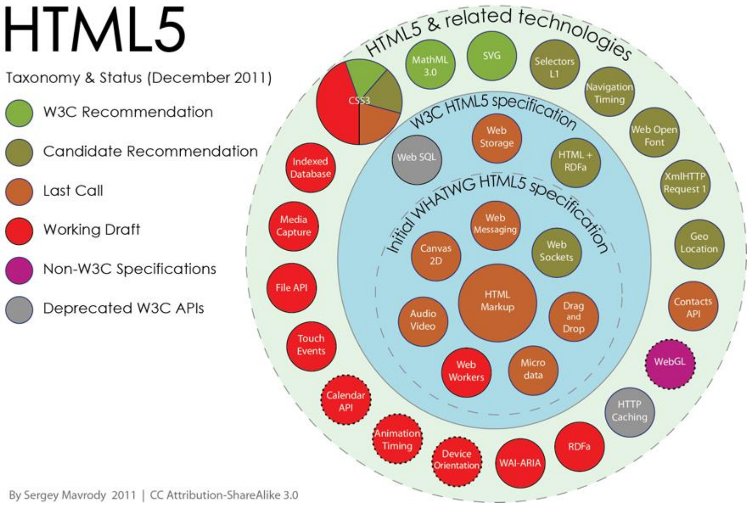
Version: 6.0.2900.2180.xpsp.050329-1536
Cipher Strength: 128-bit
Product ID:
Update Versions:; SP2;

Based on NCSA Mosaic. NCSA Mosaic(TM); was developed at the National Center for Supercomputing Applications at the University of Illinois at Urbana-Champaign.

Copyright ©1995-2004 Microsoft Corp.

HTML5 and the WHATWG (2004)

- New browsers introduced to market
  - Apple Safari (2003) and Mozilla Firefox (2004)
- Introduction of the Web Hypertext Application Technology Working Group (WHATWG)
  - Members from Apple, Mozilla and Opera
  - Concerned with "the W3C’s direction with XHTML, lack of interest in HTML and apparent disregard for the needs of real-world authors"
- Lead to a number of innovations in the browser
  - Still going on today
  - Final specification of HTML5 by November 2014



By Sergey Mavrody 2011 | CC Attribution-ShareAlike 3.0

HTML5 - Highlights

- Audio and Video tags
  - previously only possible with, e.g., Flash
- Web Storage
  - Easy key/value store on the client
  - Can "only" store strings (objects via serialization)
  - Session and (persistent) Local Storage
- Web Messaging
  - postMessages (we'll cover this soon)
- Web Sockets
  - duplex communication channels with the server

HTML5 - Highlights

- Offline Cache
  - controllable caching behavior enables offline apps
- Web Workers
  - allow developers to have tasks run in background
- Geo Location
  - handy feature when displaying maps or local info
- IndexedDB
  - Mixture of SQL and Web Storage
- New (semantic) HTML tags
  - `nav, menuitem, main, footer, header, ...`

The Web 2.0 (2004 - 2005)

- With new functionality in browser came more powerful Web applications
  - Widespread adoption of Flash
  - XMLHttpRequest (already implemented before under different name in IE) API allowed for "Asynchronous JavaScript and XML" (AJAX)
  - Dynamic mash-up web pages
- Popular application examples include:
  - Google Mail
  - Flickr
  - Facebook

BASECAMP
BACKPACK
CHALK.IT
REMEMBERTHEMILK
WRITEBOARD
TADALIST
GOOGLE MAPS
KIKO
WRITELY
GTD
GMAIL
GOOGLE
RSSREADER
GOOGLE IG
UNWIELDY
TAGTAGGER
ZIMBRA
MEEBO
TAGGLING
CLOUDALICIOUS
WIKIPEDIA
Focus on Simplicity
PROTOPAGE
TAGCLOUD
Wikis
ZOOZIO
FLOCK
DELICIOUS
COLLABORATION
WIDGETS
Joy of Use
NETVIBES
STUMBLEUPON
FLICKR
Folksonomy
SHARING
LIVE.COM
AJAX
43THINGS
DIGG
SQUIDOO
PAGERANK
Usability
FLICKR
NING
EBAY
Recommendation
USER CENTERED
BLINKSALE
FOTOLIA
ETSY
SOFLOW
LINKEDIN
FOAF
PERPETUAL BETA
OPENBC
ORKUT
Social Software
SIXDEGREES
PAYPAL
The Long Tail
EBAY
UPCOMING
FRIENDSTER
XFN
Participation
TRUST
Economy
AMAZON
MTURK
PLANZO
BLOGGER
Blogs
AGGREGATION
VC
Affiliation
ITUNES
TYPEPAD
TECHNORATI
WORDPRESS
MINT
Web2.0
beta
COSTPERCLICK
ADSENSE
LASTFM
READ.IO
MEASUREMAP
IPODDER
PODCASTING
IM
LIVELINESS
SIMPLEBITS
PANDORA
Audio
ODEO
SKYPE GTALK JABBER
Convergence
DATA INSIDE
SEO
Design
ITUNES
BIT TORRENT
GRANULARITY
MODULARITY
STYLEGALA
VIMEO
Video
VIDEOCASTING
SIMPLICITY
9RULES
YOUTUBE
WIFI
SYNDICATION
CASTPOST
Mobility
UMTS
Remixability
SOAP
REST
RUBYONRAILS
CSSBEAUTY
GOOGLE MAPS
XML
CREATIVECOMMONS
A9
PLAZES
Standardization
ZENGARDEN
YAHOO MAPS
RSS
OpenAPIs
TAGCLOUD
SEMANTIC
CSS-Design
NEWSGATOR
BLOGLINES
AMAZON
DataDriven
XHTML
ACCESSIBILITY
WASP
FEEDSTER
ROJO
GOOGLE MAPS
Web Standards
ALISTAPART
FEEDBURNER
GMAPS MASHUPS
TECHNORATI
MYYAHOO
GOOGLE NEWS
Microformats
FIREFOX
GOOGLE
A9
ICEROCKET
BUZZTRACKER
INORDER
WINK
ALEXA
MICROFORMATS.ORG
ROLLYO
TRENDMAPPER
PUBSUB

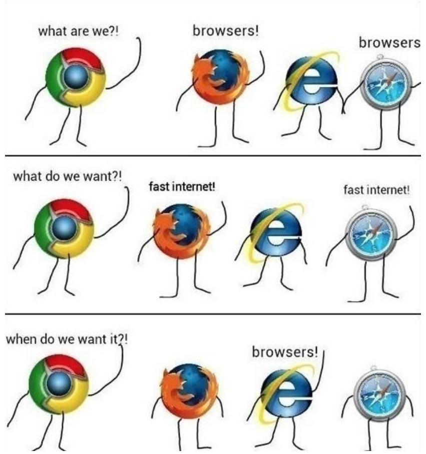
RELEASED UNDER CC 2.0 DE ATTRIBUTION SHARE ALIKE 11.11.05

MARKUS.ANGERMEIER @ APERTO.DE

The Second Browser War (since 2005)

- Four major browser: Internet Explorer, Chrome, Firefox, Safari
- Lively competition (more or less)
- Web standards have been around long enough to focus more on **speed** and not **compatibility**
  - WebKit-based browsers (Chrome, Safari) are fastest nowadays
  - Very active development especially of JavaScript engines
- Both compatibility and speed may be roadblocks for security

- Browser wars 1996-2019: https://www.visualcapitalist.com/internet-browser-market-share/

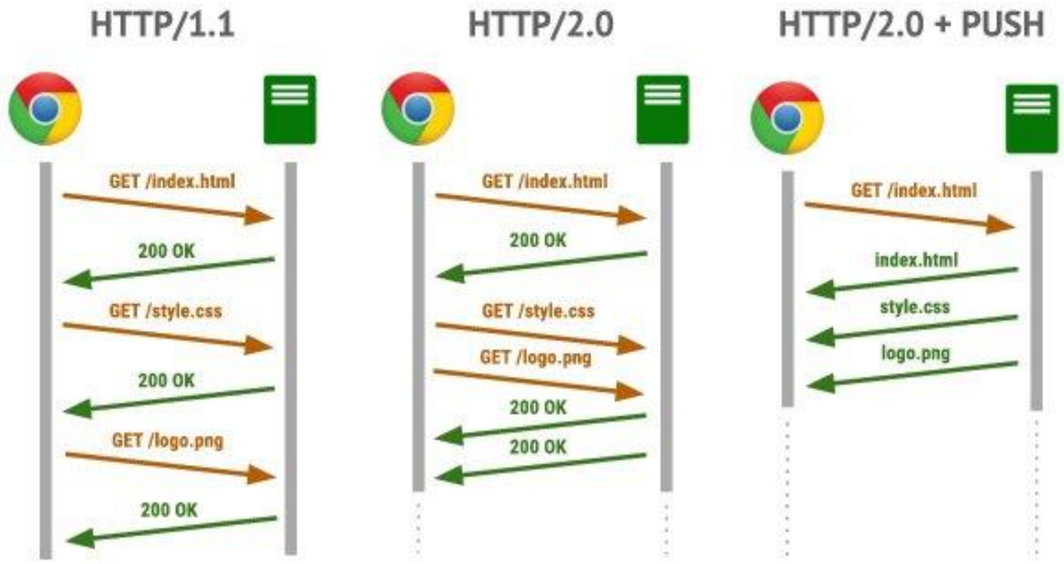


http://www.worthofweb.com/wp-content/uploads/2013/11/internet-browsers.png

http://media02.hongkiat.com/battle-of-browsers-artworks/browsers-battle.jpg

HTTP Evolution over Time: HTTP 2.0 (finalized 2015)

- Requirements
  - Reduce overhead of uncompressed HTTP headers
  - Ensure faster delivery of required resources to client
  - Fix head-of-line blocking from HTTP/1.x
- Result
  - Binary protocol
  - HPACK header compression
  - Server push

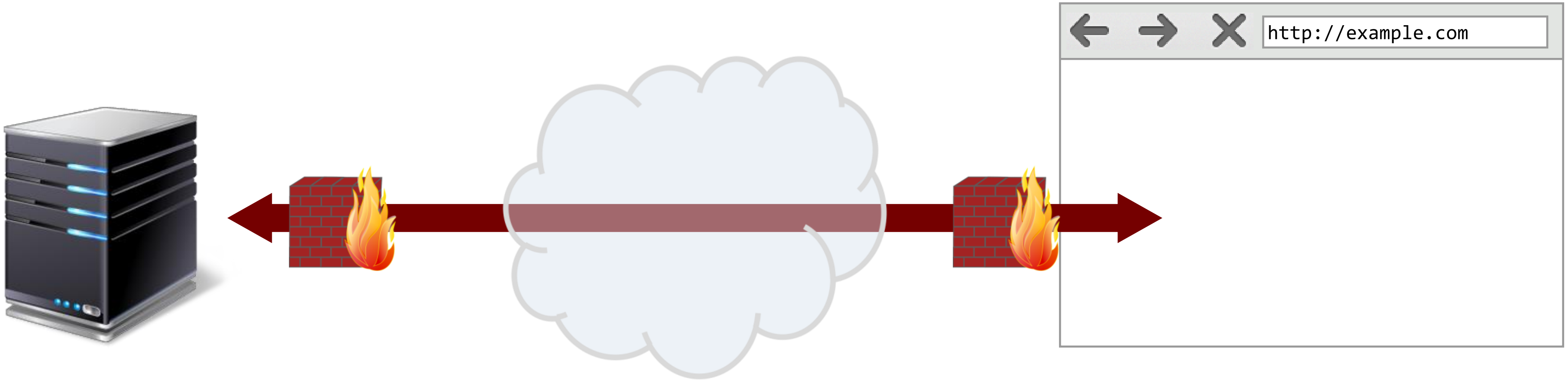


https://giuseppeciotta.net/static/images/h2mosaic_push.jpg

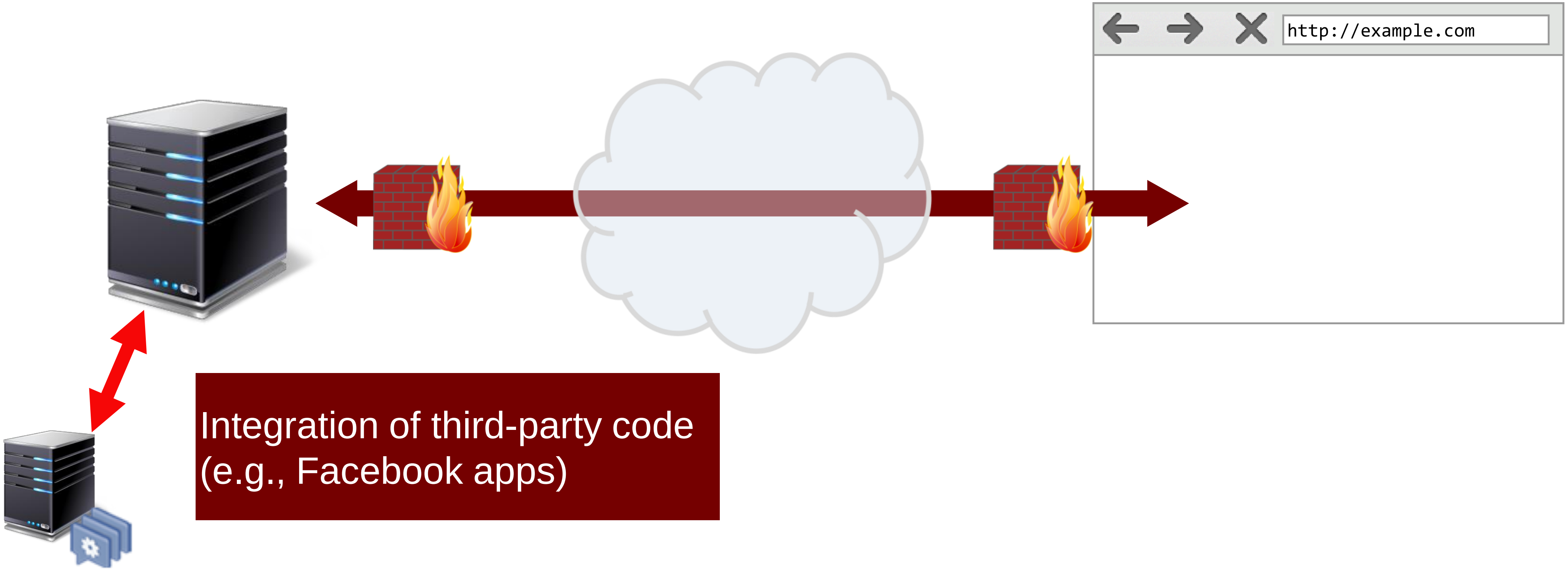
Summary (so far)

- Web was designed to link plain text documents
- Nowadays, we have an application model
  - that is based on multi-origin documents,
  - implements origin-based security models (albeit inconsistently),
  - builds its UI based on at least three languages (HTML, CSS, and JavaScript),
  - and often uses non-security mechanisms (e.g., cookies) for security purposes (e.g., authentication),
  - and supports offline applications with client-side persistence.
- What could possibly go wrong?

Basic Web Paradigm

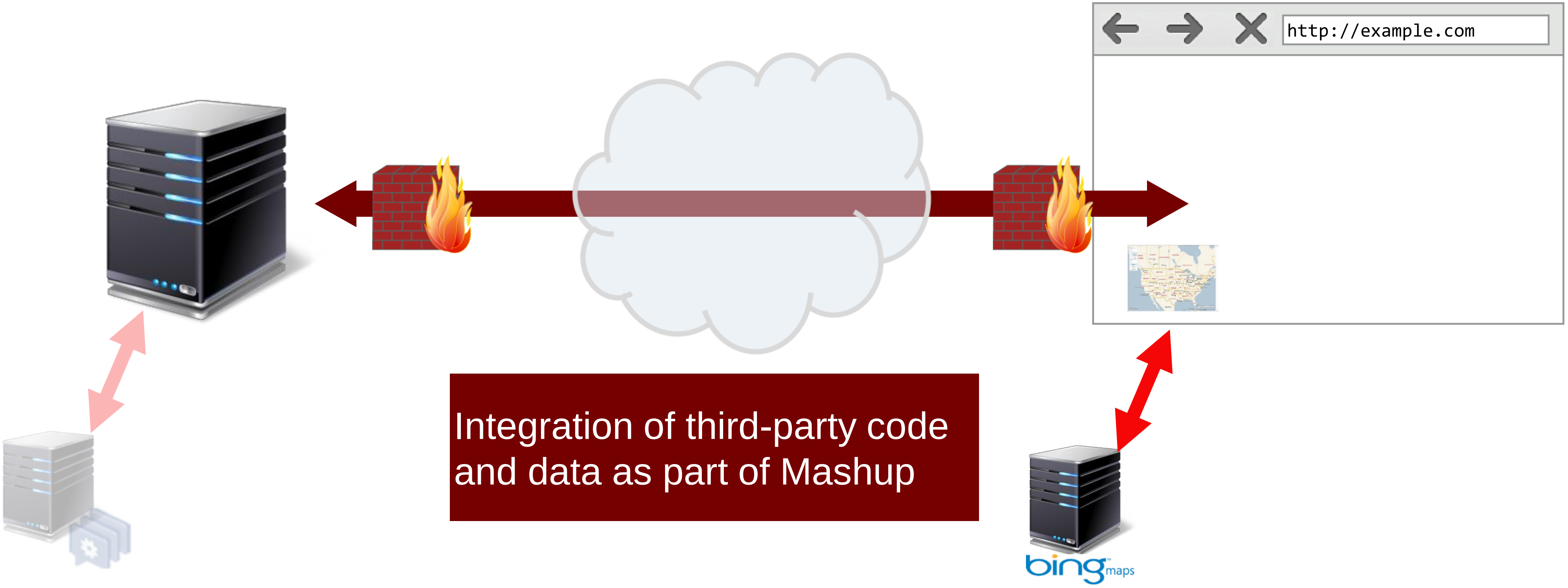


Modern Web Applications



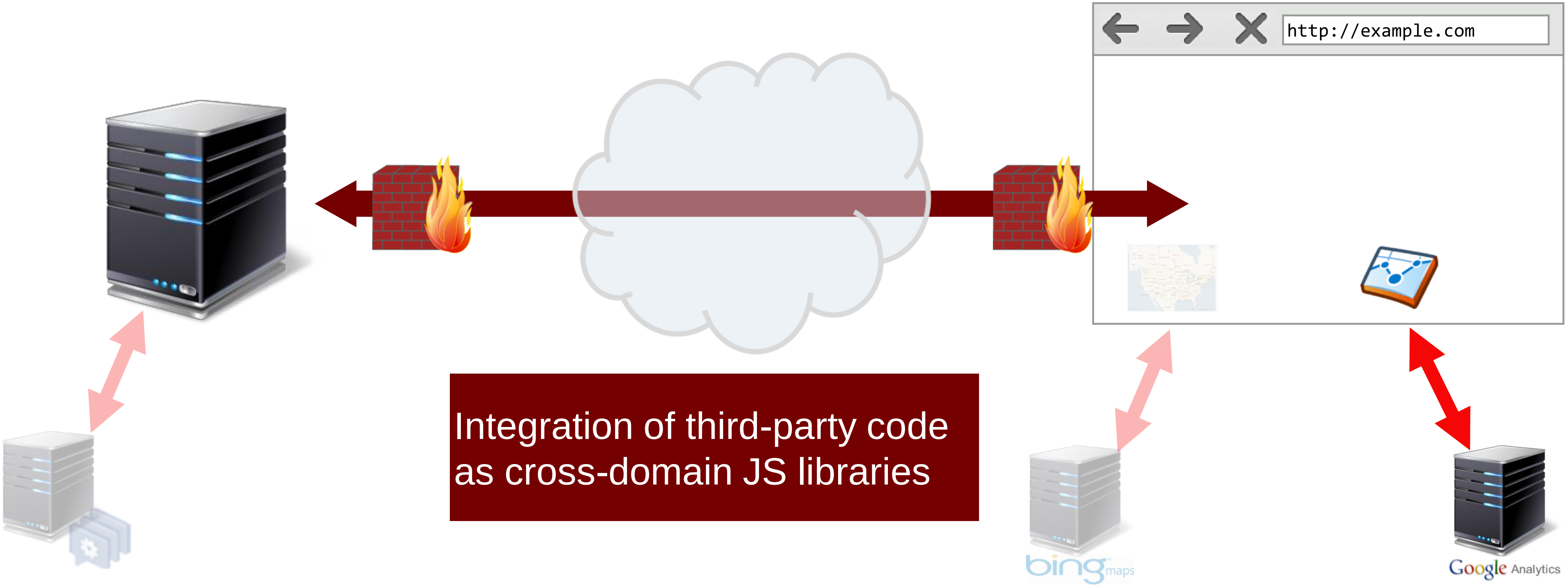
Integration of third-party code (e.g., Facebook apps)

Modern Web Applications



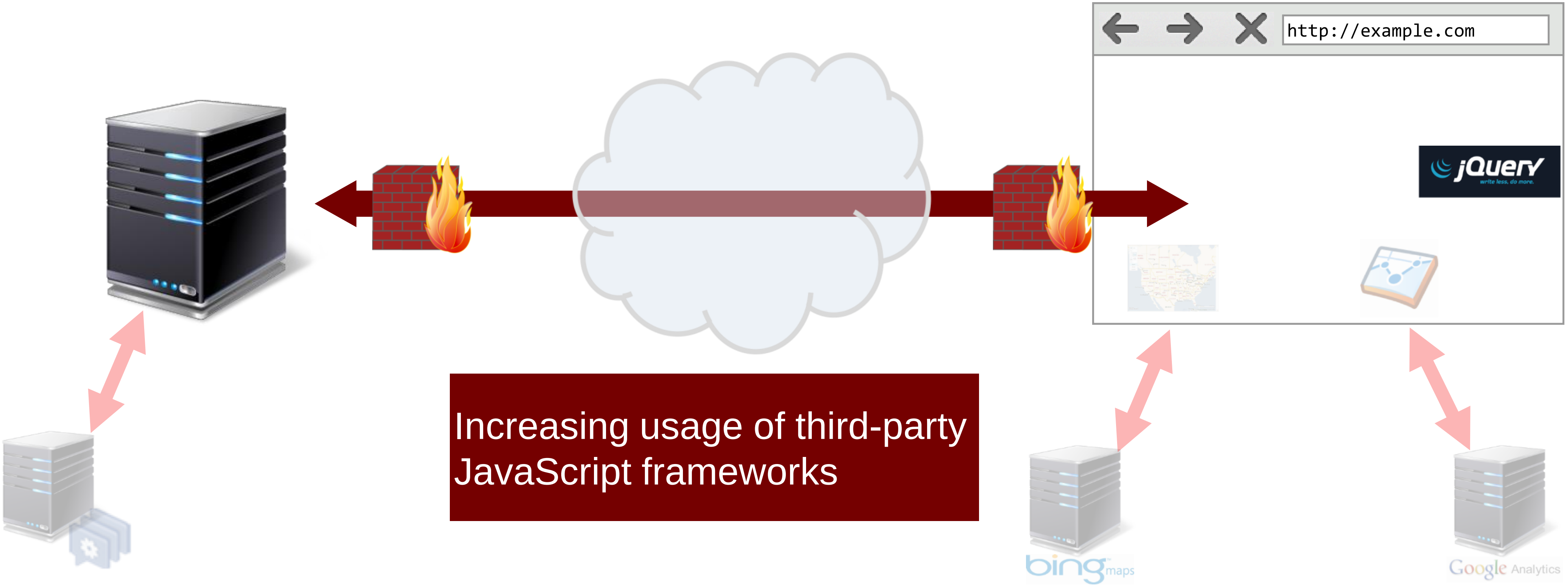
Integration of third-party code and data as part of Mashup

Modern Web Applications



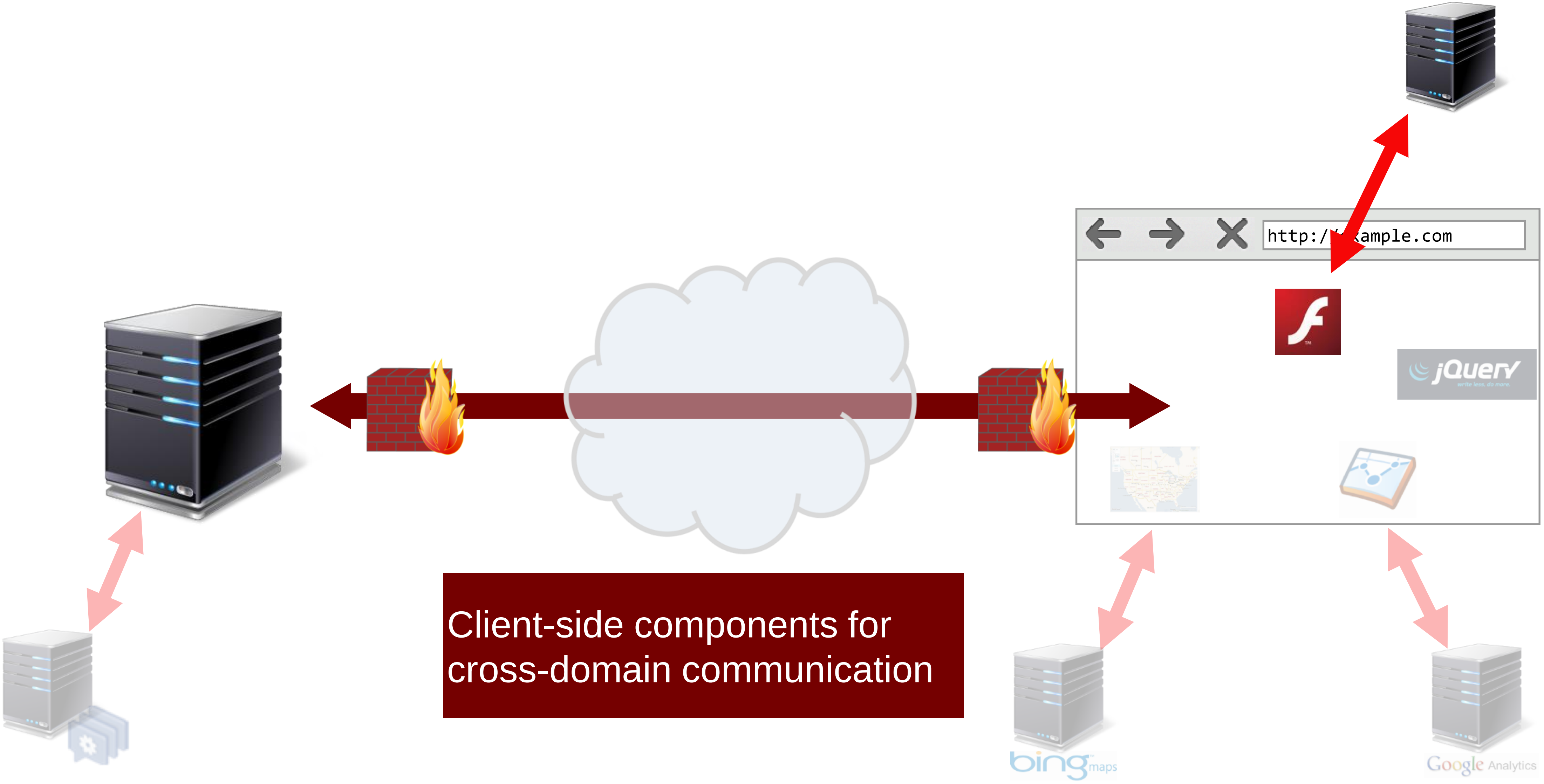
Integration of third-party code as cross-domain JS libraries

Modern Web Applications



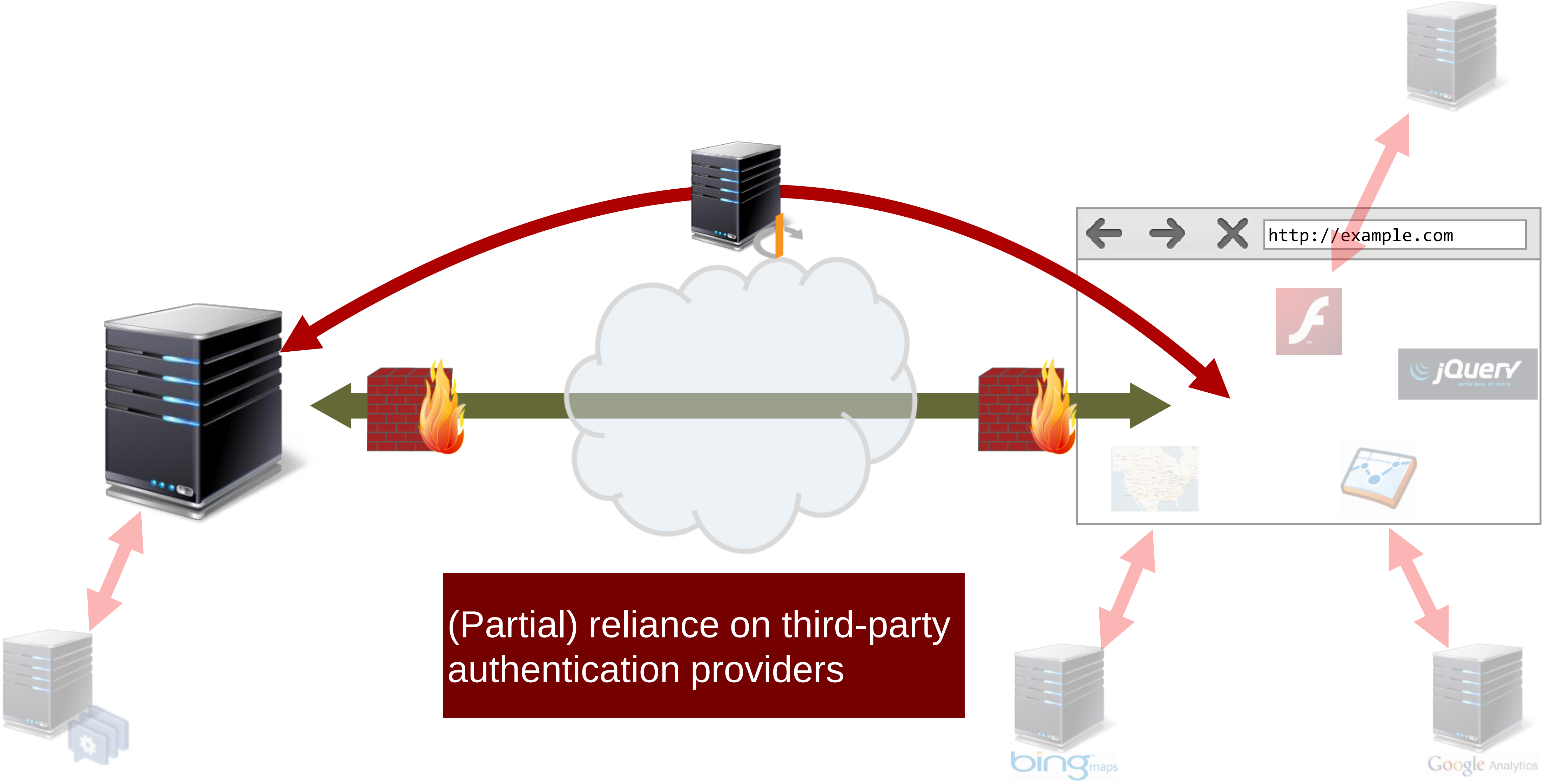
Increasing usage of third-party JavaScript frameworks

Modern Web Applications



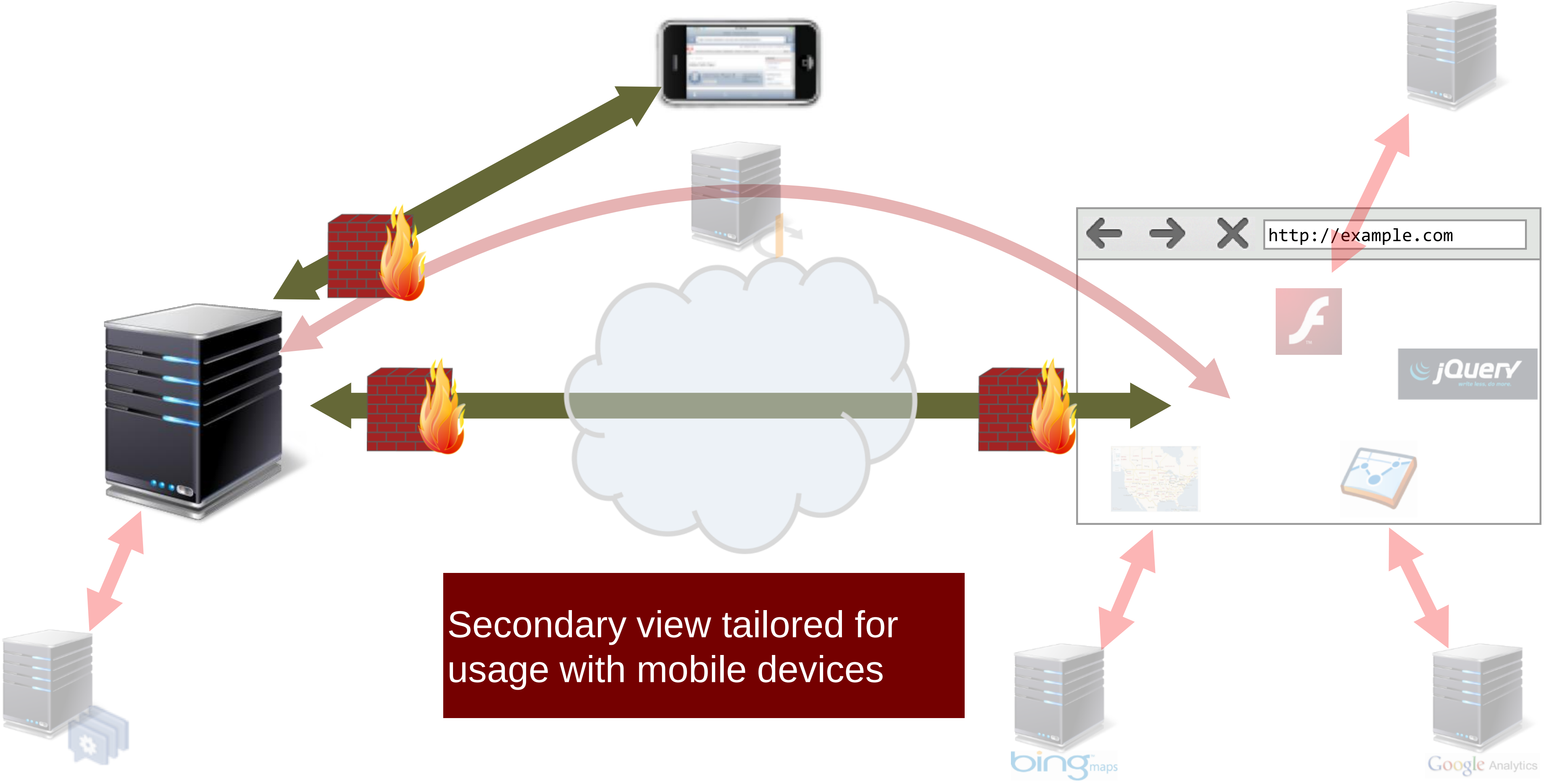
Client-side components for cross-domain communication

Modern Web Applications



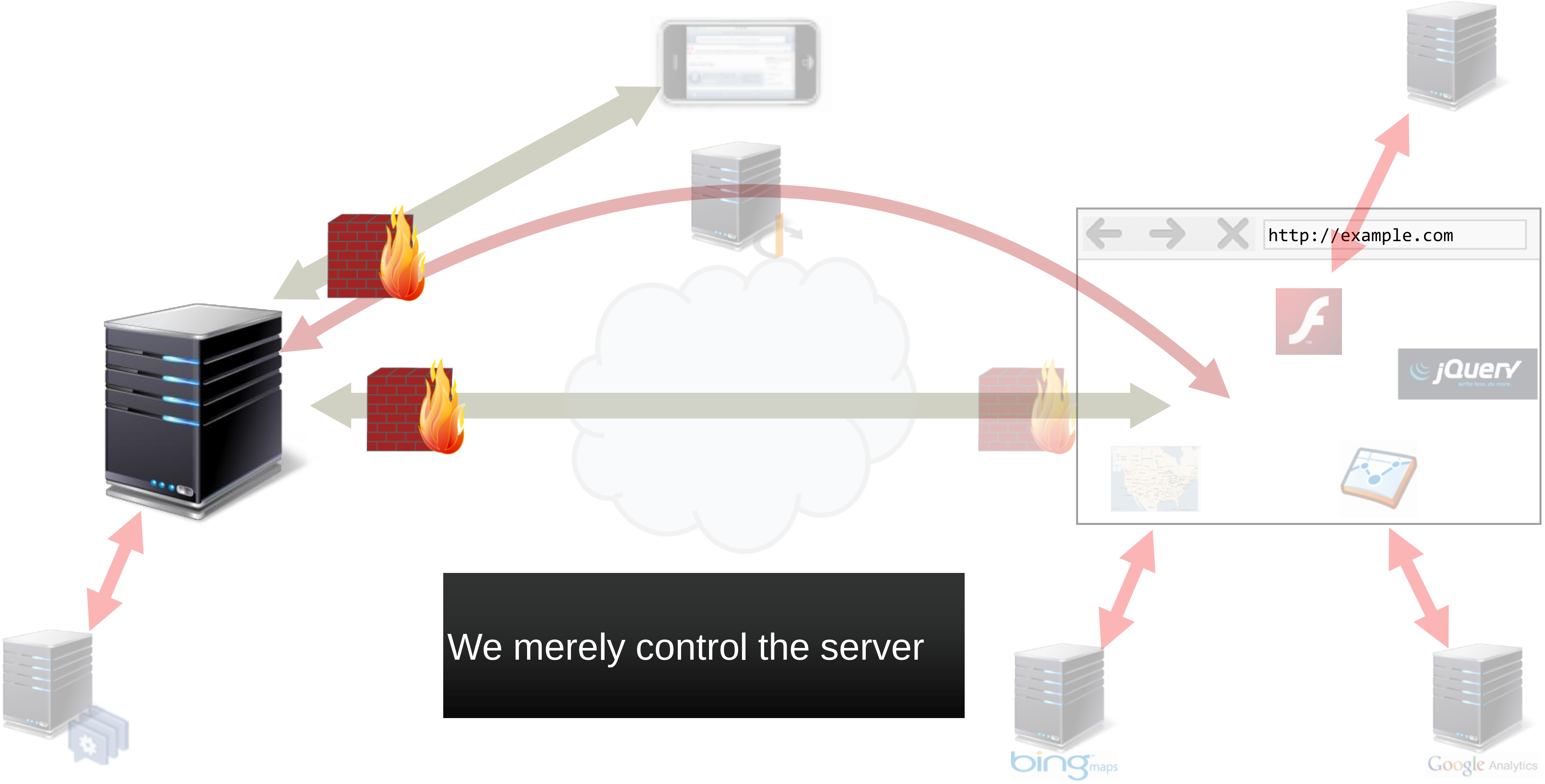
(Partial) reliance on third-party authentication providers

Modern Web Applications



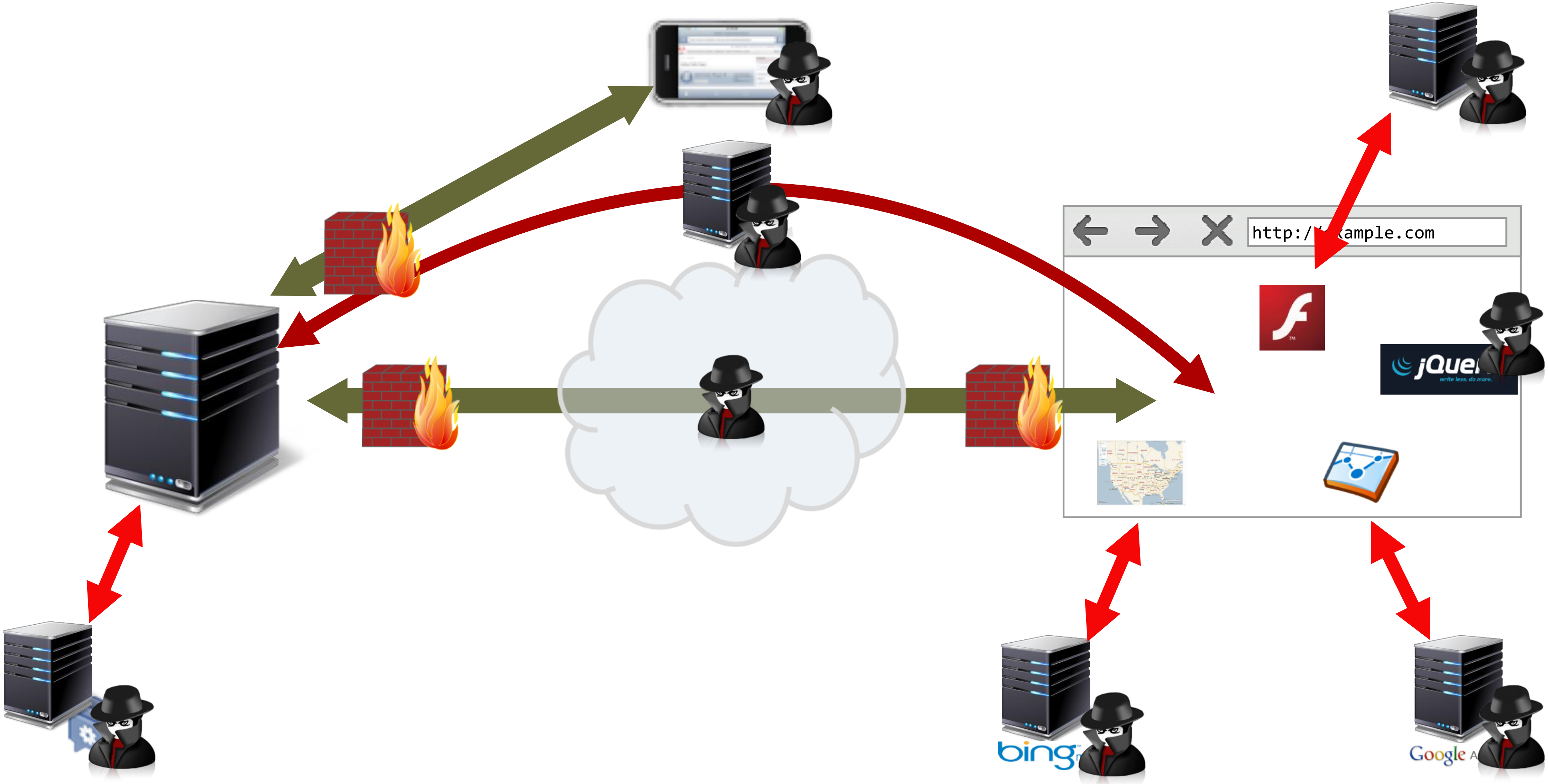
Secondary view tailored for usage with mobile devices

Security Implications



We merely control the server

Possible Attackers on the Web



Network Attacker

- Resides somewhere in the communication link between client and server
- Tries to disturb the confidentiality, integrity, and authenticity of the connection
  - Observation of traffic (passive eavesdropper)
  - Fabrication of traffic (e.g., injecting fake packets)
  - Disruption of traffic (e.g., selective dropping of packets)
  - Modification of traffic (e.g., changing unencrypted HTTP traffic)
- "Man in the middle" (MITM)

Remote Attacker

- Can connect to remote system via the network
  - mostly targets the server
- Attempts to compromise the system (server-side attacks)
  - Arbitrary code execution
  - Information exfiltration (e.g., SQL injections)
  - Information modification
  - Denial of Service

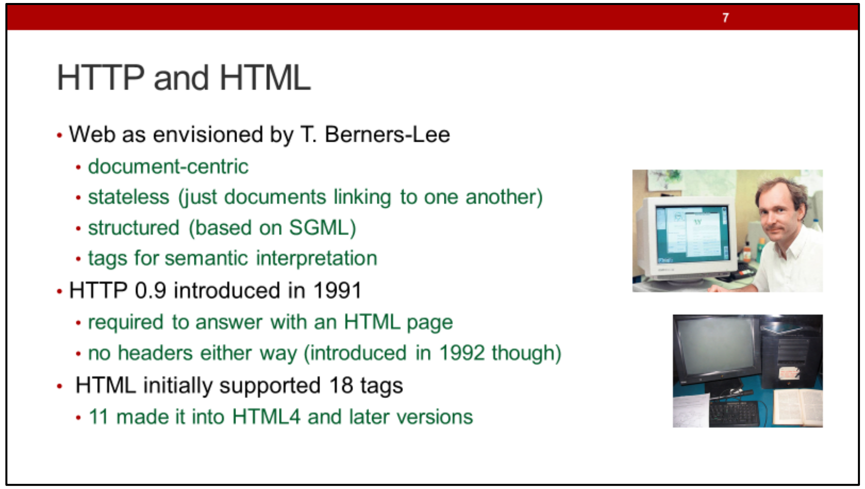
Web Attacker

- Attacker specific to Web applications
- "Man in the browser"
  - can create HTTP requests within user's browser
  - can leverage the user's state (e.g., session cookies)
  - Case of "confused deputy"
- Examples
  - Cross-Site Scripting attacker: can execute arbitrary JavaScript in authenticated user's context
  - Cross-Site Request Forgery attacker: can force user's browser to execute certain operations on vulnerable site

Social Engineering Attacker

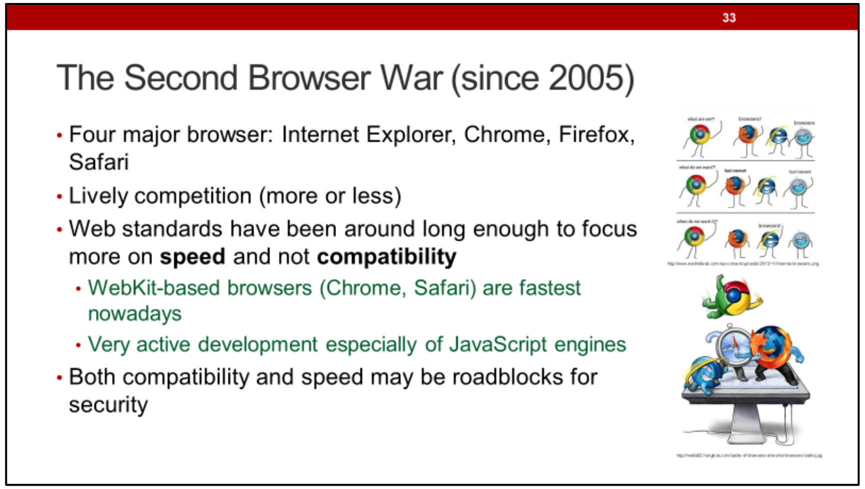
- No real technical capabilities
  - Abusing users rather than software vulnerabilities
- Can lure victim to perform certain tasks
  - Clickjacking
- May use technical measures to ease his task
  - Unicode URLs to easily fake
  - Use well-known icons to suggest "secure" sites

Summary



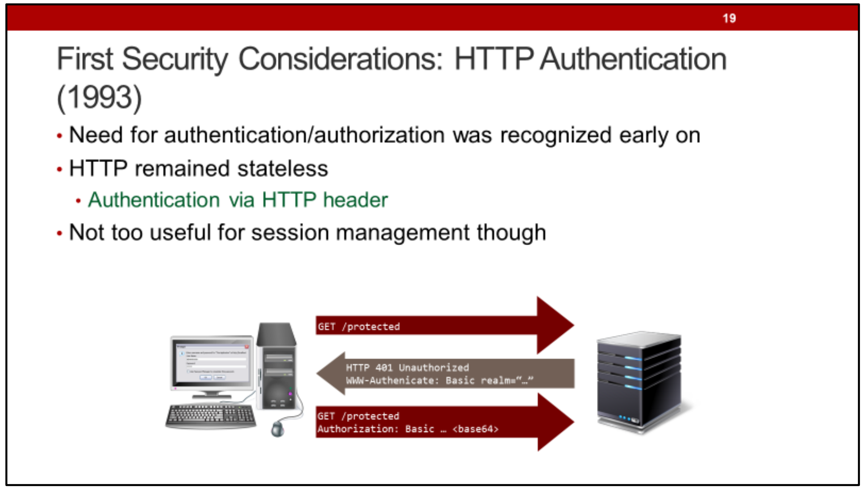
HTTP and HTML

- Web as envisioned by T. Berners-Lee
  - document-centric
  - stateless (just documents linking to one another)
  - structured (based on SGML)
  - tags for semantic interpretation
- HTTP 0.9 introduced in 1991
  - required to answer with an HTML page
  - no headers either way (introduced in 1992 though)
- HTML initially supported 18 tags
  - 11 made it into HTML4 and later versions



The Second Browser War (since 2005)

- Four major browser: Internet Explorer, Chrome, Firefox, Safari
- Lively competition (more or less)
- Web standards have been around long enough to focus more on **speed** and not **compatibility**
  - WebKit-based browsers (Chrome, Safari) are fastest nowadays
  - Very active development especially of JavaScript engines
- Both compatibility and speed may be roadblocks for security



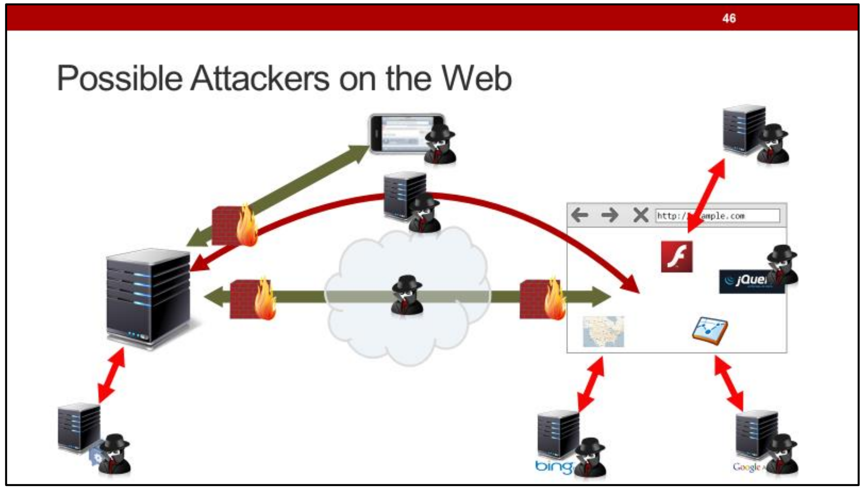
First Security Considerations: HTTP Authentication (1993)

- Need for authentication/authorization was recognized early on
- HTTP remained stateless
  - Authentication via HTTP header
- Not too useful for session management though

```
GET /protected
```

```
HTTP 401 Unauthorized
WWW-Authenicate: Basic realm="…"
```

```
GET /protected
Authorization: Basic … <base64>
```



Possible Attackers on the Web

Credits

- Original slide deck by Ben Stock
- Modified by Nick Nikiforakis