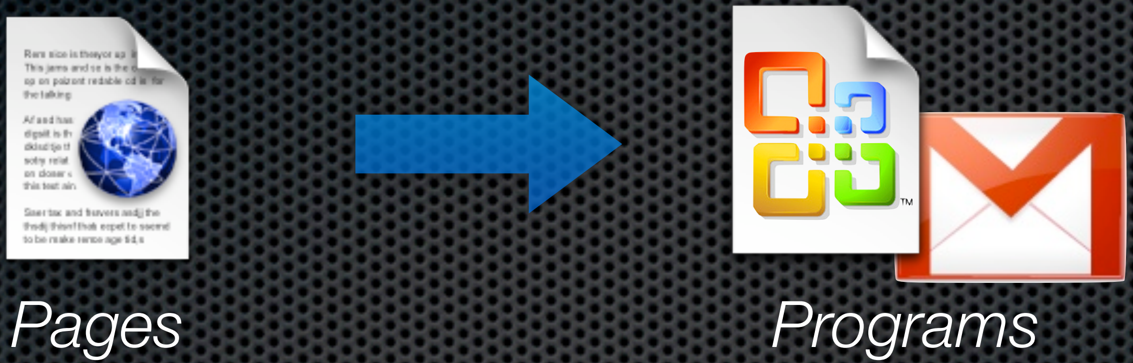
Building a Safer Web

Charles Reis

University of Washington CSE

Web is Evolving



- **More complex, active content**
- **Browser now in role of OS, but not yet safe**
  - Browsers aren't built for programs
  - Web content faces real challenges

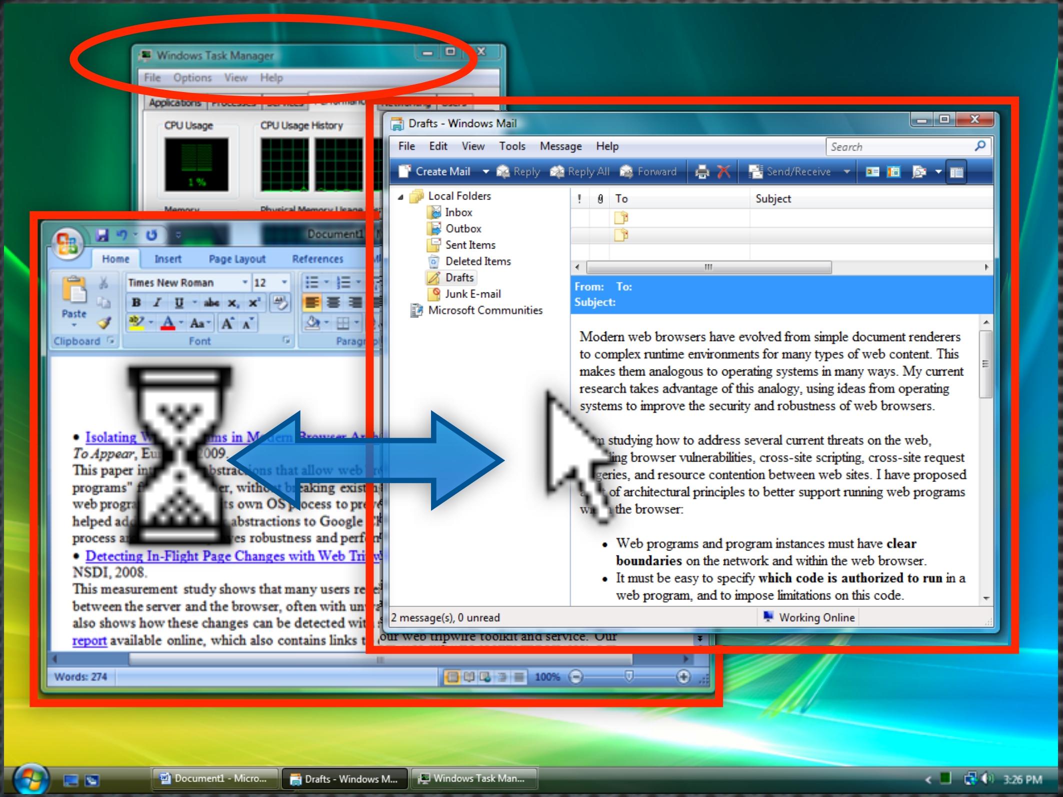
My Contributions

Problems	*Projects*
Program Interference	Multi-Process Browsers [EuroSys ’09]
In-Flight Page Changes	Web Tripwires [NSDI ’08]
Poor Program Support	Architectural Principles [HotNets ’07]
XSS	Script Whitelists
Browser Exploits	BrowserShield [OSDI ’06]

Range of Project Types

Program Interference	Multi-Process Browsers	**Practical, deployed in Google Chrome**
In-Flight Page Changes	Web Tripwires	**Measurement study of 50,000 clients**
Poor Program Support	Architectural Principles	**Position paper**
XSS	Script Whitelists	**Research prototype**
Browser Exploits	BrowserShield	**Prototype, influenced Web Sandbox**

Consider OS Landscape



Modern web browsers have evolved from simple document renderers to complex runtime environments for many types of web content. This makes them analogous to operating systems in many ways. My current research takes advantage of this analogy, using ideas from operating systems to improve the security and robustness of web browsers.

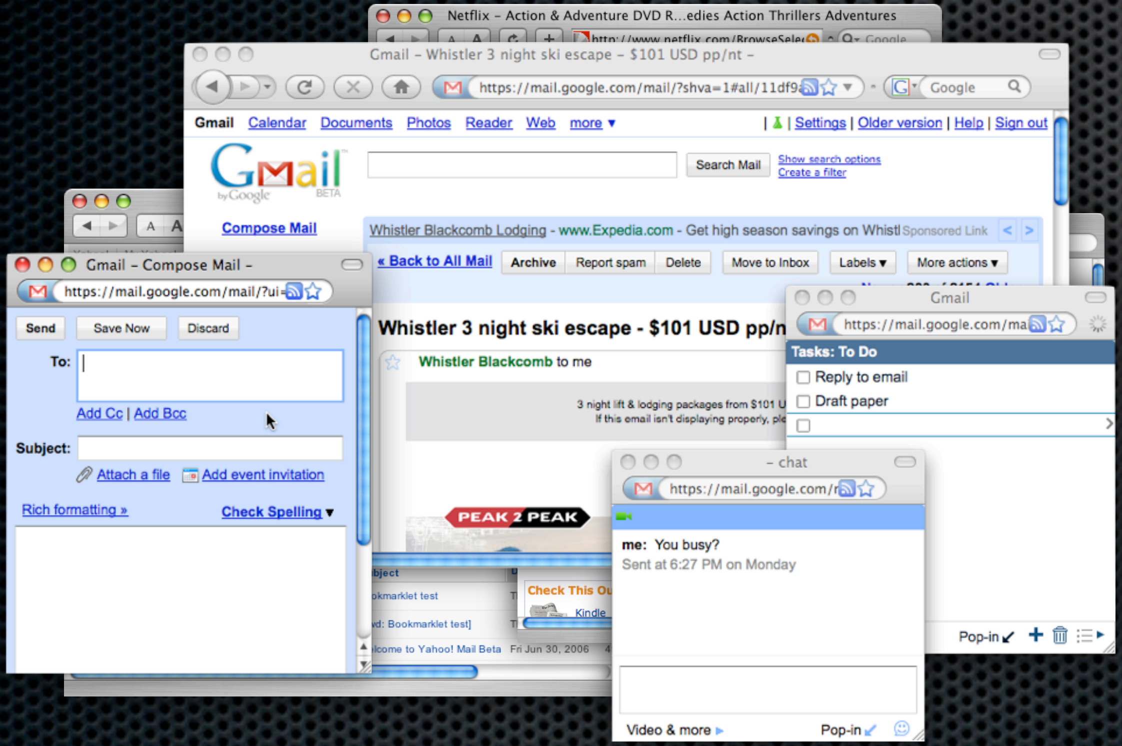
…m studying how to address several current threats on the web, …ding browser vulnerabilities, cross-site scripting, cross-site request …geries, and resource contention between web sites. I have proposed a… of architectural principles to better support running web programs wi… the browser:

- Web programs and program instances must have **clear boundaries** on the network and within the web browser.
- It must be easy to specify **which code is authorized to run** in a web program, and to impose limitations on this code.

- Isolating W… …ms in Modern Browser Ar…
To Appear, Eu… 2009.
This paper in… …bstractions that allow web… programs" f… …er, without breaking exist… web progra… …ts own OS process to pre… helped ad… …abstractions to Google C… process a… …es robustness and perf…
- Detecting In-Flight Page Changes with Web Tri…
NSDI, 2008.
This measurement study shows that many users re… between the server and the browser, often with un… also shows how these changes can be detected wit… report available online, which also contains links t… our web tripwire toolkit and service. Our

- Performance isolation
- Resource management
- Failure isolation
- **Clear program abstraction**

Browsers Fall Short


Netflix – Action & Adventure DVD R...edies Action Thrillers Adventures
Gmail – Whistler 3 night ski escape – $101 USD pp/nt –
https://mail.google.com/mail/?shva=1#all/11df9
Gmail Calendar Documents Photos Reader Web more ▾
Settings | Older version | Help | Sign out
Search Mail
Show search options
Create a filter
Compose Mail
Whistler Blackcomb Lodging - www.Expedia.com - Get high season savings on Whistl Sponsored Link
« Back to All Mail
Archive
Report spam
Delete
Move to Inbox
Labels ▾
More actions ▾
Gmail – Compose Mail –
Send
Save Now
Discard
To:
Add Cc | Add Bcc
Subject:
Attach a file
Add event invitation
Rich formatting »
Check Spelling ▾
Gmail
Tasks: To Do
Pop-in
– chat
Video & more
Pop-in

Whistler 3 night ski escape - $101 USD pp/n

Whistler Blackcomb to me

3 night lift & lodging packages from $101 U
If this email isn't displaying properly, ple

Reply to email
Draft paper

me: You busy?
Sent at 6:27 PM on Monday

- Unresponsiveness
- Jumbled accounting
- Browser crashes
- **Unclear what a program is!**

Thesis: Learn from the OS

- **Improve browser and web content architecture**
  - Define a precise program abstraction
  - Isolate programs from each other
  - Make it possible to authorize program code
  - Interpose on program behavior

[HotNets '07]

Outline

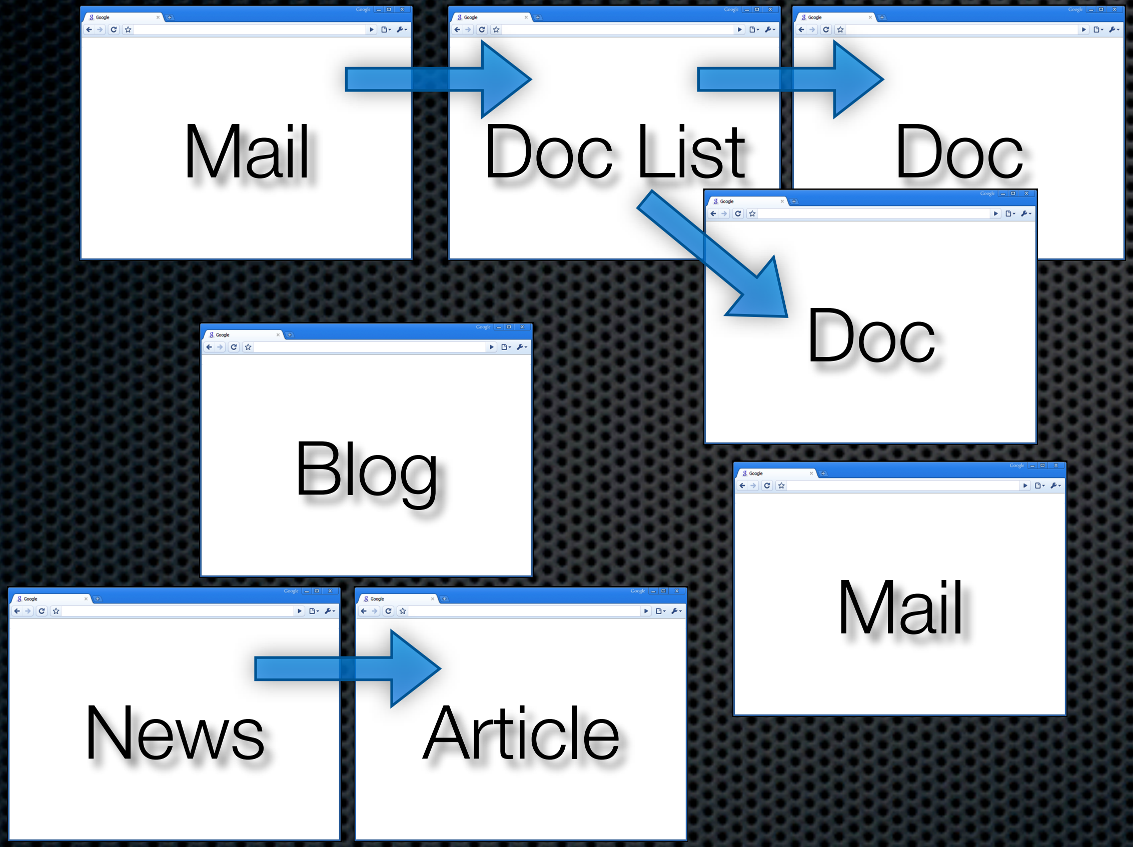
- **Browser Architecture: Chromium**
  - Define program abstractions
  - Isolate programs from each other

Web Tripwires

Previous Work

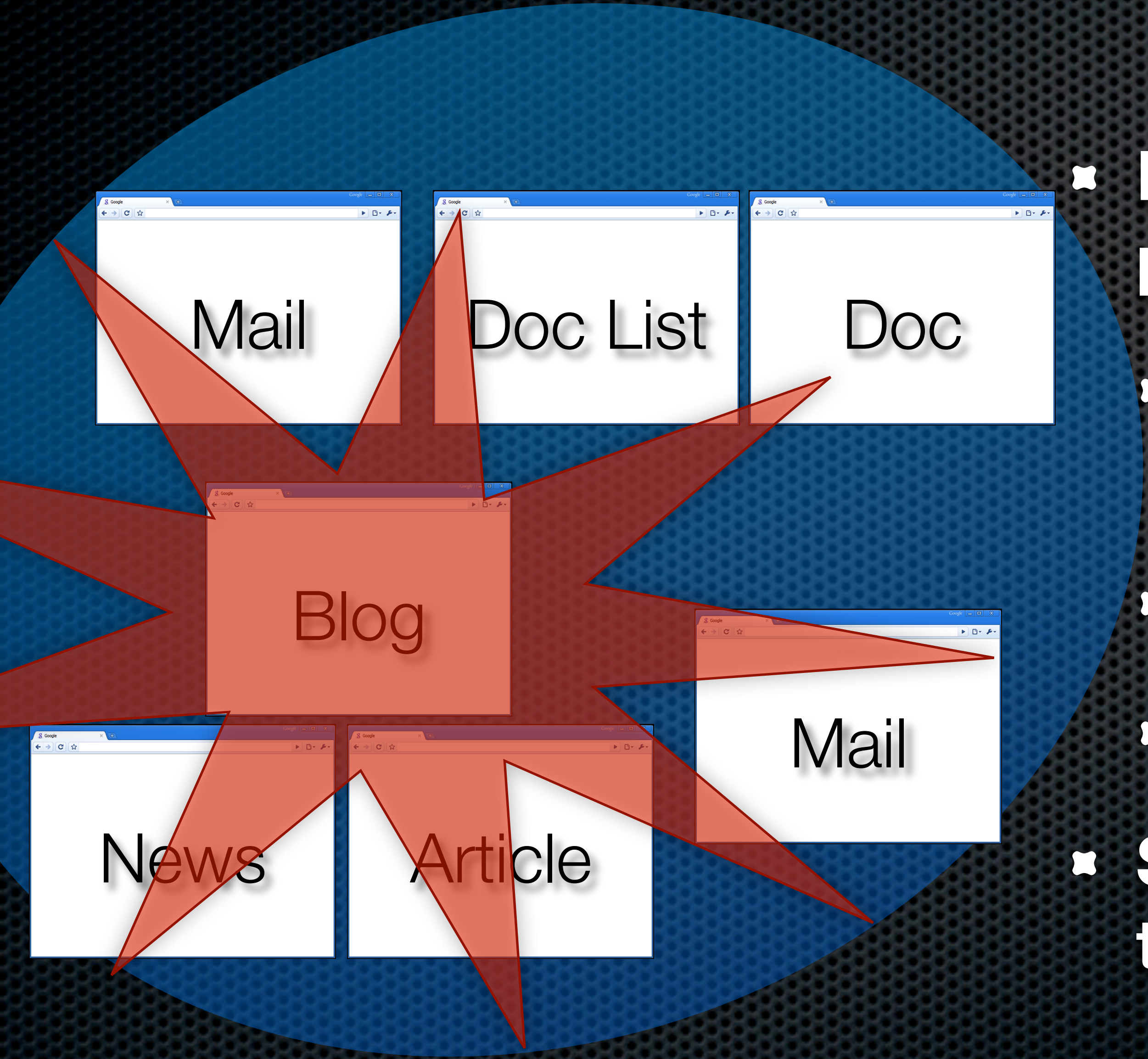
Future Directions

Programs in the Browser



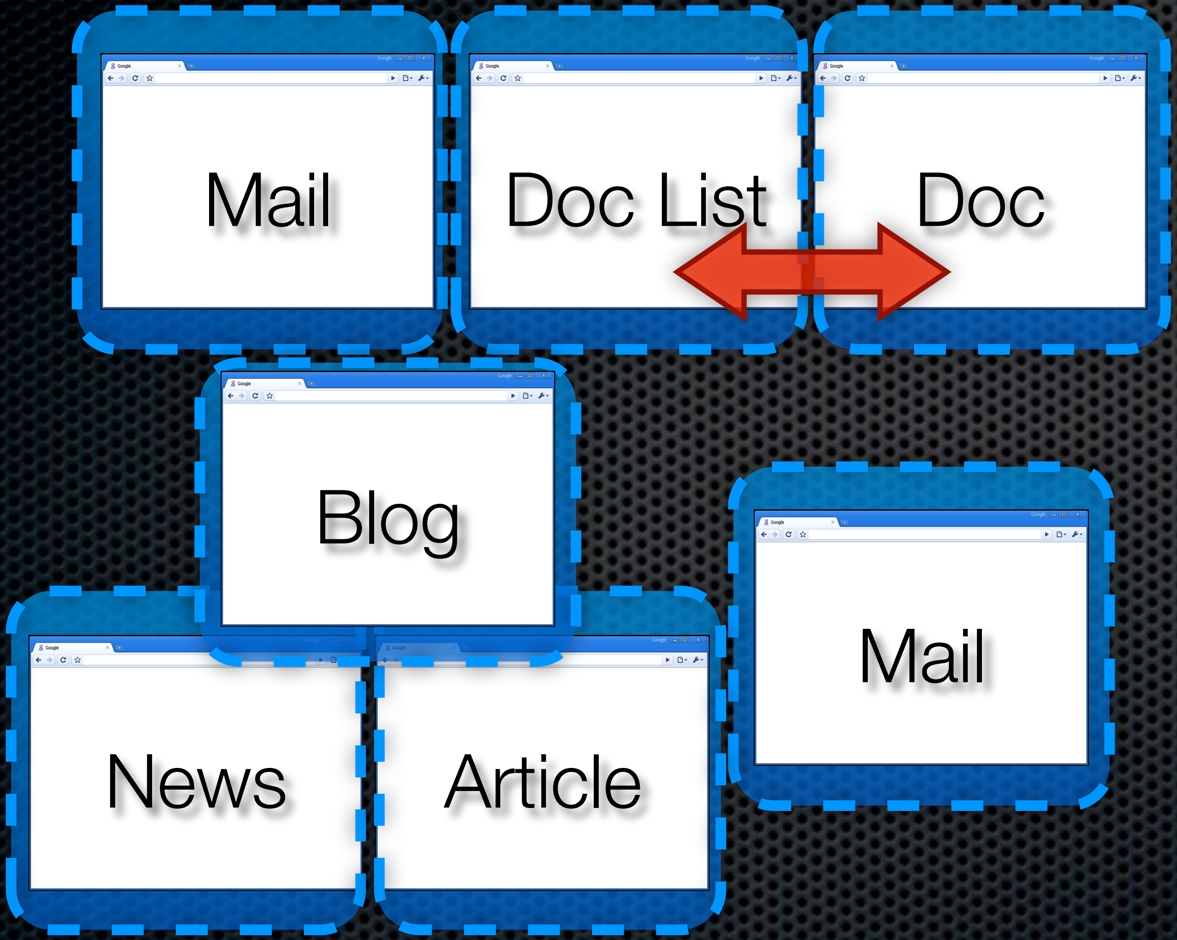
- Consider an example browsing session
  - Several independent programs

Monolithic Browsers



- **Most browsers put all pages in one process**
  - Poor performance isolation
  - Poor failure isolation
  - Poor security
- **Should re-architect the browser**

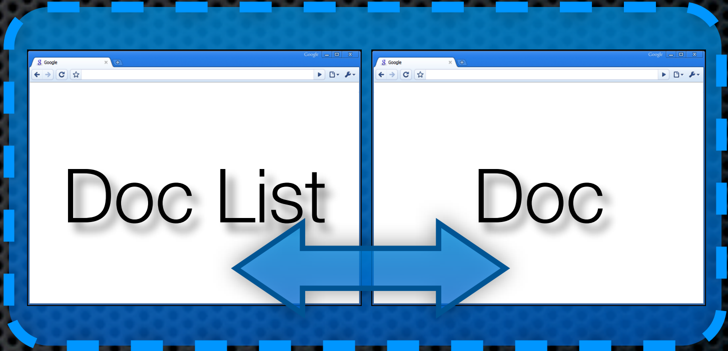
Process per Window?



- **Breaks pages** that directly communicate
  - Shared access to data structures, etc.
  - **Connected** pages from **same-origin**
- **Fails as a program abstraction**

Need a Program Abstraction

- Aim for **new groupings** that:
  - **Match our intuitions**
  - **Preserve compatibility**
- Take cues from browser's existing rules
- Isolate each grouping in an OS process
- Will get **performance and failure isolation**, but not security between sites



Outline

Browser Architecture

Program Abstractions

Program Isolation

Evaluation

Ideal Abstractions

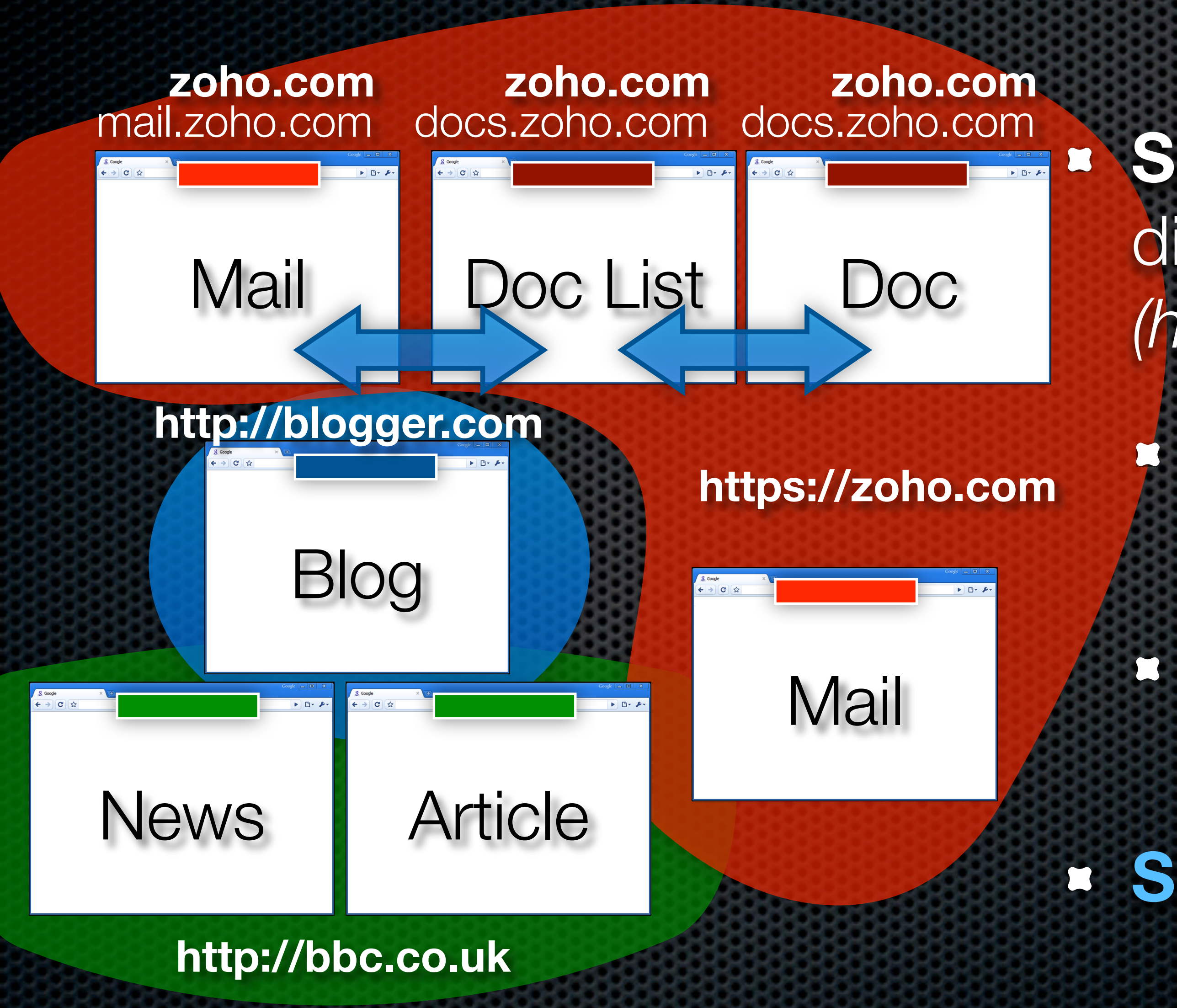
- **Web Program**
  - Set of pages and sub-resources providing a service
- **Web Program Instance**
  - Live copy of a web program in the browser
  - Will be isolated in the browser's architecture

Intuitive, but how to define concretely?

Compatible Abstractions

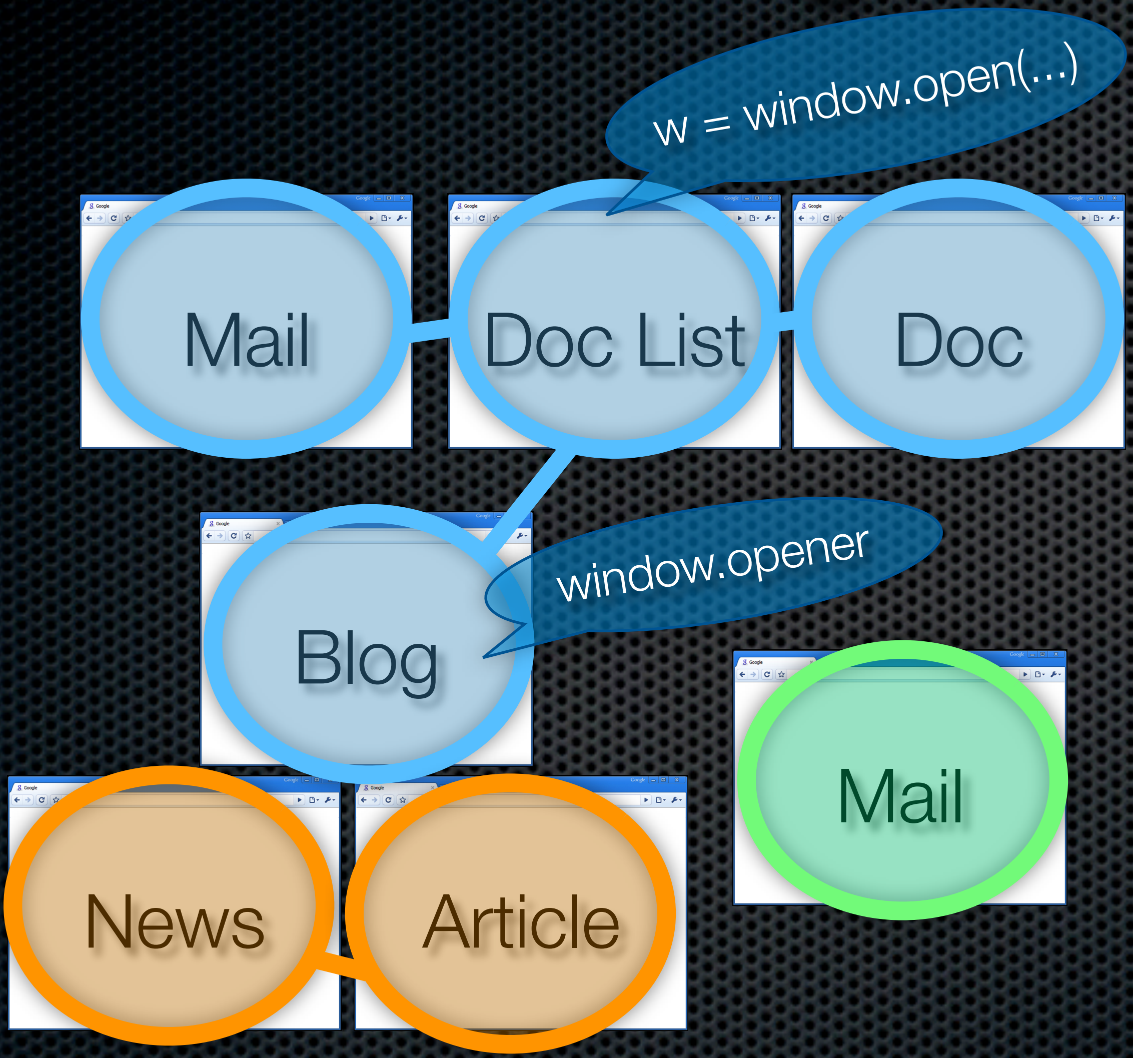
- Three ways to group pages into processes:
  1. **Site:** based on browser's *access control policies*
  2. **Browsing Instance:** *communication channels* between pages
  3. **Site Instance:** intersection of the first two

1. Sites



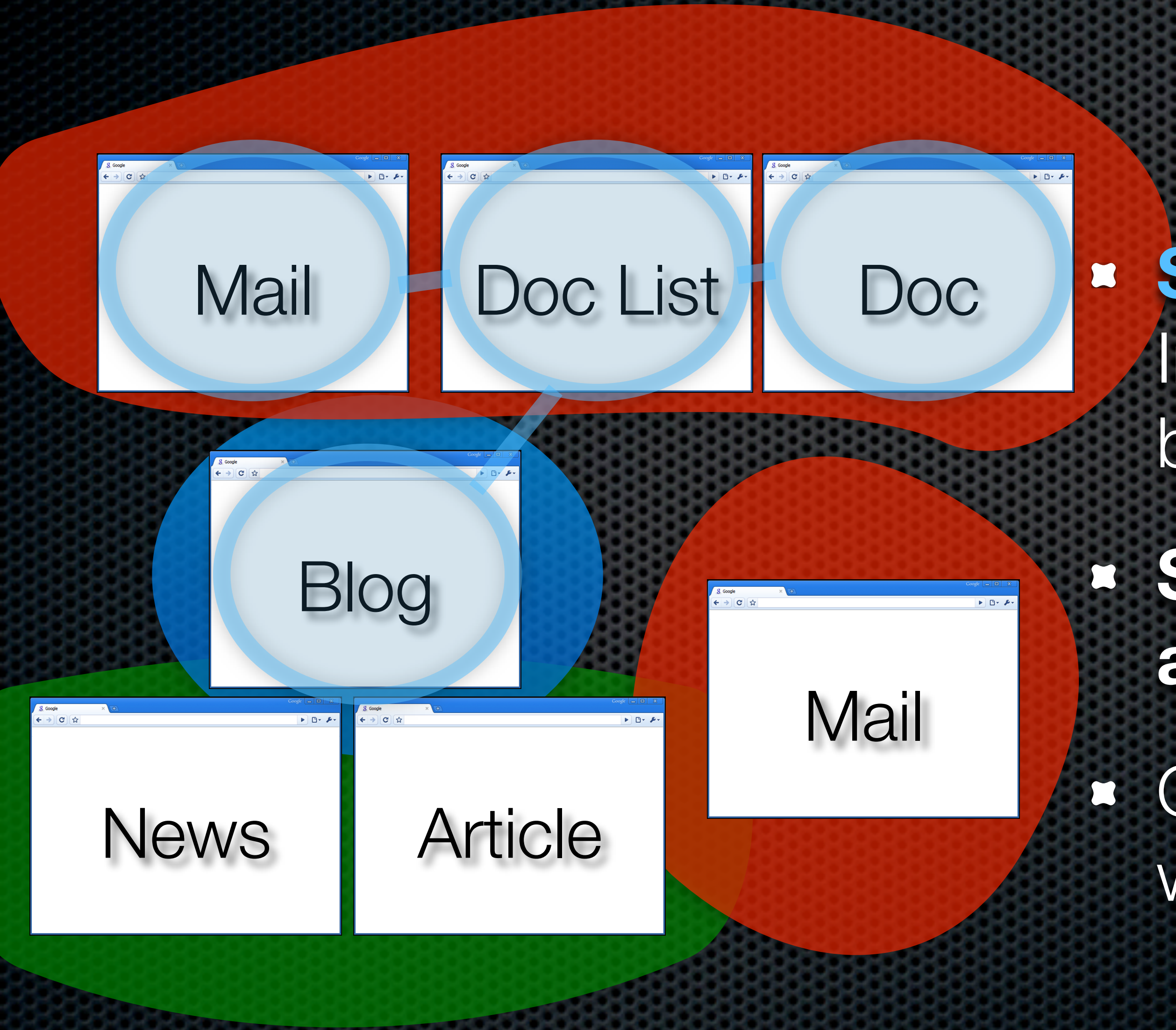
- **Same Origin Policy** dictates some isolation *(host+protocol+port)*
  - Pages can change document.domain
  - *Registry-controlled domain name* limit
- **Site:** RCDN + protocol

2. Browsing Instances



- Not all pages can talk
- References between "related" windows
  - Parents and children
  - Lifetime of window
- **Browsing Instance:** connected windows, regardless of site

3. Site Instances



- **Site Instance:** Intersection of site & browsing instance
- **Safe to isolate from any other pages**
- Compatible notion of a web program instance

Abstractions Recap

- **Site**
  - e.g., All pages from https://bbc.co.uk
- **Browsing Instance**
  - Windows with script references to each other
- **Site Instance**
  - Connected, same-site pages

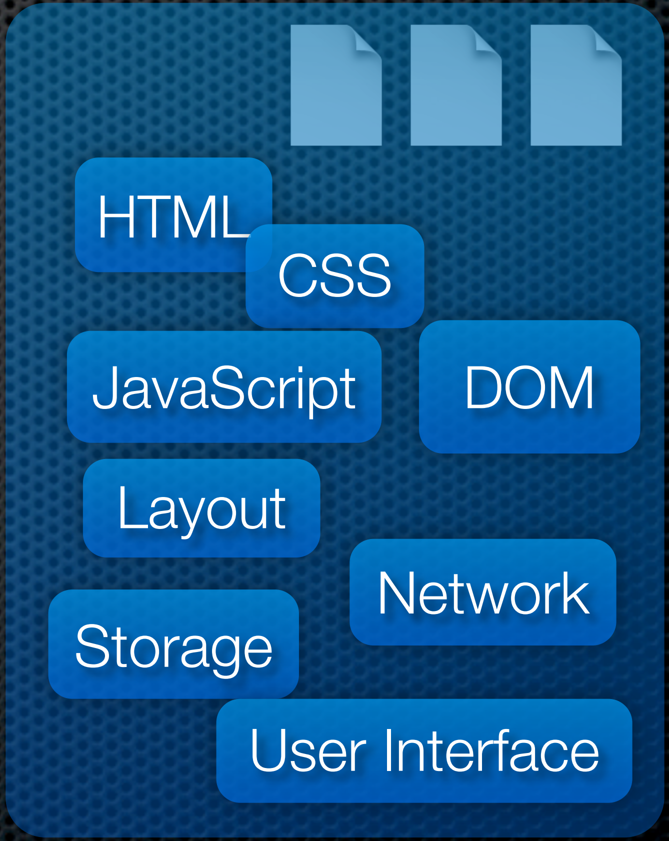
Compatibility Compromises

- **Coarse granularity**
  - Some logical apps grouped together (instances help)
- **Imperfect isolation**
  - Shared cookies, some window-level JS calls
- **Not a secure boundary**
  - Must still rely on renderer to prevent certain leaks

Outline

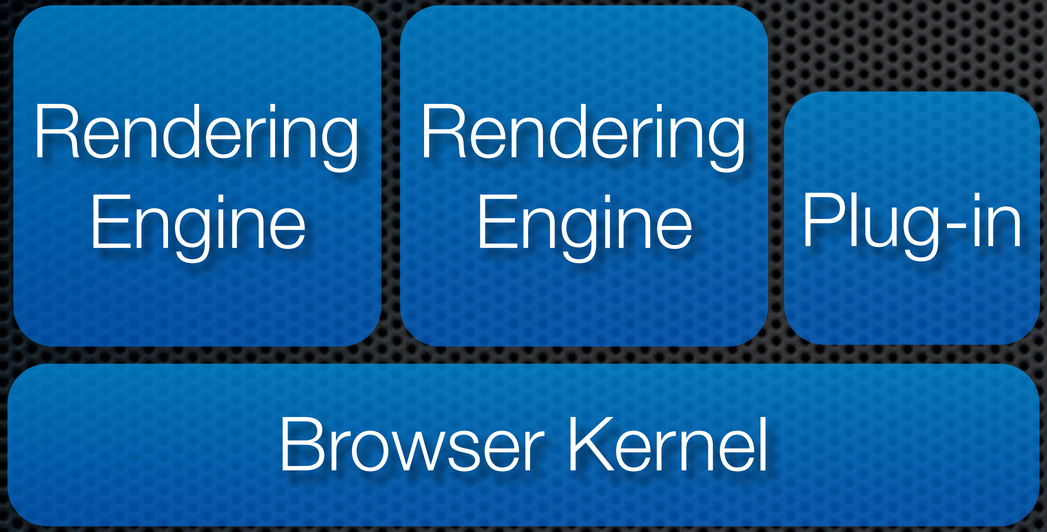
Most Browsers are Monolithic

- All browser parts in one process
- Could divide into separate modules
  - **Isolate with OS processes:** address spaces, concurrency, failure isolation



One OS Process

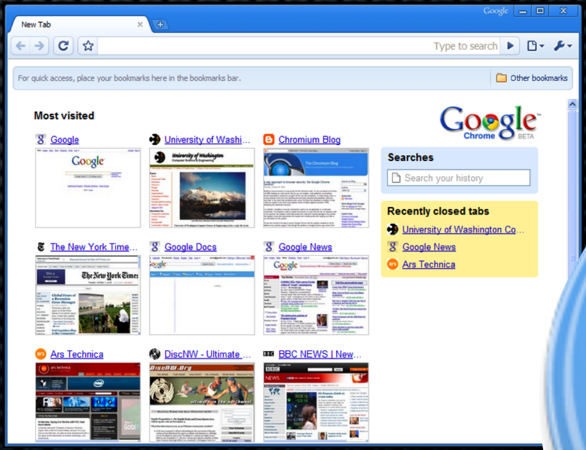
Multi-Process Browser



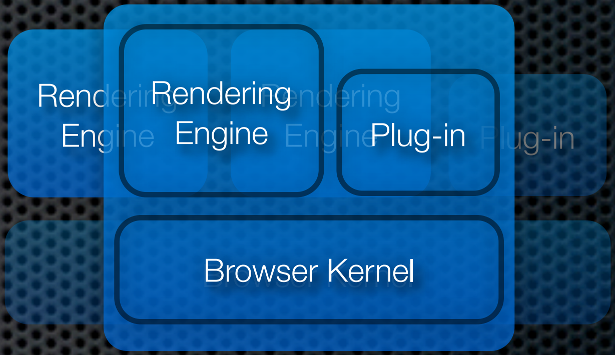
- **Browser Kernel**
  - Storage, network, UI
- **Rendering Engines**
  - Web program and runtime environment
- **Plug-ins**

Implementations

- **Konqueror Prototype** (2006)
  - Proof of concept on Linux
- **Chromium** (Google Chrome, 2008)
  - Added support for Site Instance isolation (including creating processes during navigations)



Chromium Process Models



1. **Monolithic**
2. **Process-per-Browsing-Instance**
   - New window = new renderer process
3. **Process-per-Site-Instance** *(default)*
   - Create renderer process when navigating cross-site
4. **Process-per-Site**
   - Combine instances: fewer processes, less isolation

Implementation Caveats

- **Sites may sometimes share processes**
  - Not all cross-site navigations change processes
  - Frames still in parent process
  - Process limit (20), then randomly re-used

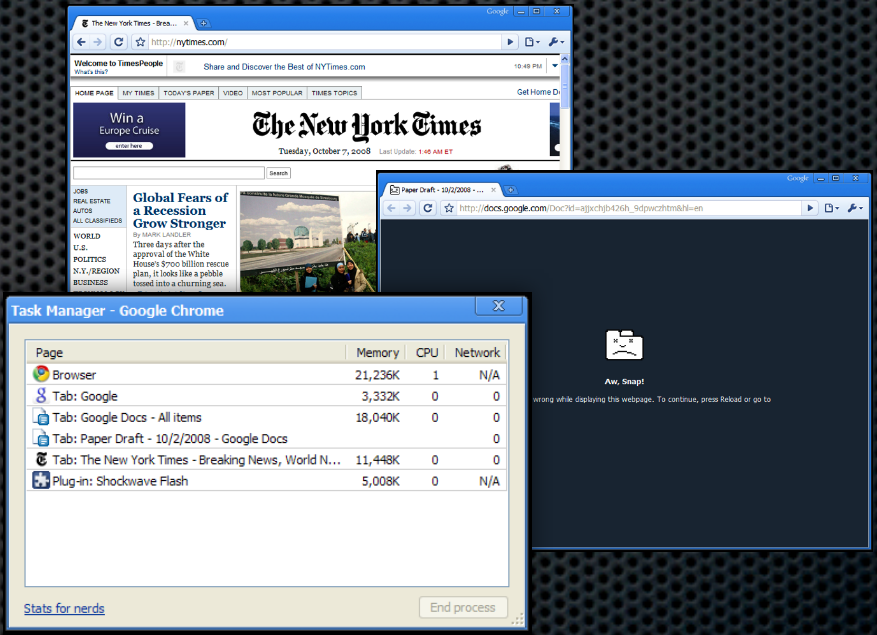
Outline

- Browser Architecture
- Program Abstractions
- Program Isolation
- **Evaluation**

Robustness Benefits

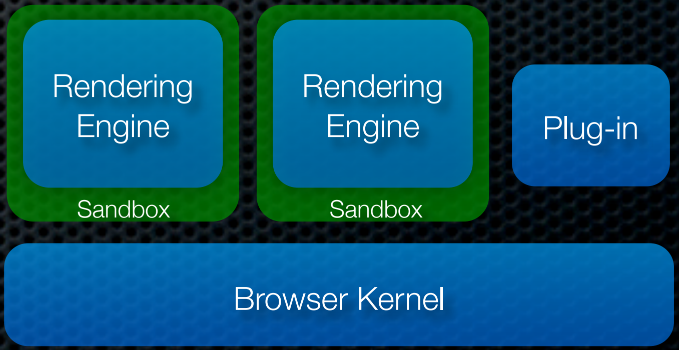
- Failure Isolation
- Accountability
- Memory Management

- Some additional security (e.g., Chromium’s sandbox)


The New York Times - Brea...
http://nytimes.com/
The New York Times
Paper Draft - 10/2/2008 - ...
http://docs.google.com/Doc?id=ajjxchjb426h_9dpwczhtm&hl=en
Aw, Snap!
Stats for nerds
End process

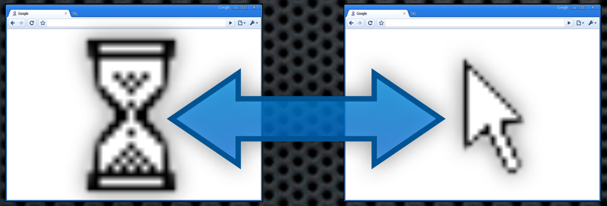
Task Manager - Google Chrome

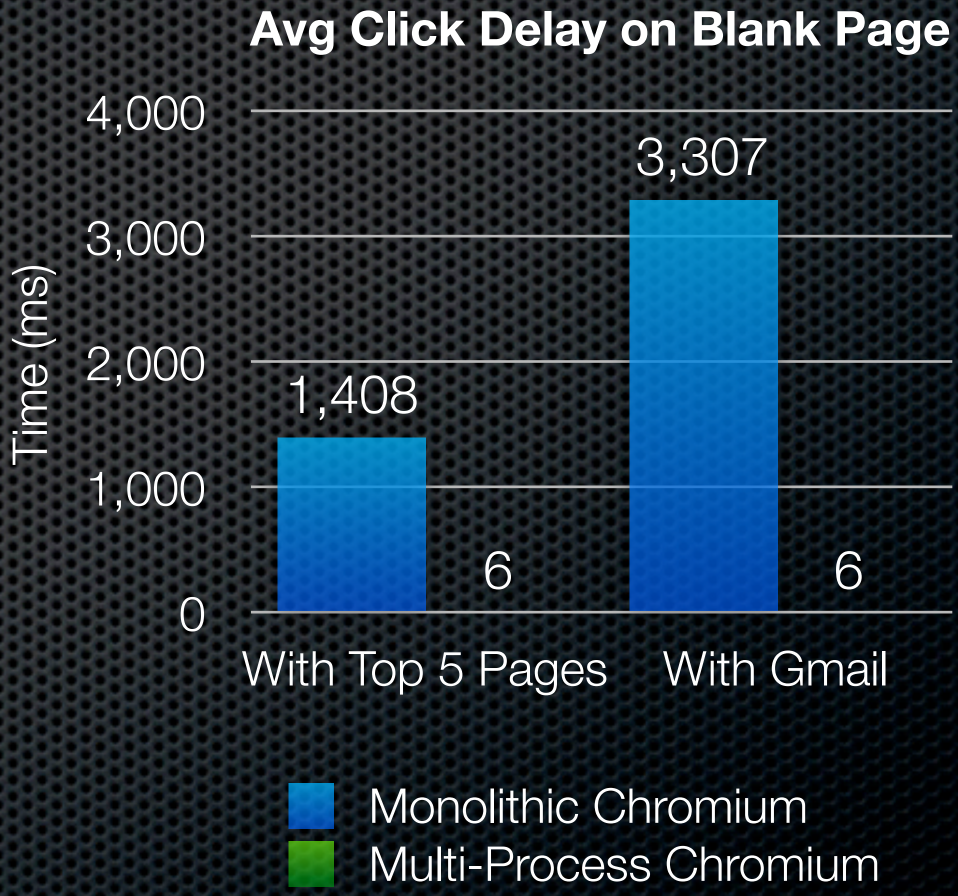
Page	Memory	CPU	Network
Browser	21,236K	1	N/A
Tab: Google	3,332K	0	0
Tab: Google Docs - All items	18,040K	0	0
Tab: Paper Draft - 10/2/2008 - Google Docs			0
Tab: The New York Times - Breaking News, World N...	11,448K	0	0
Plug-in: Shockwave Flash	5,008K	0	N/A



Performance Isolation

- **Responsive** while other web programs working
  - No click latency



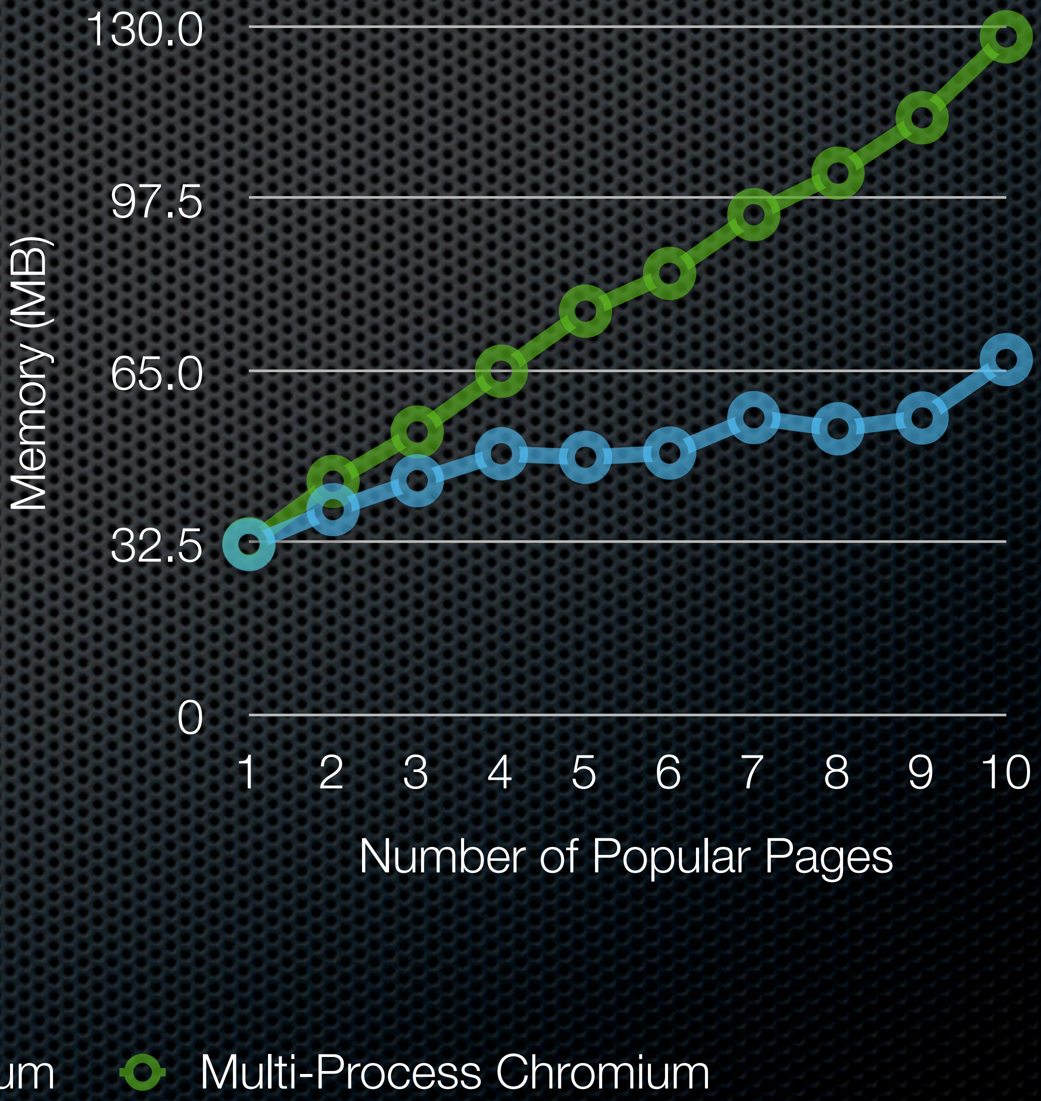


Other Performance Impact

- **Speedups**
  - More work done concurrently, leveraging cores
  - e.g., Session restore of several tabs
- **Process Latency**
  - 100 ms, but masked by other speedups in practice

Memory Overhead

- Robustness benefits do have a cost
  - Reasonable for many real users



Compatibility Evaluation

- No known compat bugs due to architecture
  - Distributed tests check top million pages
- Some minor behavior changes
  - e.g., **Narrower scope of window names:** browsing instance, not global

Related Architecture Work

- **Internet Explorer 8**
  - Multi-process architecture, no program abstractions
- **Gazelle**
  - Like Chromium, but values security over compatibility
- **Other research: OP, Tahoma, SubOS**
  - Break compatibility (isolation too fine-grained)

Summary

- Browsers must recognize programs to support them
    - **Site Instances** capture this
    - **Compatible** with existing web content
    - Can prevent interference with **process isolation**

Outline

Browser Architecture

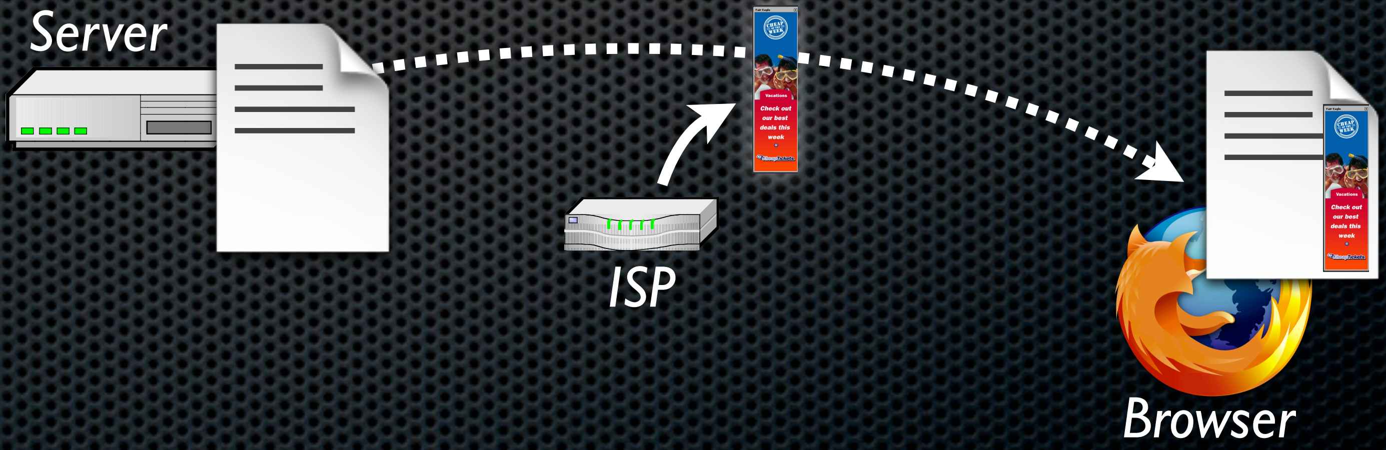
- **Web Tripwires**
  - Simple integrity checks to protect programs

Previous Work

Future Directions

Web Program Integrity

- Can users or publishers trust web program contents?
  - HTTP can be **modified in-flight**
  - Changes become part of the site instance

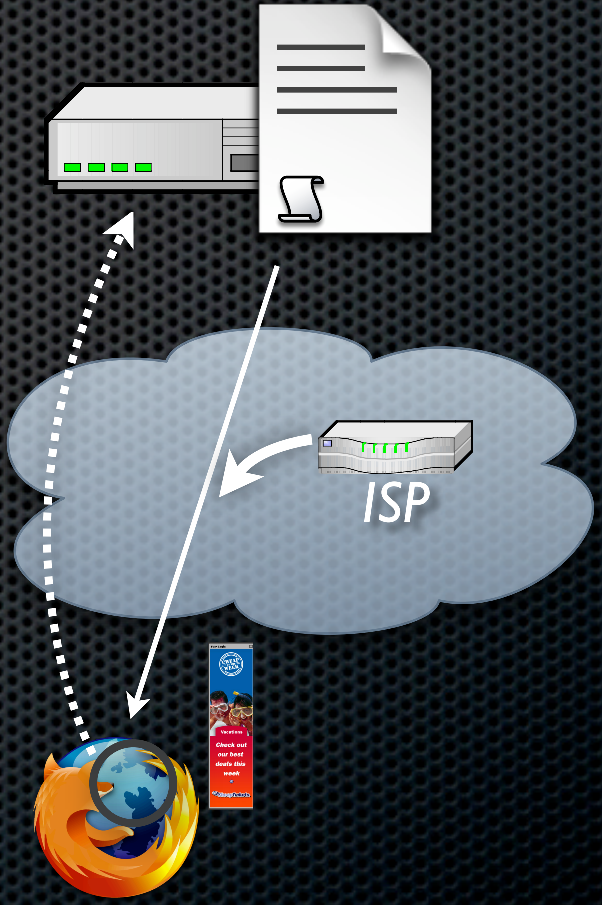


Is this a concern?

- **Measurements say it is!**
  - Of 50,000 clients, 1% saw in-flight changes (653)
  - Ads, exploits, broken pages, new vulnerabilities

Detecting Page Changes

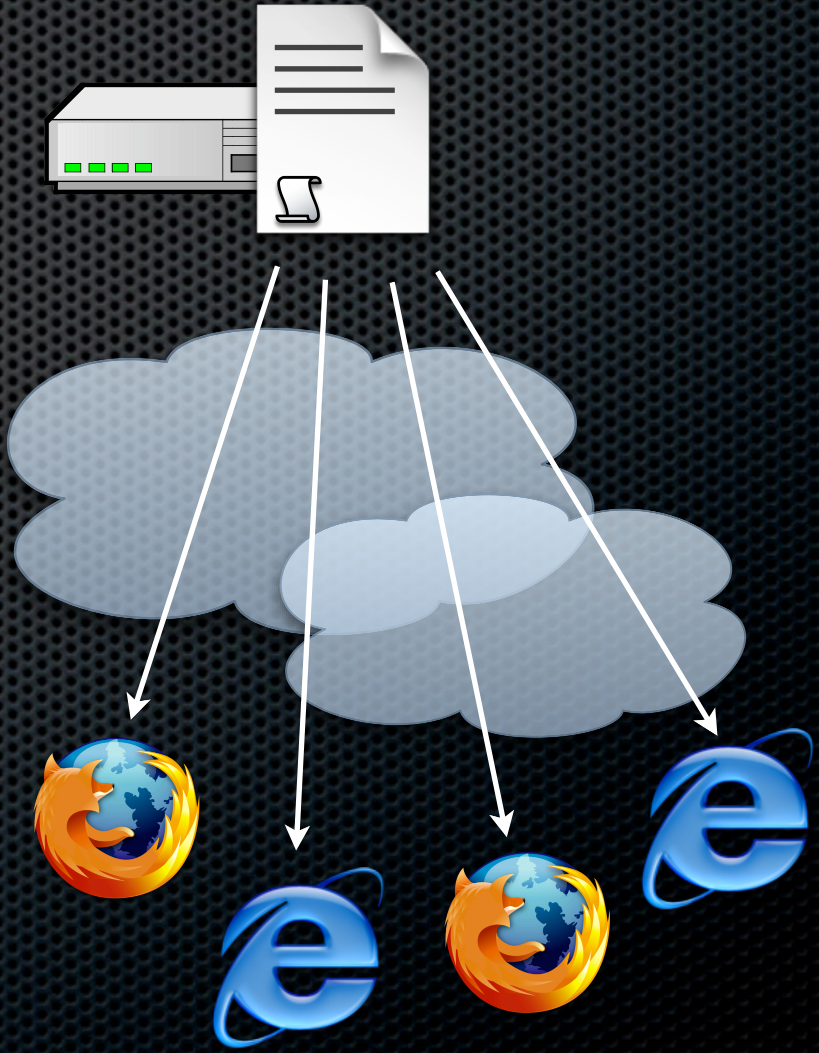
- Can detect with JavaScript
- Built a **Web Tripwire:**
  - Runs in client's browser
  - Finds most changes to HTML
  - Reports to user & server



http://vancouver.cs.washington.edu

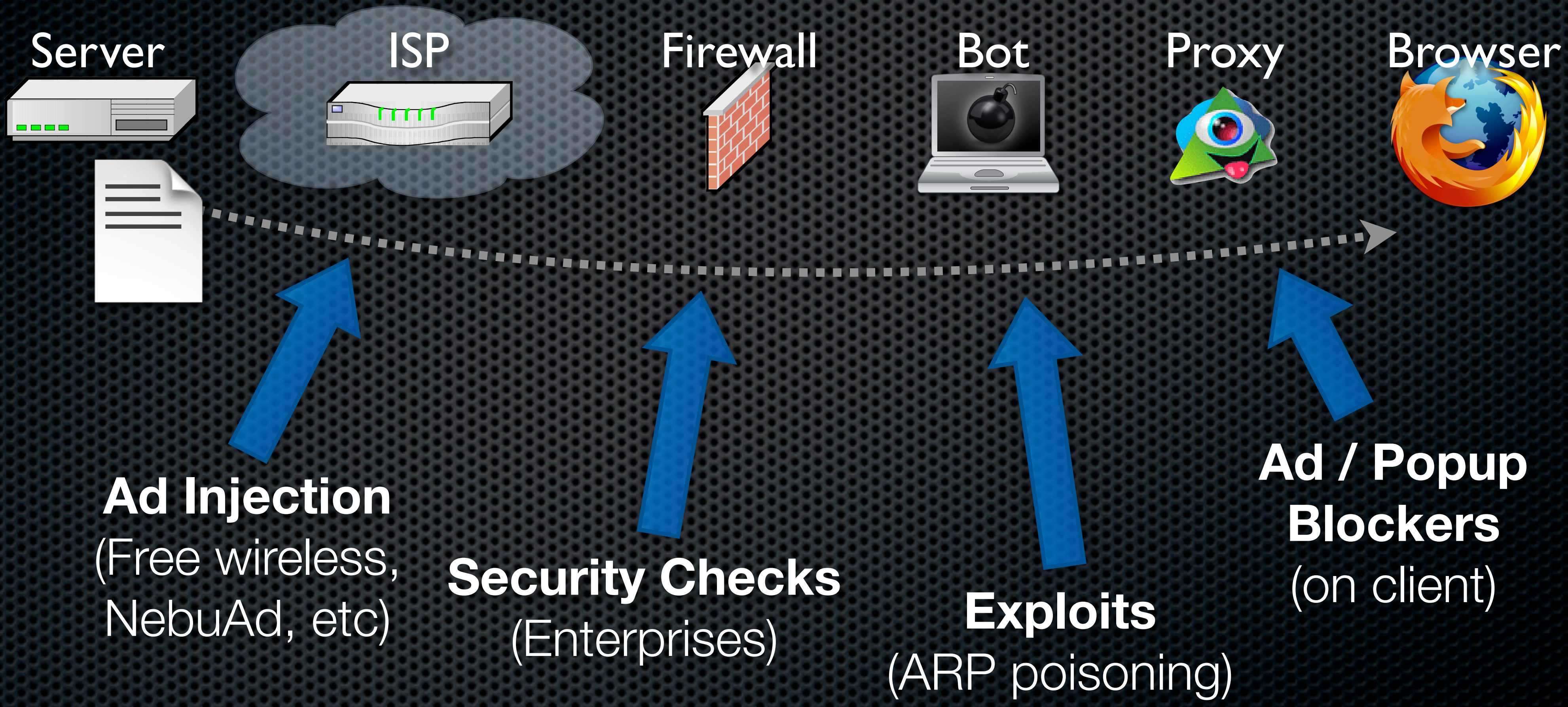
Measurement Study

- Wanted view of many clients on many networks
- Posted to **Slashdot**, **Digg**, etc.
    - Visits from over 50,000 unique IP addresses
    - 653 reported changes



http://vancouver.cs.washington.edu

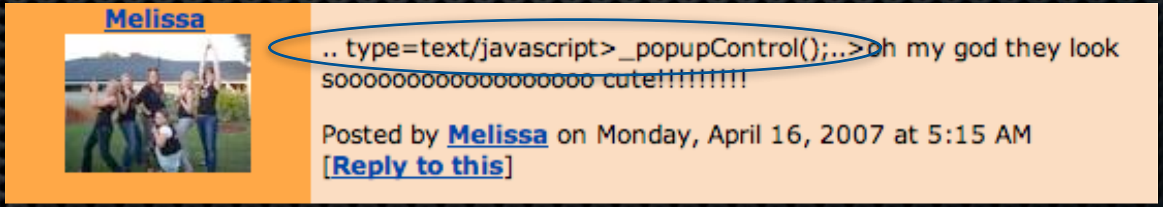
Diverse Changes Observed



http://vancouver.cs.washington.edu

The best intentions...

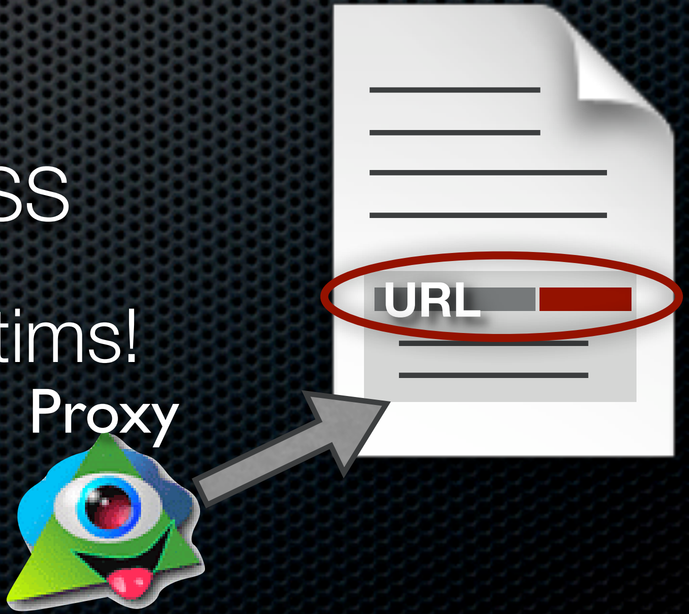
- **Bugs introduced**
  - Web forums broken by popup blockers


Melissa

.. type=text/javascript>_popupControl();..>oh my god they look soooooooooooooooooo cute!!!!!!!!!

Posted by Melissa on Monday, April 16, 2007 at 5:15 AM
[Reply to this]

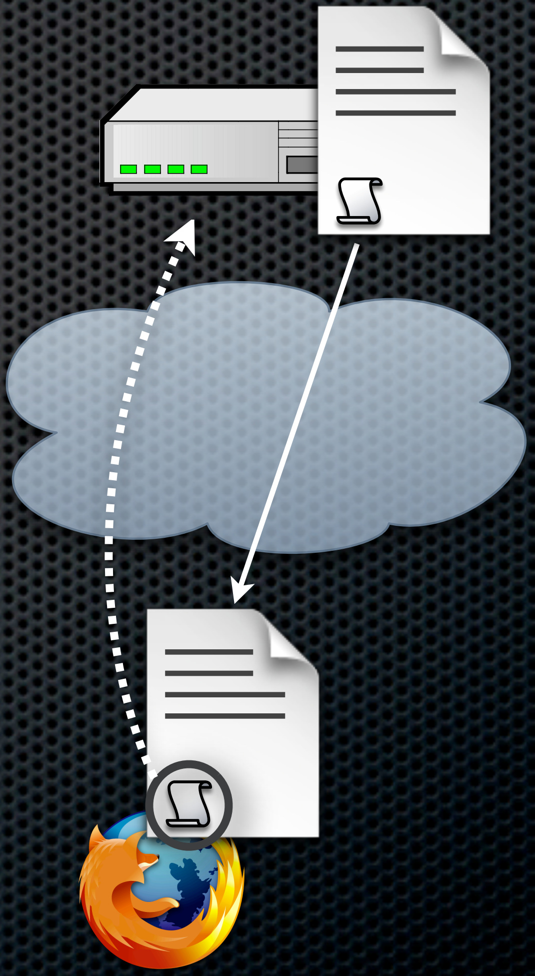
- **Vulnerabilities introduced**
  - Ad blocker code vulnerable to XSS
  - User's web programs are the victims!



http://vancouver.cs.washington.edu

Web Tripwires for Publishers

- HTTPS too costly for some sites
- Can detect changes with JavaScript
- Easy for publishers to deploy
  - **Configurable toolkit**
  - **Web tripwire service**



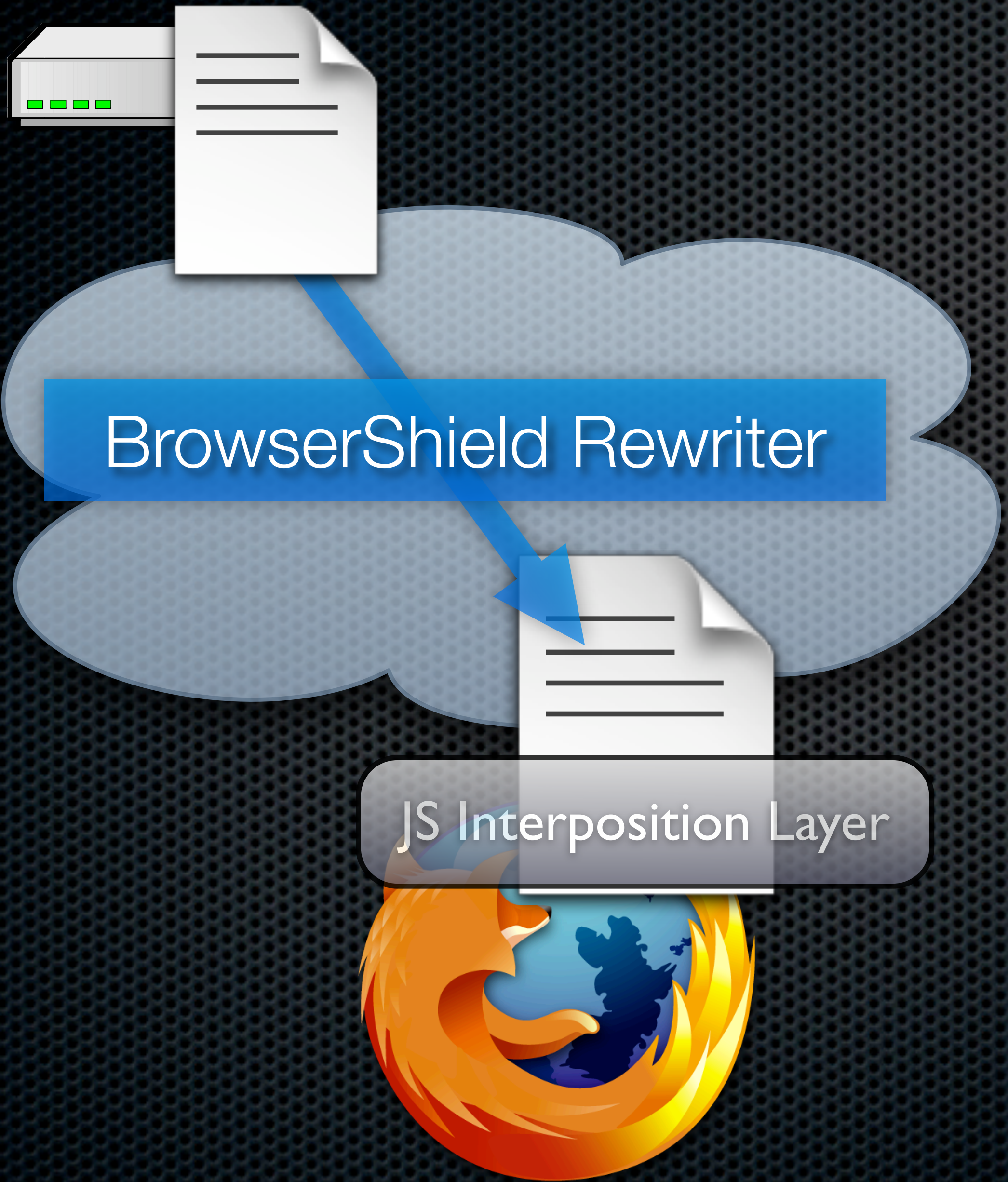
http://vancouver.cs.washington.edu

Summary

- Not safe to blindly patch code of web programs
- Many parties with incentives to do so
- Publishers can detect it with **web tripwires**

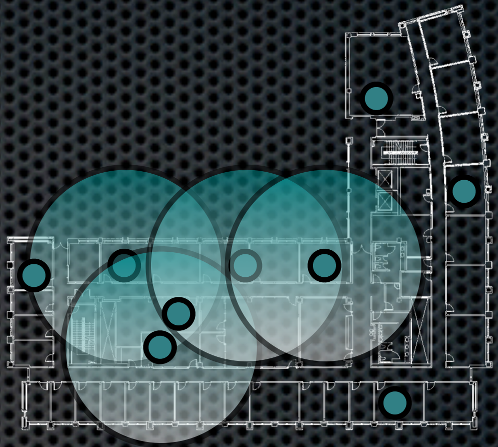
Outline

BrowserShield [OSDI ’06]



- **Block exploits** of known browser vulnerabilities
    - Interpose to enforce flexible policies
- Rewrites JavaScript code in-flight
- Has influenced Live Labs’ Web Sandbox

Earlier Research



- **Wireless Networking**
  - Study low-level 802.11 behavior [EWIND ‘05]
  - Predict behavior from measurements [SIGCOMM ’06]

- **Education with DrJava**
  - Teach production programming [SIGCSE ’03]
  - Simplify Eclipse for students [SIGCSE ’04]

Outline

Short Term Directions

- **Secure + Compatible isolation** of Site Instances
  - Better ways to evaluate compatibility
- **Opt-in mechanisms** for secure web apps
  - e.g., Alternatives to Same Origin Policy
- **Enforcing policies** on content, plug-ins, extensions

Long Term Directions

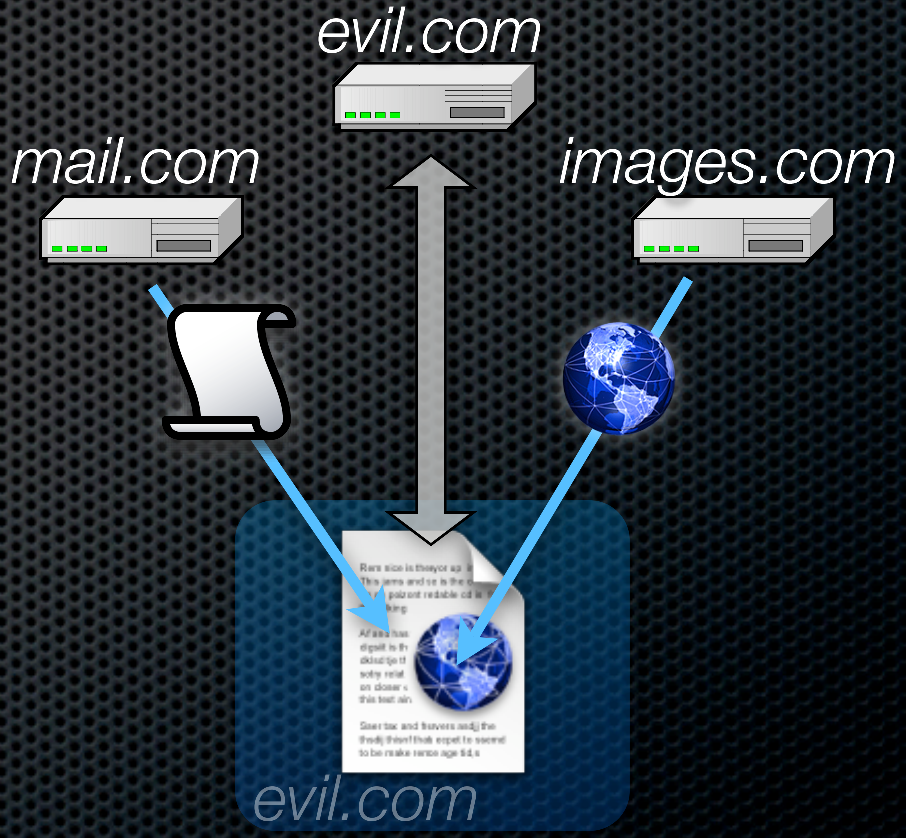
- What will **networked applications** look like?
  - How will browsers & OSes evolve to support them?
- How will **trust models** change?
  - How to grant some programs more rights?
- **Robust and secure systems** in general

Conclusion

- Web is becoming an **application platform**
    - Browser architectures must **support programs**
    - Web publishers must **protect content**
- **Great opportunity to reshape the web**

Relevant for security?

- **Pages are free to embed objects from any site**
  - Scripts, images, plugins
  - Carry user's credentials
  - *Inaccessible info within each Site Instance*
- **Compatibility makes us rely on internal logic**


evil.com
mail.com
images.com
evil.com