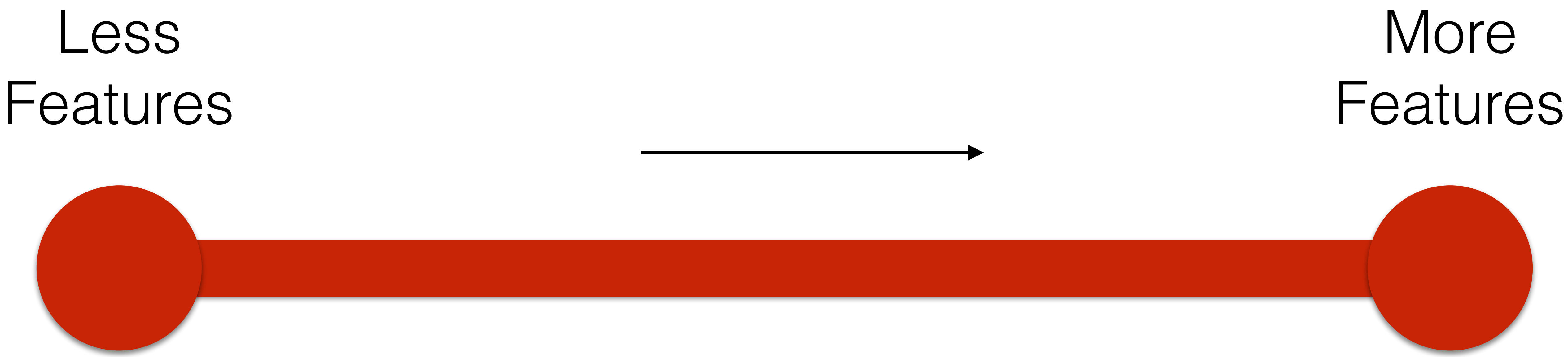
Serving Two Masters

An Empirical Study of Browser API Cooptation

Pete Snyder, Chris Kanich
University of Illinois at Chicago

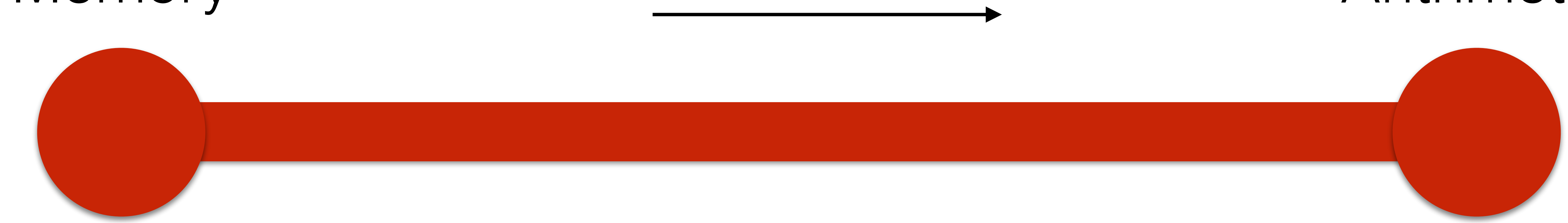


Less Features

More Features

Less Features

More Features



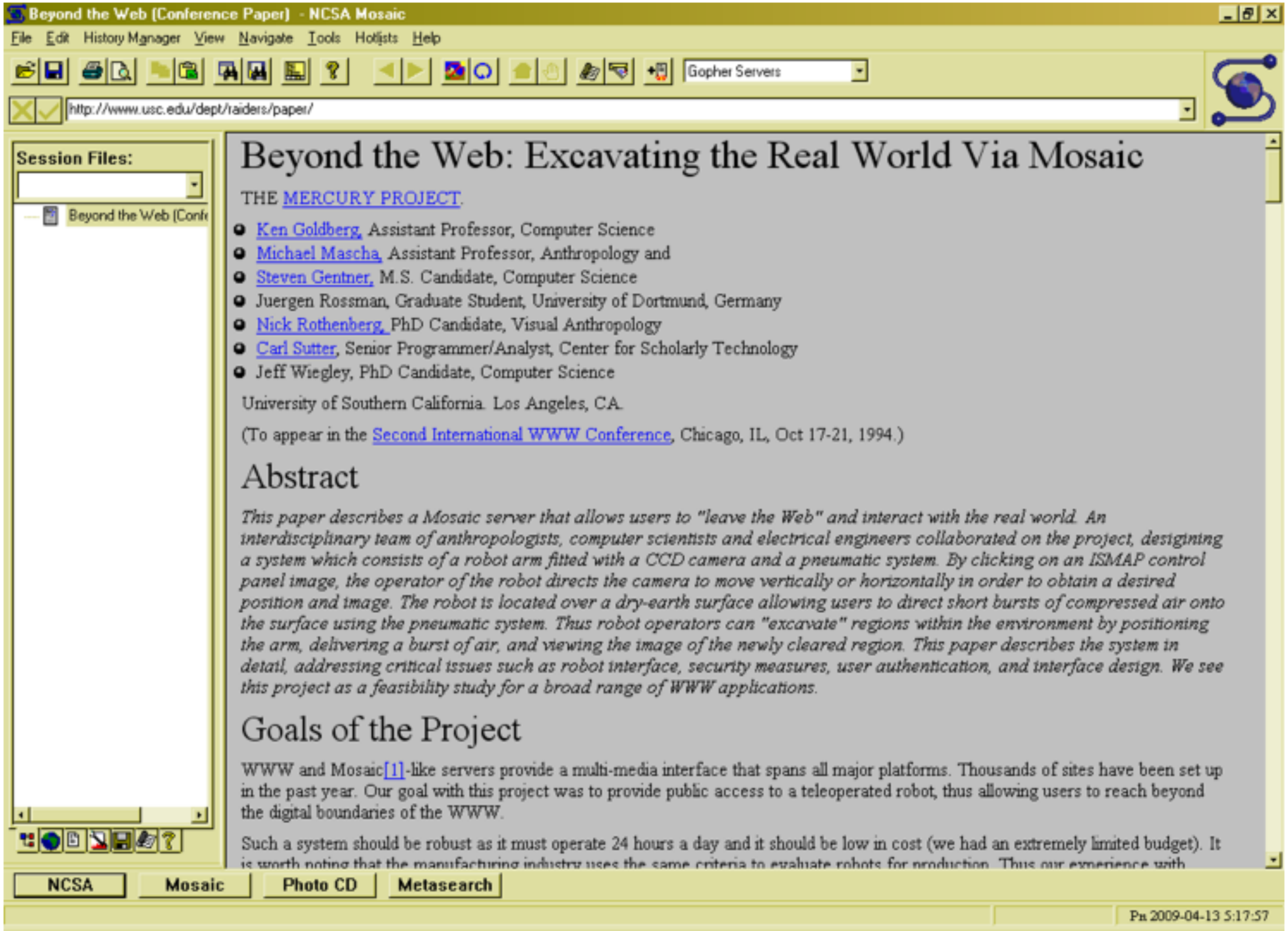
Managed Memory

Pointer Arithmetic

Outline

- Browser Complexity is Increasing
- Complexity is Often Not Useful
- Complexity is Harmful to Privacy
- Is Complexity is Harmful to Security?

1. Browser Complexity is Growing


Beyond the Web (Conference Paper) - NCSA Mosaic
File Edit History Manager View Navigate Tools Hotlists Help
Gopher Servers
http://www.usc.edu/dept/raiders/paper/
Session Files:
Beyond the Web (Confe
NCSA
Mosaic
Photo CD
Metasearch
Pn 2009-04-13 5:17:57

Beyond the Web: Excavating the Real World Via Mosaic

THE MERCURY PROJECT.

- Ken Goldberg, Assistant Professor, Computer Science
- Michael Mascha, Assistant Professor, Anthropology and
- Steven Gentner, M.S. Candidate, Computer Science
- Juergen Rossman, Graduate Student, University of Dortmund, Germany
- Nick Rothenberg, PhD Candidate, Visual Anthropology
- Carl Sutter, Senior Programmer/Analyst, Center for Scholarly Technology
- Jeff Wiegley, PhD Candidate, Computer Science

University of Southern California. Los Angeles, CA.

(To appear in the Second International WWW Conference, Chicago, IL, Oct 17-21, 1994.)

Abstract

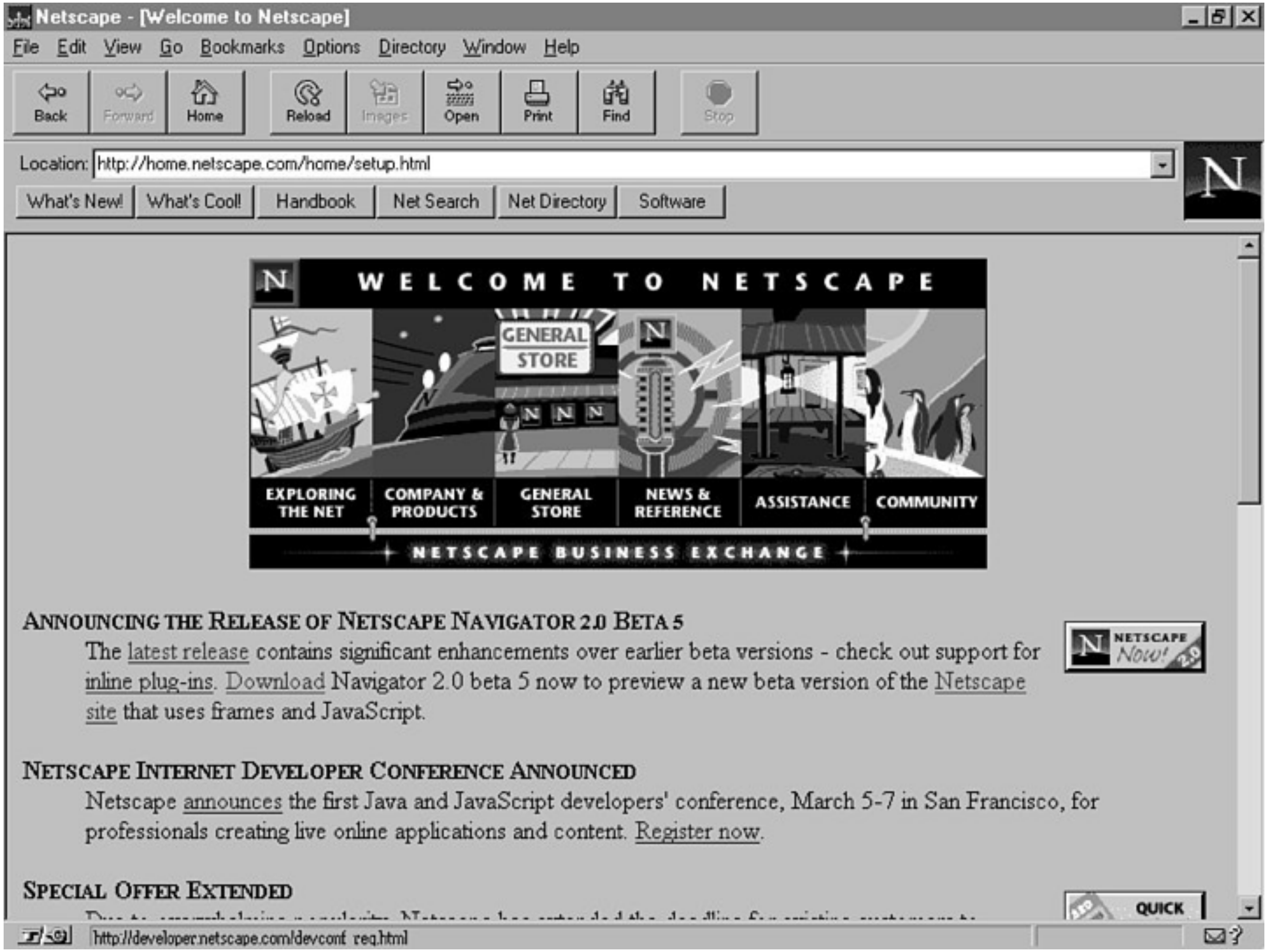
This paper describes a Mosaic server that allows users to "leave the Web" and interact with the real world. An interdisciplinary team of anthropologists, computer scientists and electrical engineers collaborated on the project, designing a system which consists of a robot arm fitted with a CCD camera and a pneumatic system. By clicking on an ISMAP control panel image, the operator of the robot directs the camera to move vertically or horizontally in order to obtain a desired position and image. The robot is located over a dry-earth surface allowing users to direct short bursts of compressed air onto the surface using the pneumatic system. Thus robot operators can "excavate" regions within the environment by positioning the arm, delivering a burst of air, and viewing the image of the newly cleared region. This paper describes the system in detail, addressing critical issues such as robot interface, security measures, user authentication, and interface design. We see this project as a feasibility study for a broad range of WWW applications.

Goals of the Project

WWW and Mosaic[1]-like servers provide a multi-media interface that spans all major platforms. Thousands of sites have been set up in the past year. Our goal with this project was to provide public access to a teleoperated robot, thus allowing users to reach beyond the digital boundaries of the WWW.

Such a system should be robust as it must operate 24 hours a day and it should be low in cost (we had an extremely limited budget). It is worth noting that the manufacturing industry uses the same criteria to evaluate robots for production. Thus our experience with

1993: Mosaic



Location: http://home.netscape.com/home/setup.html

Announcing the Release of Netscape Navigator 2.0 Beta 5

The latest release contains significant enhancements over earlier beta versions - check out support for inline plug-ins. Download Navigator 2.0 beta 5 now to preview a new beta version of the Netscape site that uses frames and JavaScript.

Netscape Internet Developer Conference Announced

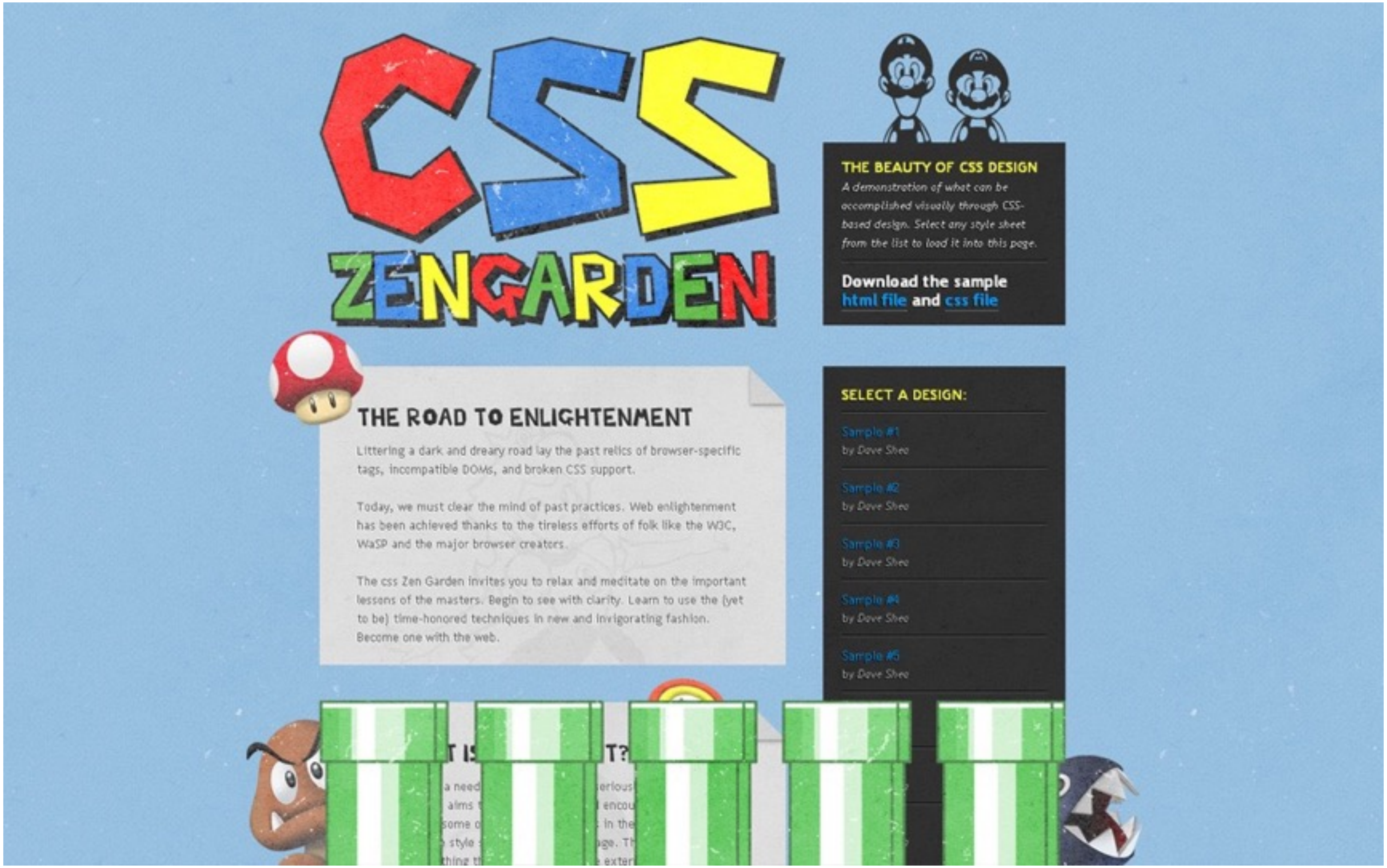
Netscape announces the first Java and JavaScript developers' conference, March 5-7 in San Francisco, for professionals creating live online applications and content. Register now.

Special Offer Extended

Due to overwhelming popularity, Netscape has extended the deadline for existing customers to

http://developer.netscape.com/devconf_req.html

1995: Netscape 2.0



THE BEAUTY OF CSS DESIGN

A demonstration of what can be accomplished visually through CSS-based design. Select any style sheet from the list to load it into this page.

Download the sample html file and css file

THE ROAD TO ENLIGHTENMENT

Littering a dark and dreary road lay the past relics of browser-specific tags, incompatible DOMs, and broken CSS support.

Today, we must clear the mind of past practices. Web enlightenment has been achieved thanks to the tireless efforts of folk like the W3C, WaSP and the major browser creators.

The css Zen Garden invites you to relax and meditate on the important lessons of the masters. Begin to see with clarity. Learn to use the (yet to be) time-honored techniques in new and invigorating fashion. Become one with the web.

SELECT A DESIGN:

Sample #1
by Dave Shea

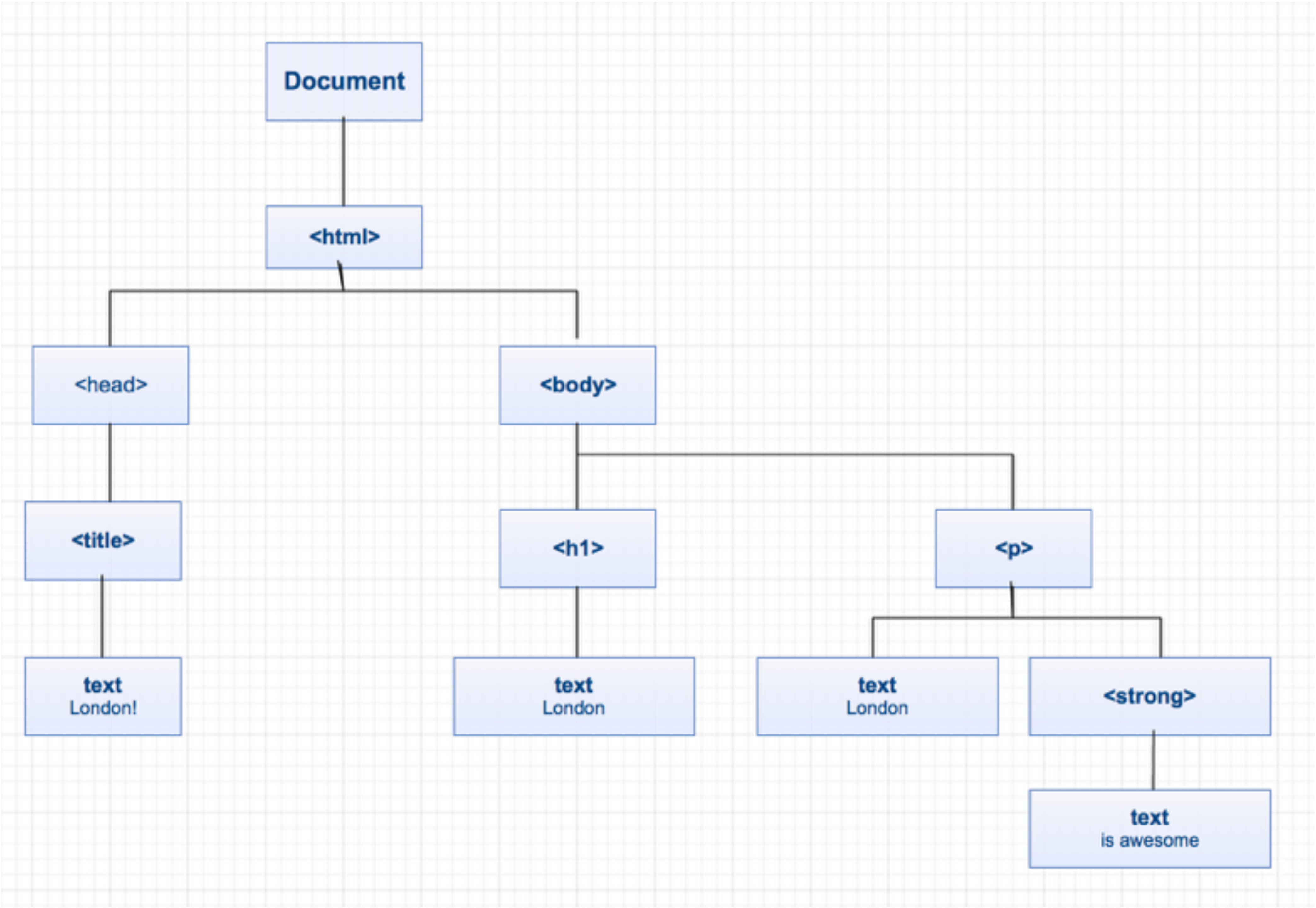
Sample #2
by Dave Shea

Sample #3
by Dave Shea

Sample #4
by Dave Shea

Sample #5
by Dave Shea

1996: CSS

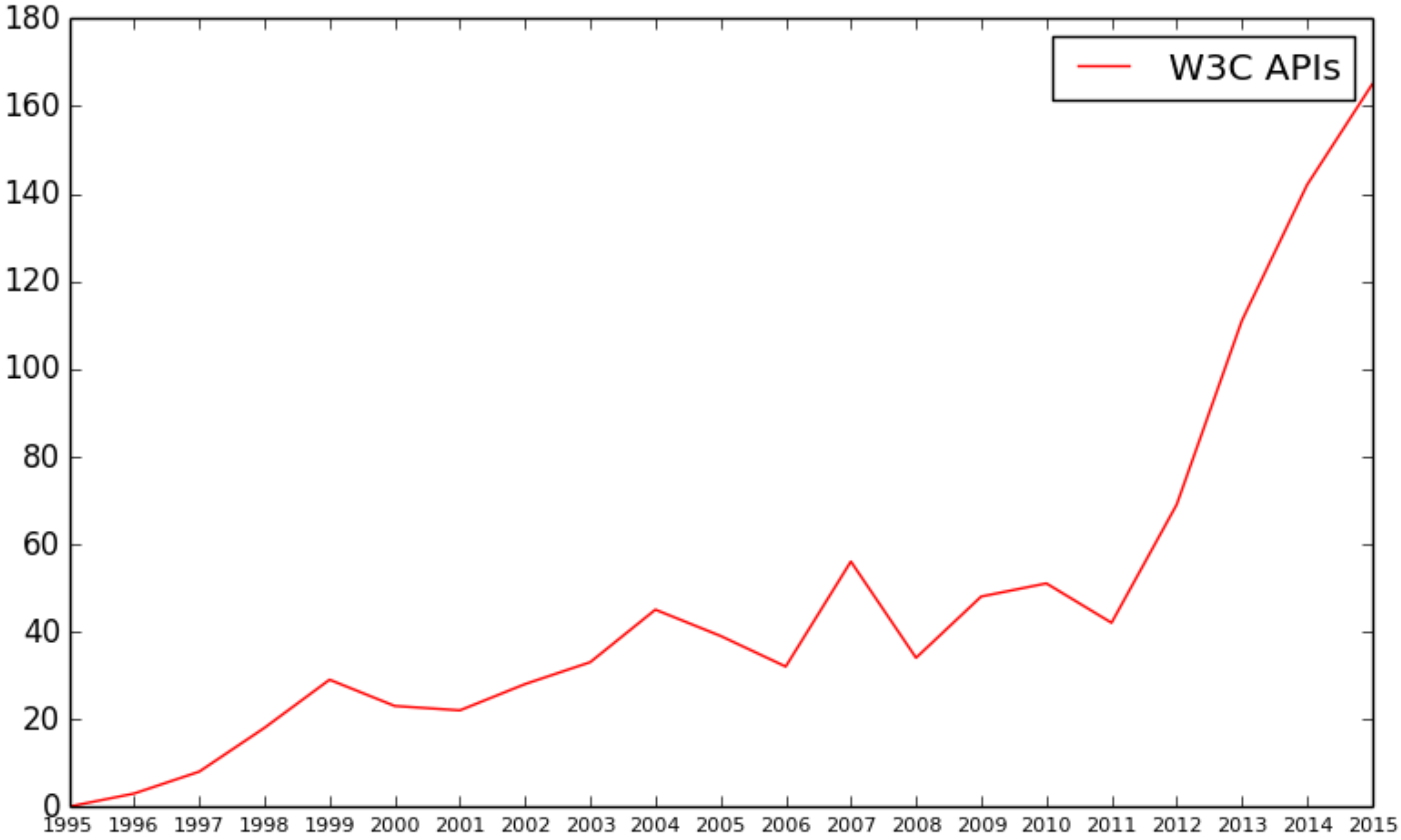

Document
<html>
<head>
<body>
<title>
<h1>
<p>
text
London!
text
London
text
London
<strong>
text
is awesome

1998: DOM1

1999: AJAX / XMLHttpRequest

Observations

- API growth started off very slow
- API growth was “document” centric
- “Broad” APIs



180
160
140
120
100
80
60
40
20
0
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
W3C APIs

API Growth

2013

- CSSOM View Module
- Web Audio API
- Proximity Events
- Crypto Extensions
- Touch Events
- GeoLocation API
- Pointer API
- CSS Animations

2014

- Calendar API
- Messaging API
- RDF Extensions
- Progress events
- Network Info API
- Ambient Light API
- HTML 5
- WebCrypto API

2015

- Encrypted Media Extensions
- Web MIDI
- Service Workers
- Performance API
- Raw Socket API
- WebDriver API
- SVG 2 API
- WebRTC

2. Is This Complexity Useful?

Determining API “Usefulness”

- Measure how often APIs are called
- Decide whether those calls are “useful"
- Simulate real world web browsing

Measuring API Calls

- Selected 45 APIs and features
- Instrumented PhantomJS / WebKit
- Implemented missing APIs

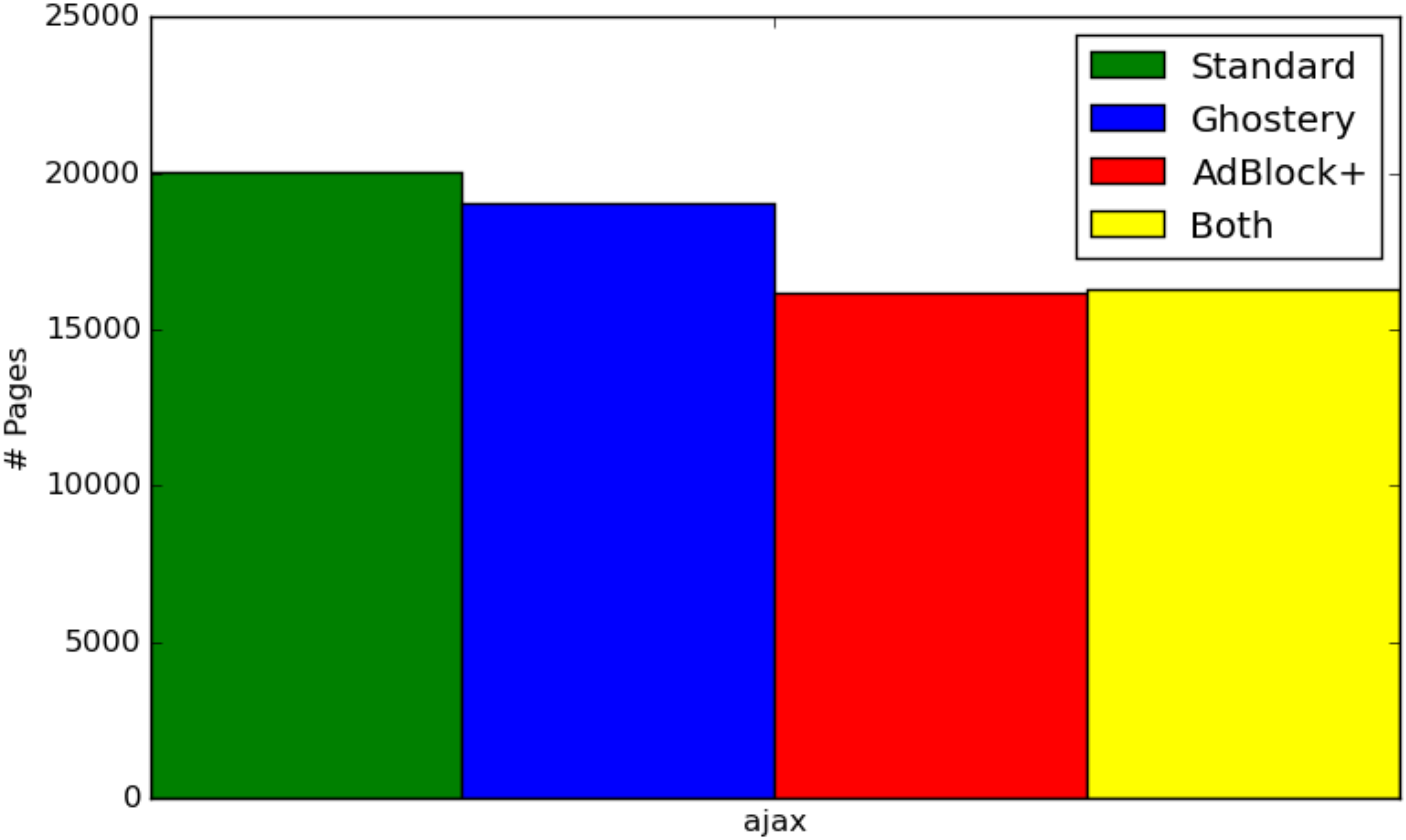
“Usefulness” Oracle

- Subjective measure
- Ghostery and AdBlock+ filter rules
- Measure API usage pre-and-post filters

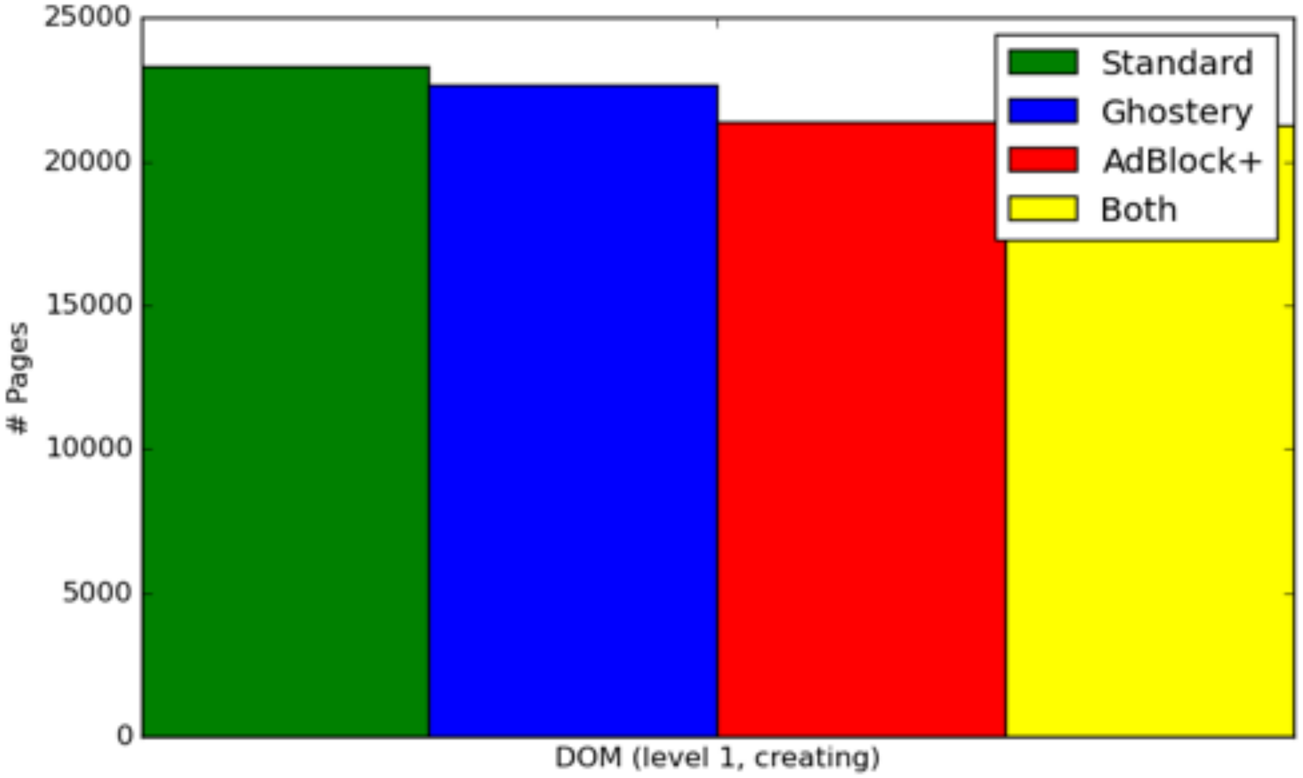
Simulated Browsing

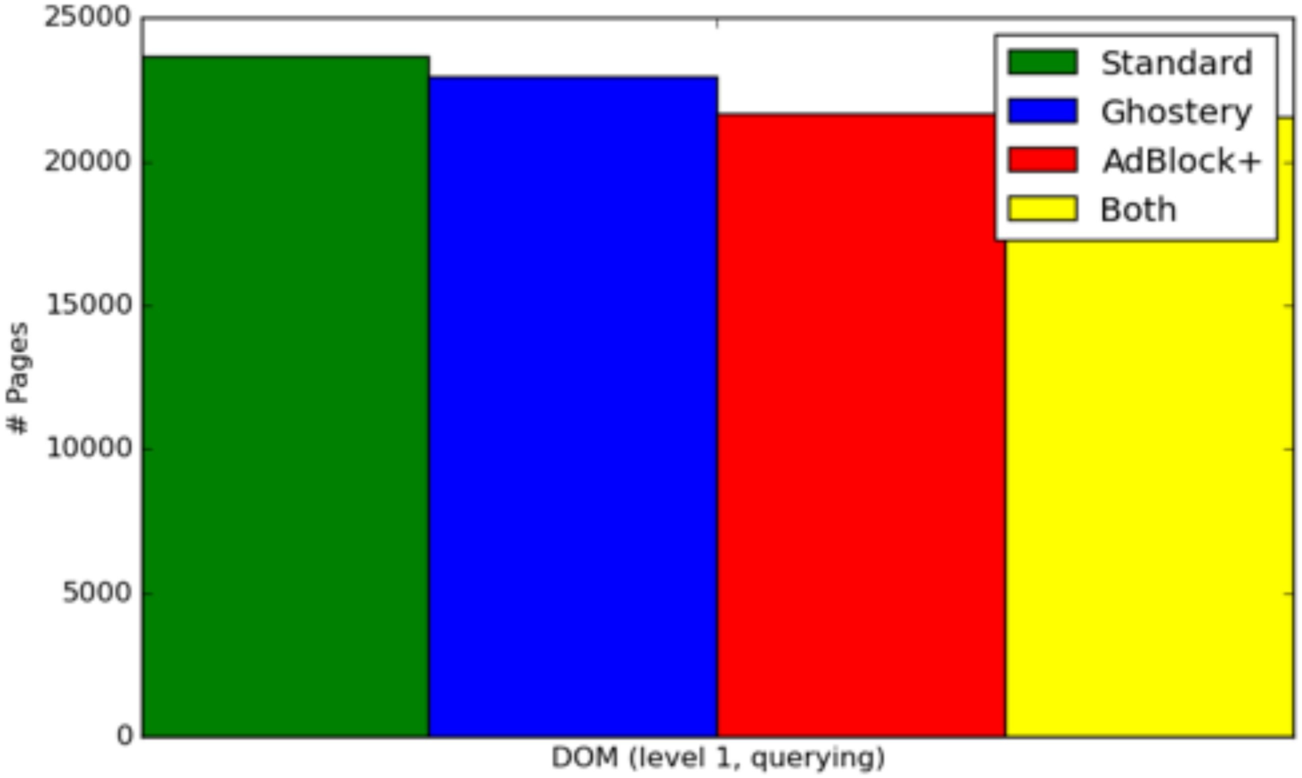
DuckDuckGo

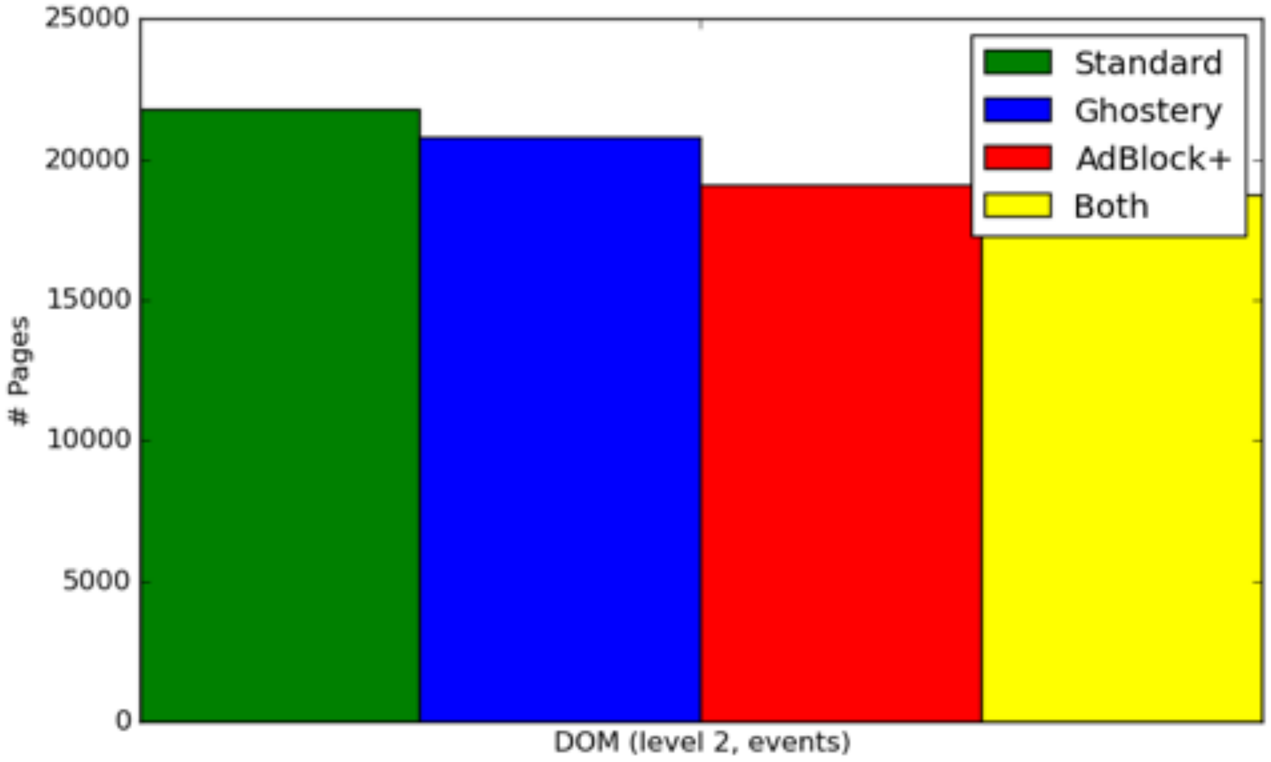
- Alexa 10,000
- 10,000 random URLs
- 10,000 random Hosts
- “Random” sites taken from searching UNIX dictionary tri-grams on DDG


25000
20000
15000
10000
5000
0
# Pages
ajax
Standard
Ghostery
AdBlock+
Both

AJAX


25000
20000
15000
10000
5000
0
# Pages
🟩 Standard
🟦 Ghostery
🟥 AdBlock+
🟨 Both
DOM (level 1, creating)


25000
20000
15000
10000
5000
0
# Pages
🟩 Standard
🟦 Ghostery
🟥 AdBlock+
🟨 Both
DOM (level 1, querying)


25000
20000
15000
10000
5000
0
# Pages
🟩 Standard
🟦 Ghostery
🟥 AdBlock+
🟨 Both
DOM (level 2, events)

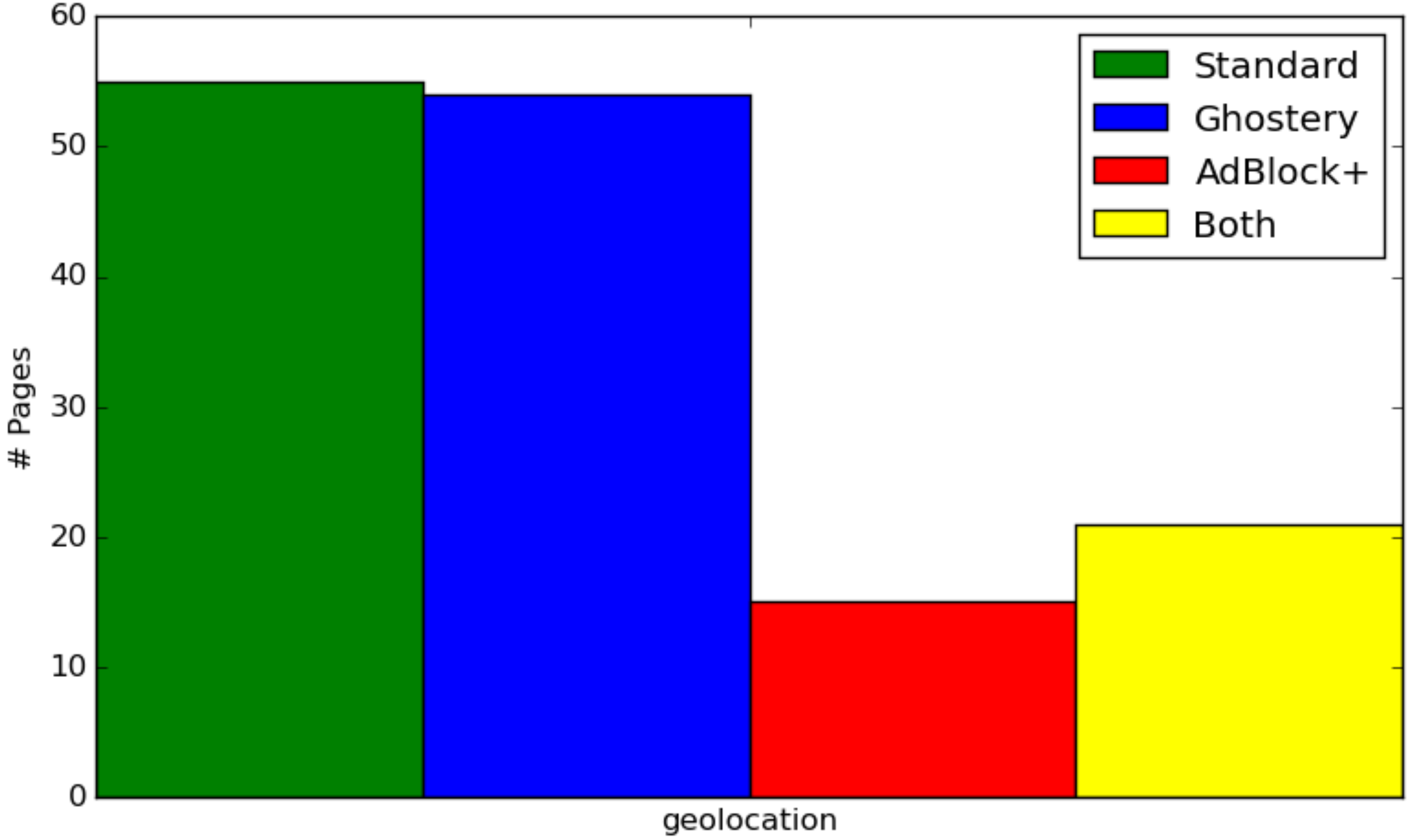
DOM 1 + 2 APIs

Rare APIs

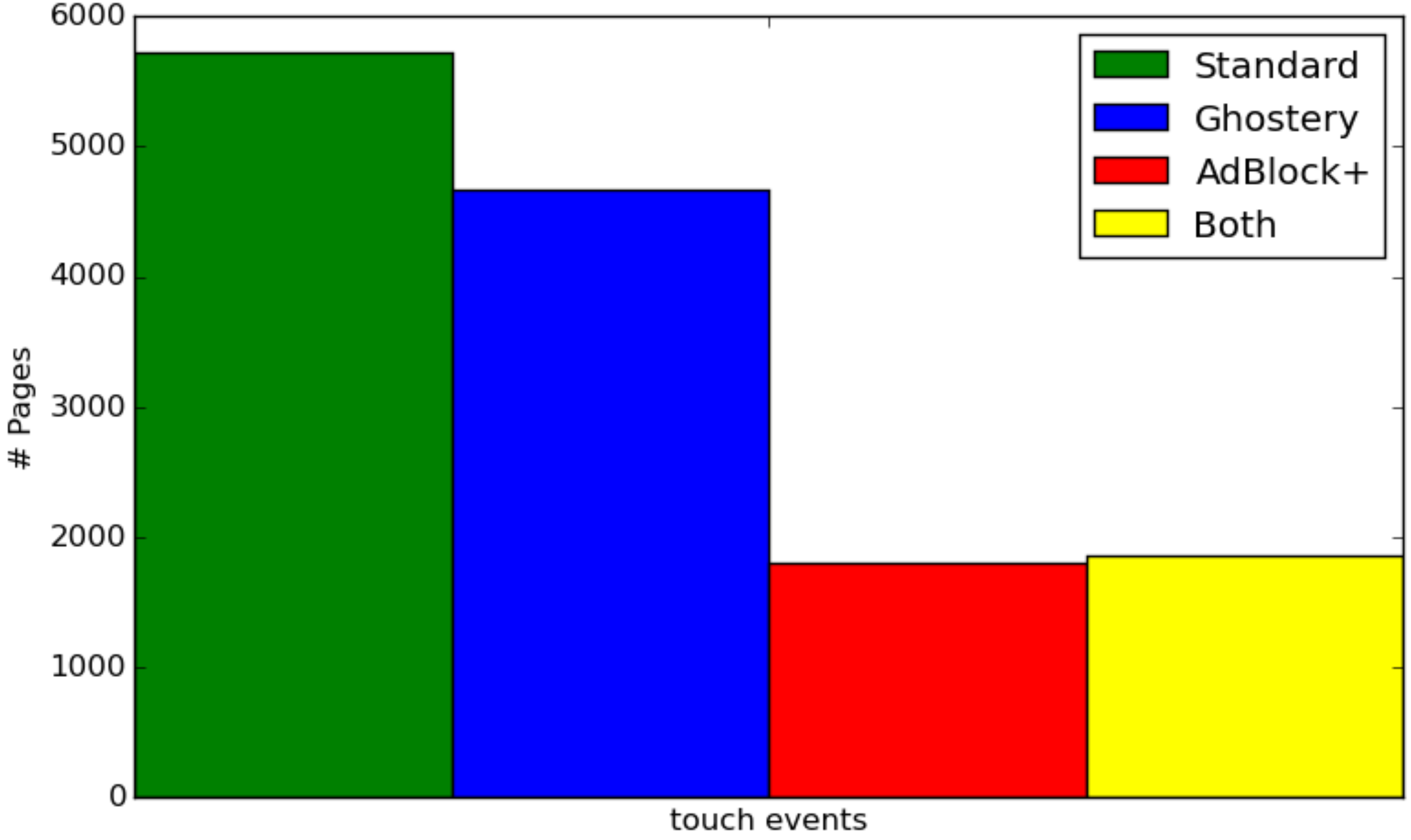
API Name	URLs
Battery API	21
Page Transition API	9
GeoLocation API	55
Shadow DOM	5

Non-used APIs

- IndexDB
- WebGL
- WebRTC
- Browser Name API
- Gamepad API
- SVG API
- Vibration API
- WebAudio API
- WebWorker API



GeoLocation API



Touch Events API

3. Browser Complexity is Harmful to Privacy

Example: WebRTC

- Intent: Allow peer-to-peer applications
- Attack: Leaks local IP address
- Widely available (56.22%)
- Rarely used for intended purpose

Browser	Version Since
Firefox	22
Chrome	23
Android Browser	40
Opera	30

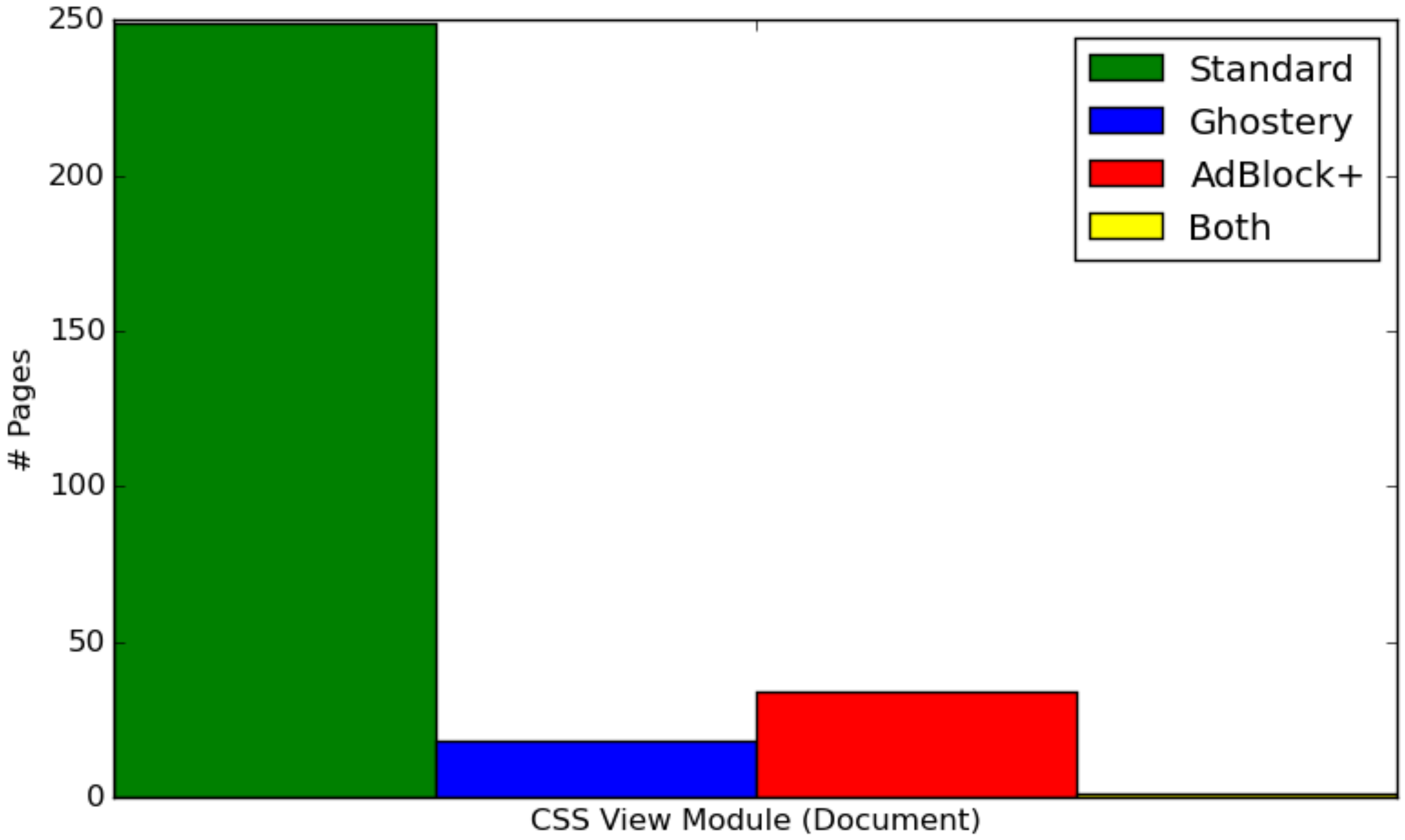
Example: Crypto

- Intent: Allow applications to perform crypto operations
- Use: Generates persistant random identifiers
- Widely available (70.24%)
- Rarely used for intended purpose

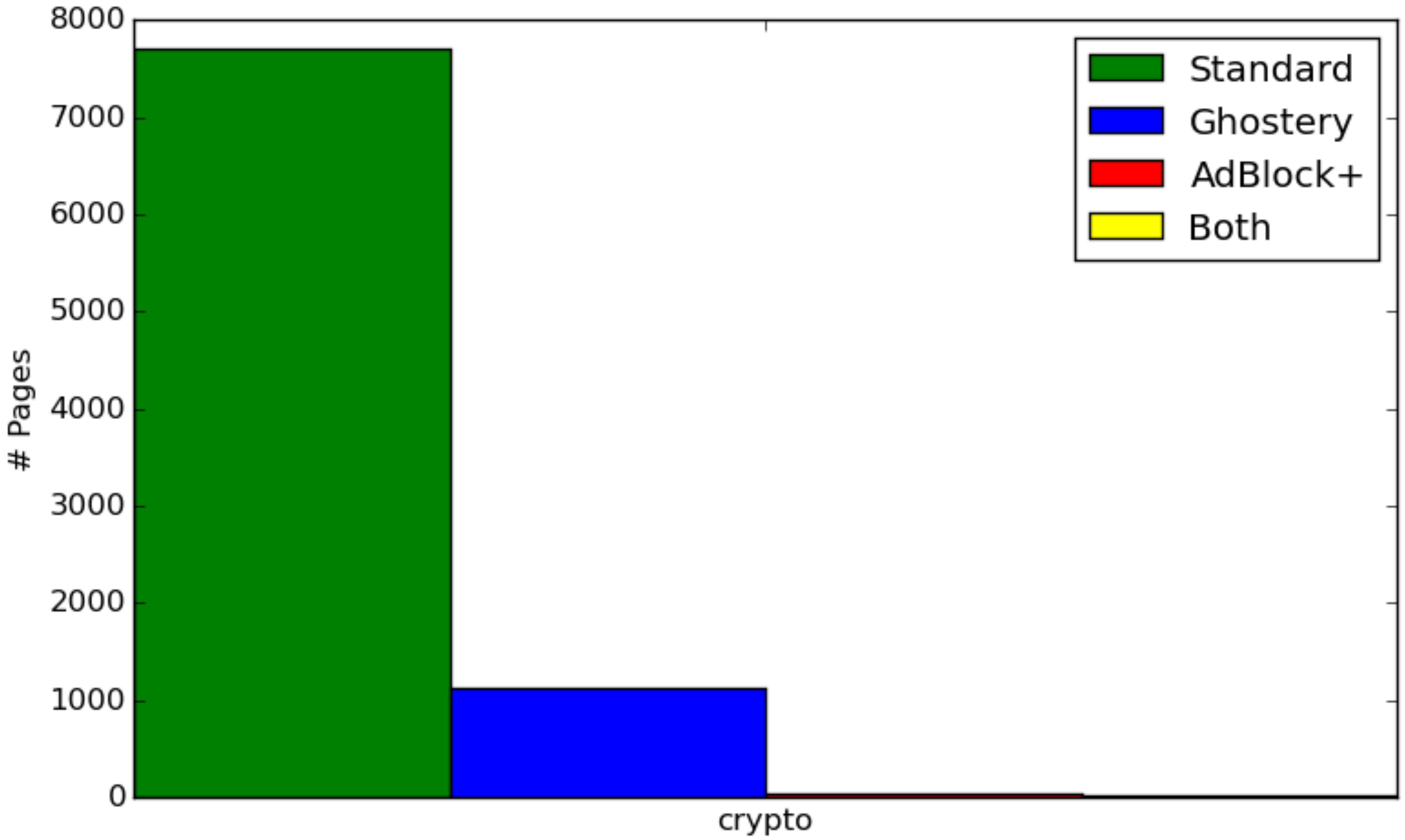
Browser	Version Since
Firefox	38
Chrome	31
Android Browser	4.4
Opera	30
IE	11
iOS	7.1

Methodology

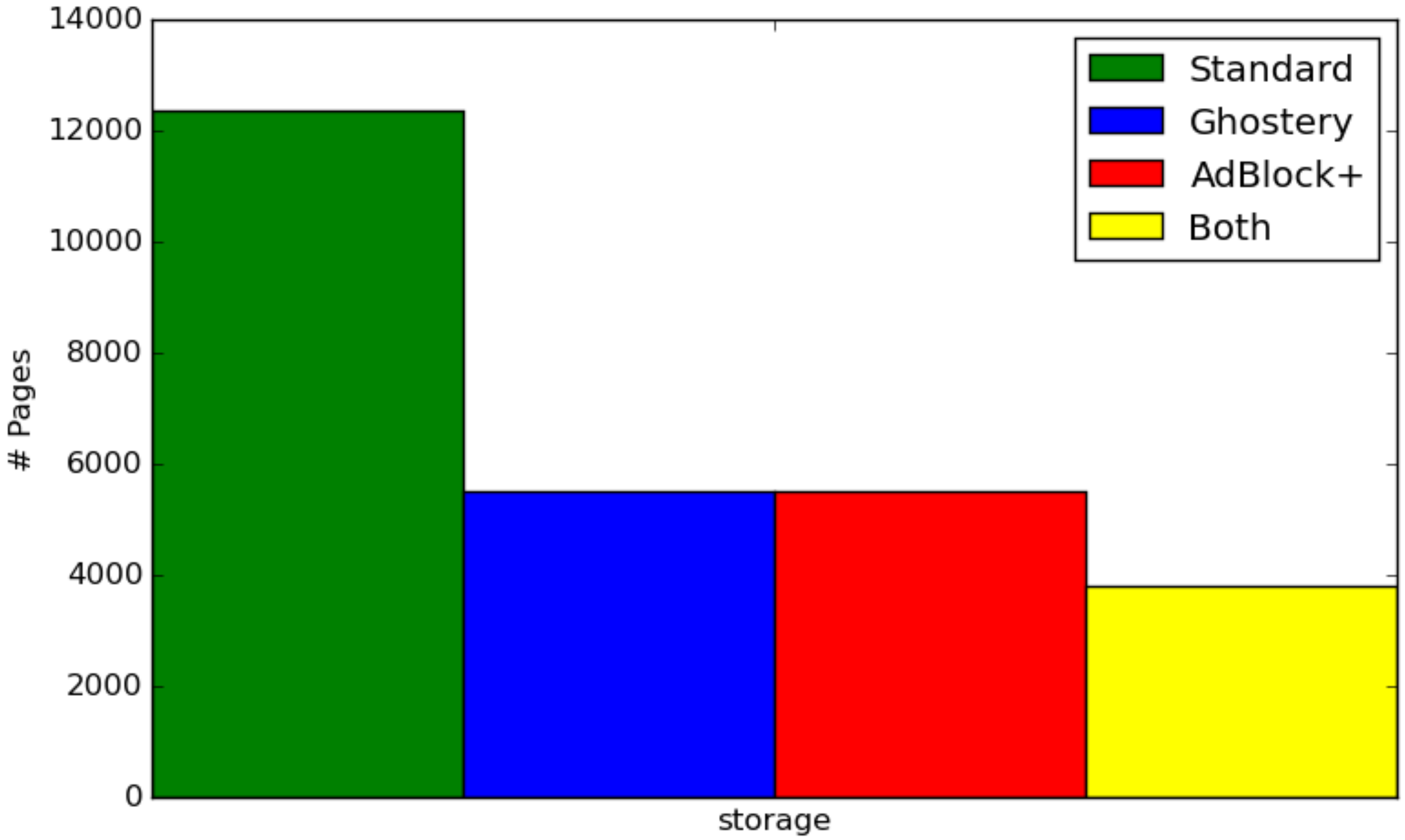
- Load and measure each URL
- Reload and remeasure with Ghostery
- Big differences in API usage -> privacy-harmful APIs



CSSOM API (Document)


8000
7000
6000
5000
4000
3000
2000
1000
0
# Pages
crypto
Standard
Ghostery
AdBlock+
Both

Crypto API



14000
12000
10000
8000
6000
4000
2000
0
# Pages
storage
Standard
Ghostery
AdBlock+
Both

Storage API

“Non-User Serving” APIs

API	Pages #	Ghost #	Ghost %	ABP #	ABP %	Both #	Both %
CSSOM (Doc)	249	18	92.8	34	86.3	1	99.6
Crypto	7,713	1,123	85.4	38	99.5	27	99.6
Language	16,909	2,242	86.7	2,072	87.7	1,131	93.3
<iframe> Injection	12,110	3,202	73.6	4,464	63.1	1,351	88.8
Page Visibility	729	228	68.7	81	88.9	86	88.2
Websocket	225	99	56.0	58	74.2	43	80.9
Plugin Detection	18,116	5,870	67.6	4,133	77.2	3,512	80.6
Battery API	21	17	19.0	4	81.0	6	71.4
Storage	12,357	5,499	55.5	5,496	55.5	3,817	69.1

“User Serving” APIs

API	Pages #	Ghost #	Ghost %	ABP #	ABP %	Both #	Both %
DOM 1 (creating)	23,304	22,651	2.8	21,409	8.1	21,266	8.7
DOM 1 (querying)	23,659	22,965	2.9	21,705	8.3	21,580	8.8
AJAX	20,016	19,027	4.9	16,153	19.3	16,303	18.6
Canvas API	2,095	1,949	7.0	1,676	20.0	1,694	19.1
User Agent	23,439	21,195	9.6	19,602	16.4	18,870	19.5
<audio>	307	292	4.9	247	19.5	242	21.2
Blob API	308	287	6.8	233	24.4	238	22.7
<svg>	860	798	7.2	520	39.5	527	38.7
History API	576	490	14.9	374	35.1	349	39.4

4. Is Complexity is Harmful to Security?

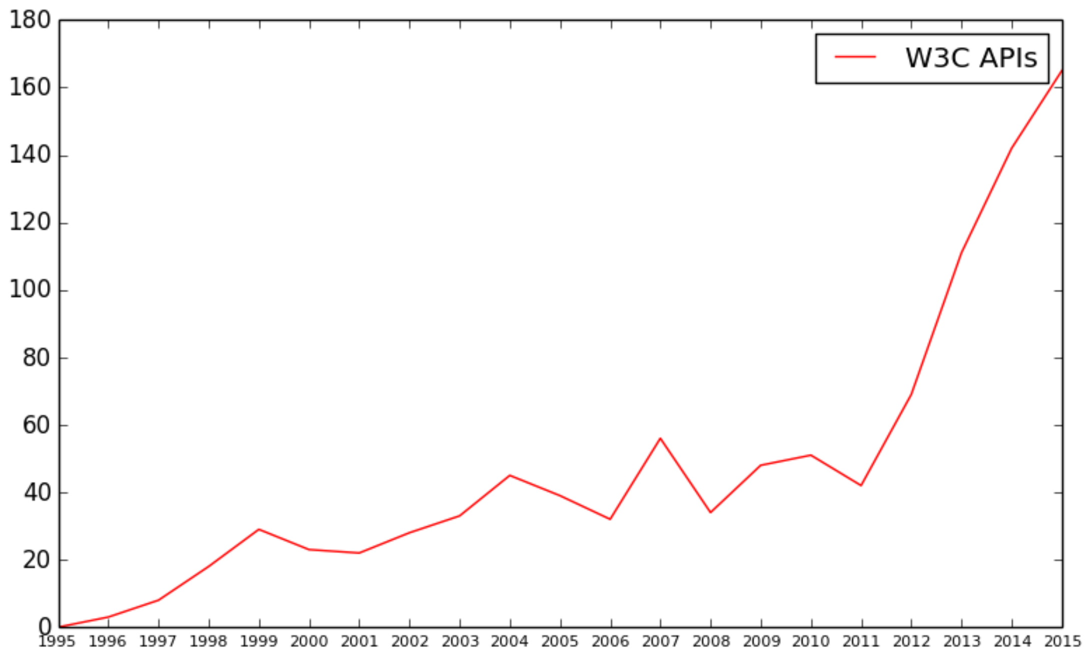
@todo

- Status quo violates “principle of least privilege”
- Gathering data from open bug databases
- Lots of hand labeling involved…
- On going…

5. Conclusions

Conclusions

- Browsers are growing in complexity quickly
- Mismatch between user intent and web author intent
- Mismatch between need and capability
- Harms privacy, might harm security



Thanks!