The Web Browser Is a Transitional Technology

Talk, by Allen Wirfs-Brock

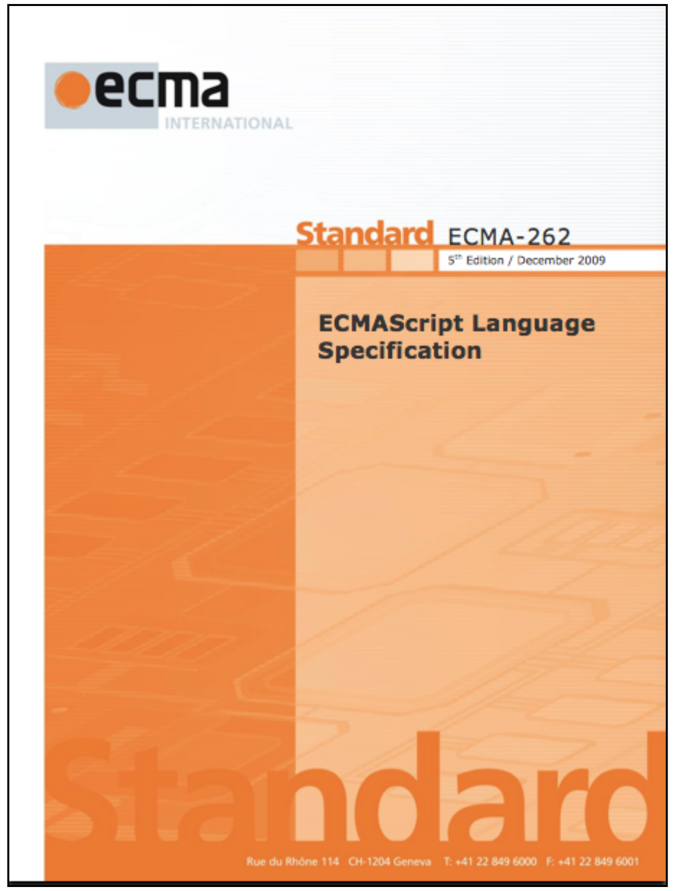
YOW! 2011, December, 2011

Today, many projects are increasingly focused on web based applications using web browsers as their primary user interface. How durable is this application architecture? Is the browser likely to continue to expand its primacy? Can we expect the basic structure of our web based applications to remain fairly stable for the foreseeable future or do we need to be preparing to make drastic changes? If the browser is a transitional technology, what might replace it? This talk will explore these and related issues and help you understand how application development is about to change.

A little background about me

- Compilers, Smalltalk virtual machines, GCs, language design, development tools
- Launched first commercial Smalltalk: Tektronix 4404
- Digitalk/Parcplace-Digitalk: Enterprise Scale Smalltalk
- (Re-) Instantiations: JOVE Java optimizing compiler, Eclipse tools
- Microsoft: JavaScript/ECMAScript 5
- Mozilla: Future of JavaScript and the Web platform

http://youtu.be/8yxCJfayW-8



ecma INTERNATIONAL

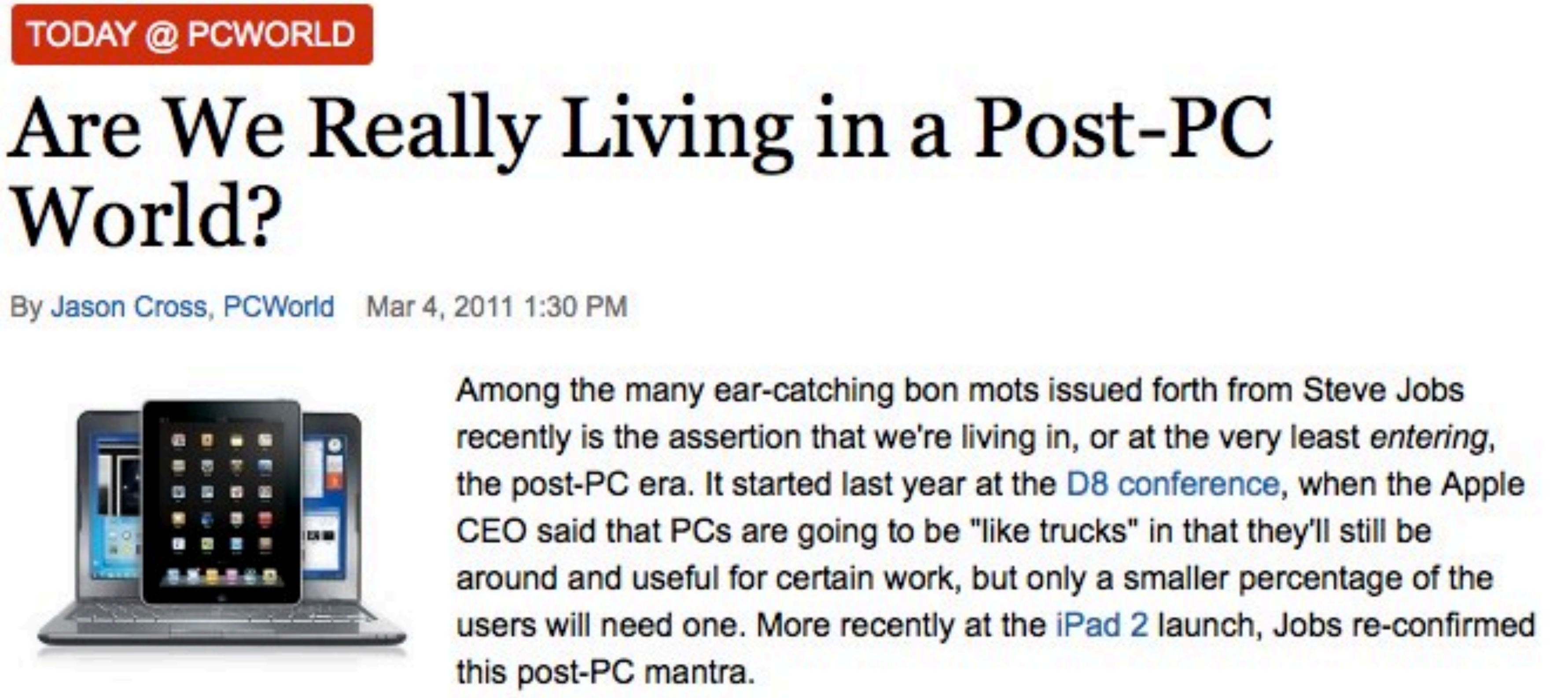
Standard ECMA-262

5th Edition / December 2009

ECMAScript Language Specification

Standard

Rue du Rhône 114 CH-1204 Geneva T: +41 22 849 6000 F: +41 22 849 6001



TODAY @ PCWORLD

Are We Really Living in a Post-PC World?

By Jason Cross, PCWorld Mar 4, 2011 1:30 PM

Among the many ear-catching bon mots issued forth from Steve Jobs recently is the assertion that we're living in, or at the very least *entering*, the post-PC era. It started last year at the D8 conference, when the Apple CEO said that PCs are going to be "like trucks" in that they'll still be around and useful for certain work, but only a smaller percentage of the users will need one. More recently at the iPad 2 launch, Jobs re-confirmed this post-PC mantra.

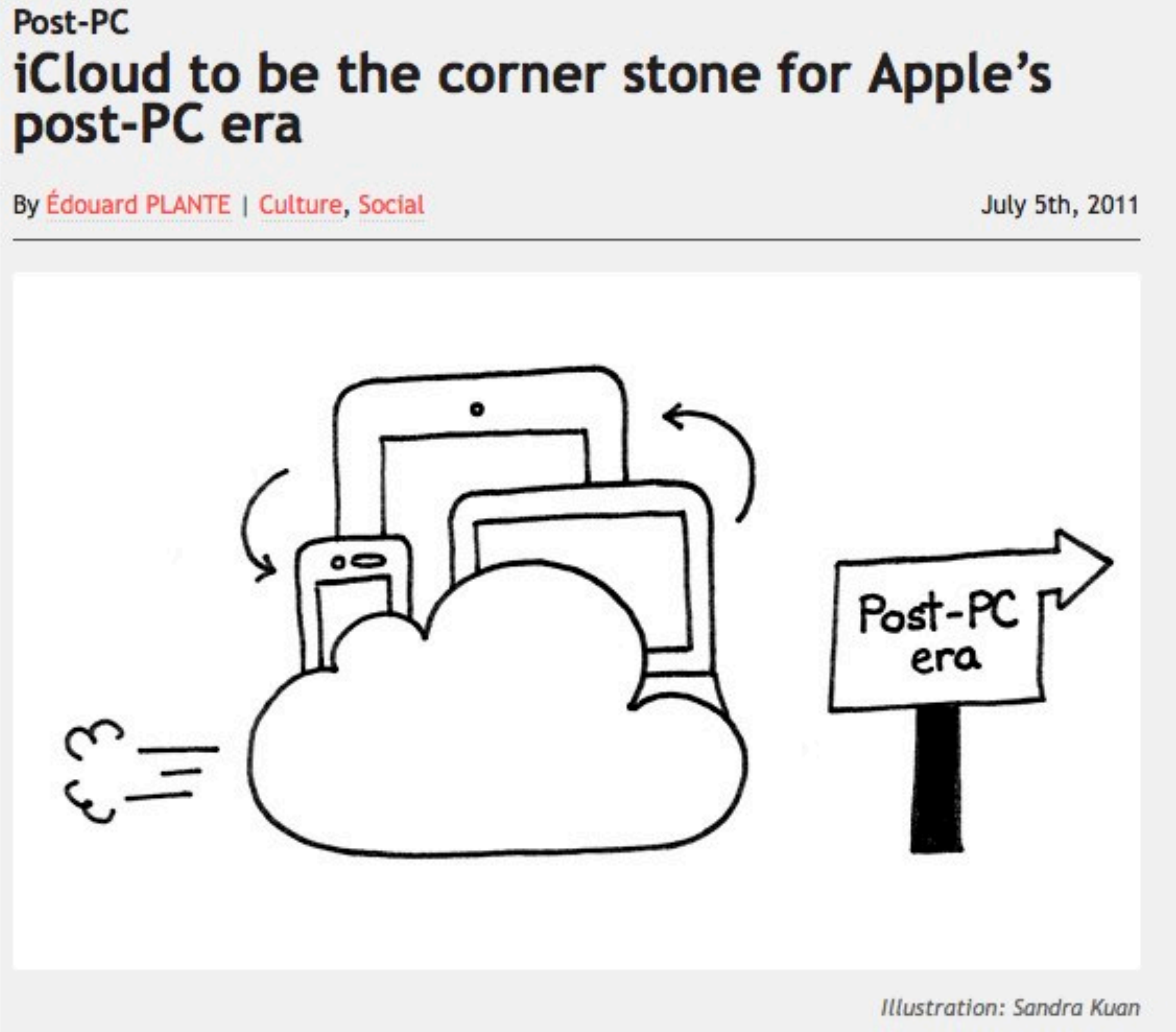
FILED UNDER: BUSINESS TECH

VMware boss focuses on post-PC era at VMworld

By: Jay Greene

AUGUST 29, 2011 6:41 PM PDT

Print E-mail



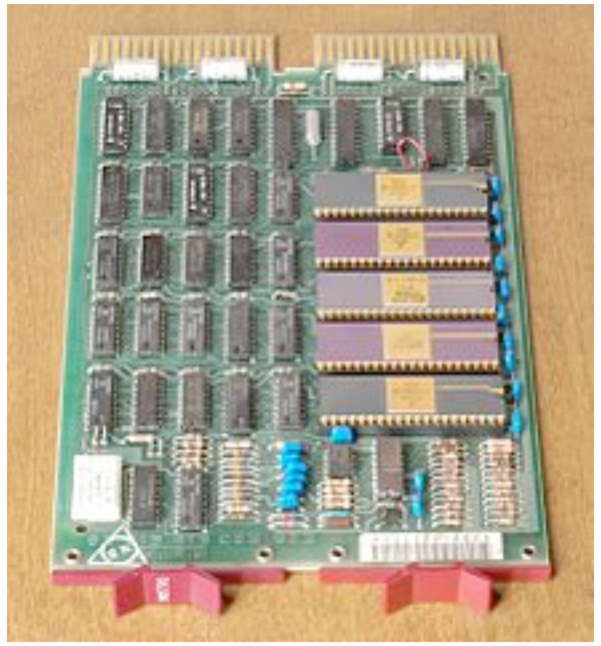
Post-PC

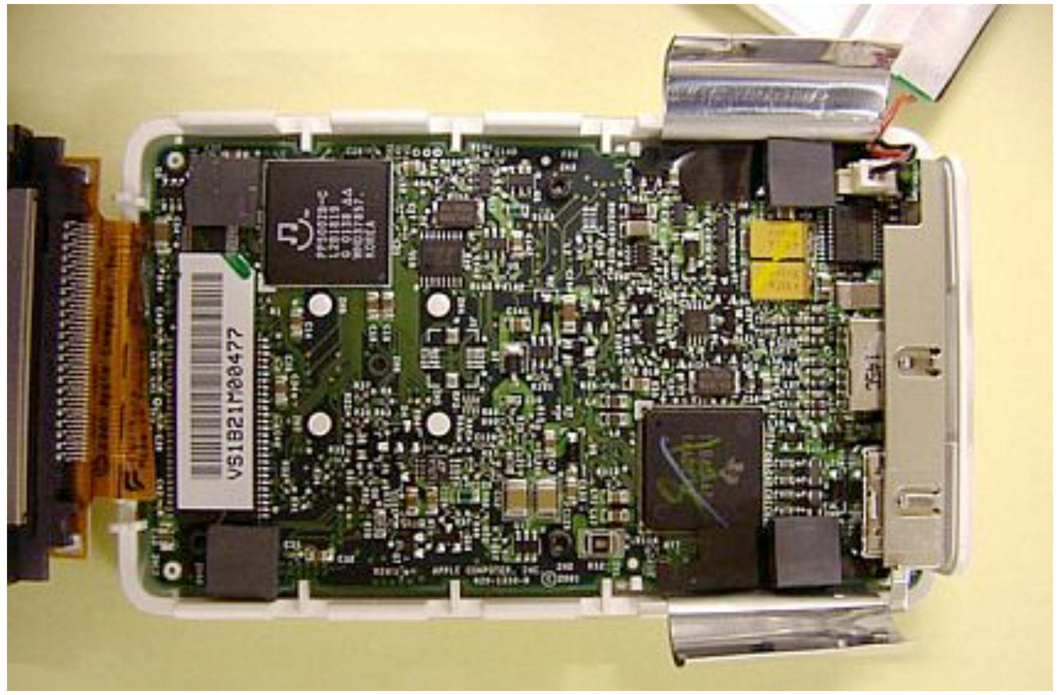
iCloud to be the corner stone for Apple's post-PC era

By Édouard PLANTE | Culture, Social July 5th, 2011

Illustration: Sandra Kuan

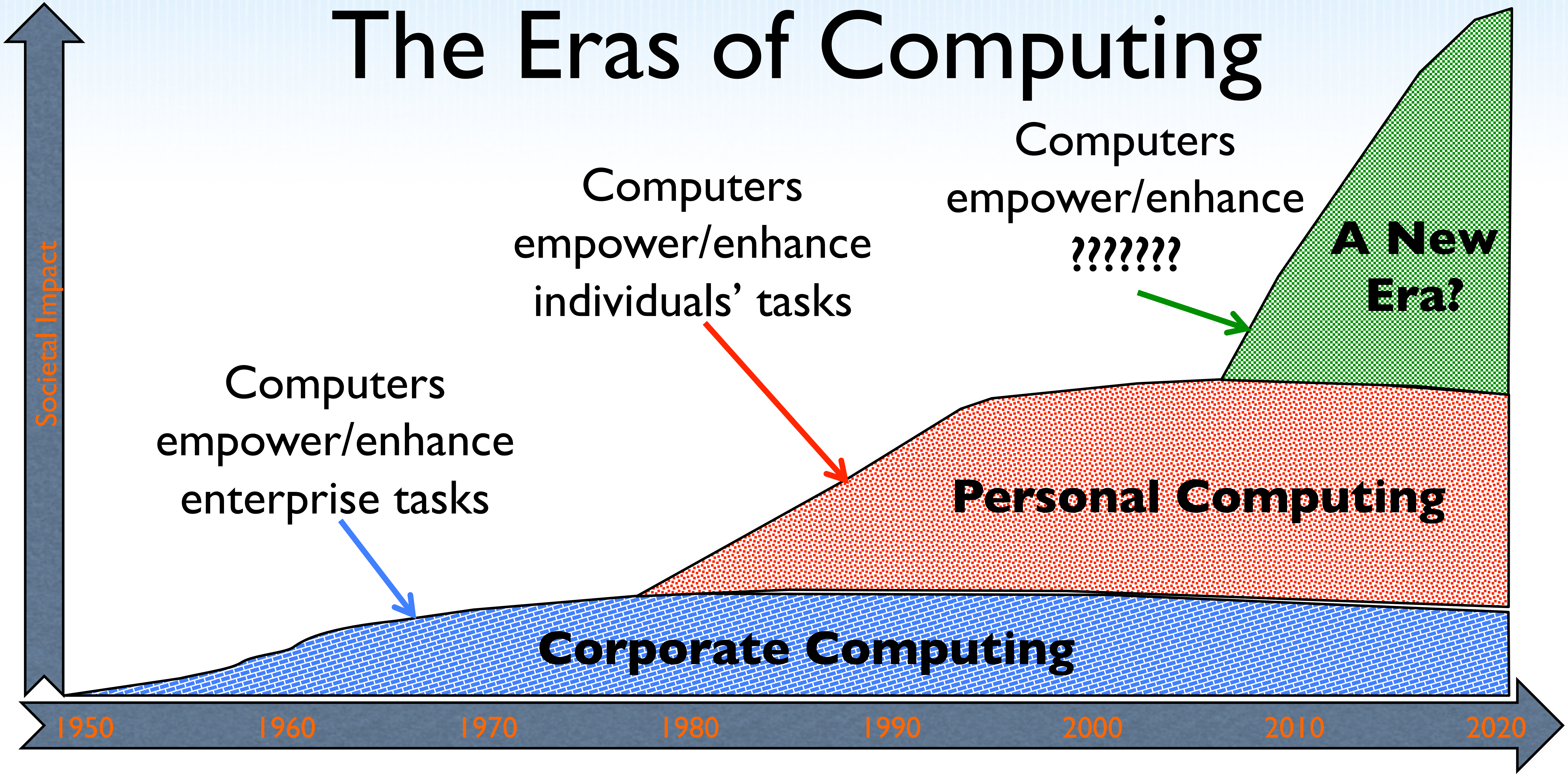
Hardware evolution doesn't define a computing era





Computing eras are about the impact of computing on society

The Eras of Computing



mozilla

If we are entering a new era of computing, what is it?

- The Post-PC Era?
- The Mobile Era?
- The Cloud Era?
- The Tablet Era?
- The Games and Media Player Era?
- The Monetize the User Era?

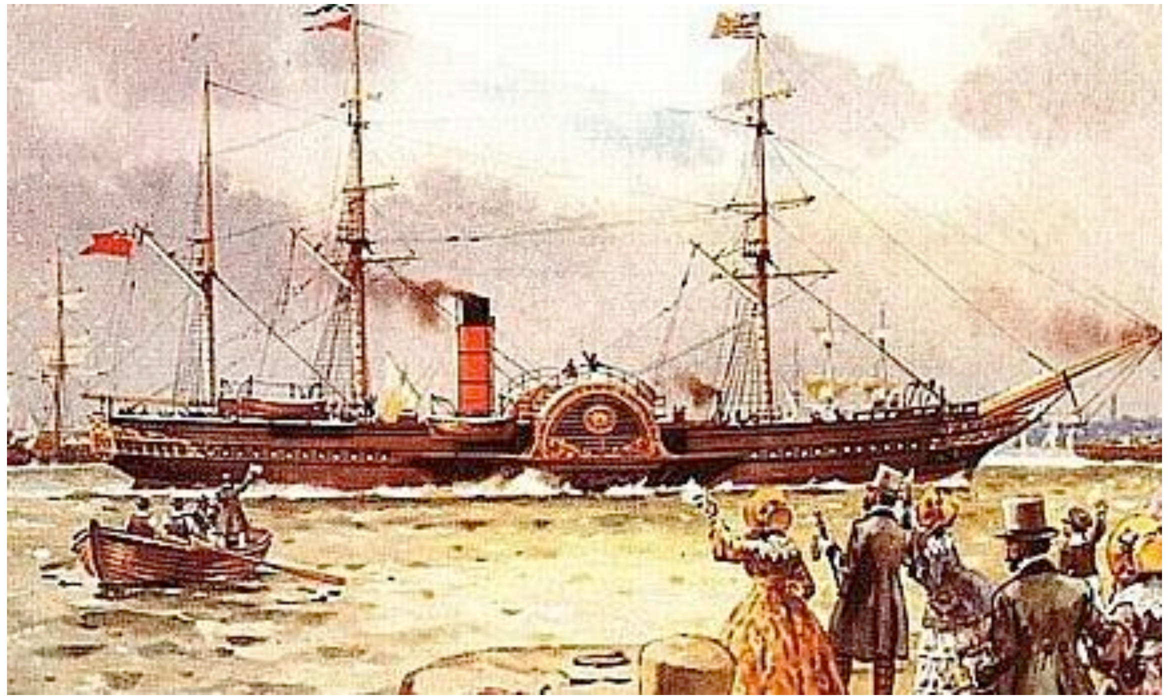
The Ambient Computing Era

- Devices not Computers
- Ubiquitous access to information
- Computing augmented life

Computers enhance the world I live in. I need my stuff (data and apps) right now, wherever I am, using whatever device is available. I can't live without it!

Transitional Technologies

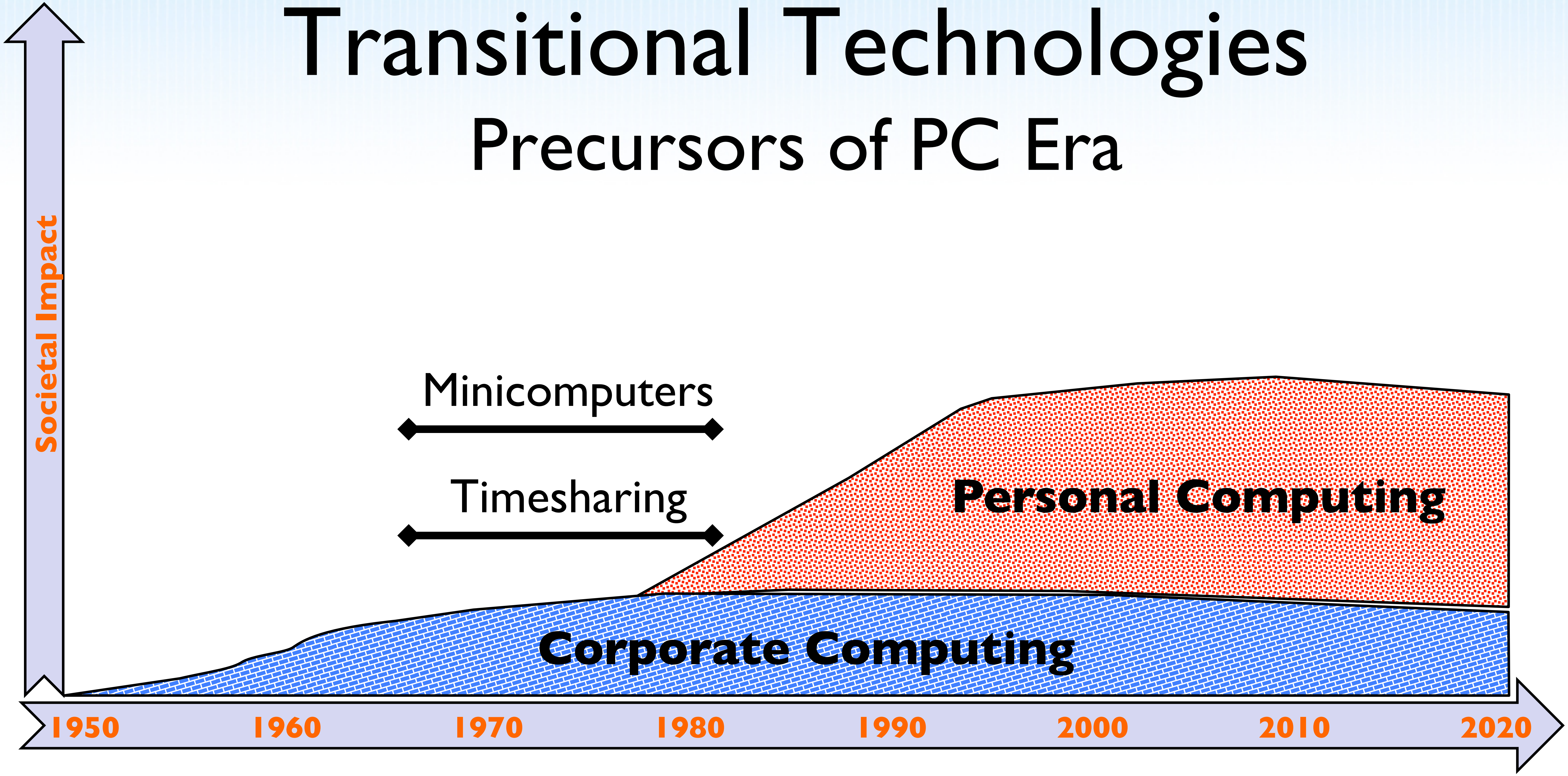
- Precursors of fundamental change
- Exhibit some characteristics of what will come
- Firmly rooted in the status quo



mozilla

Transitional Technologies

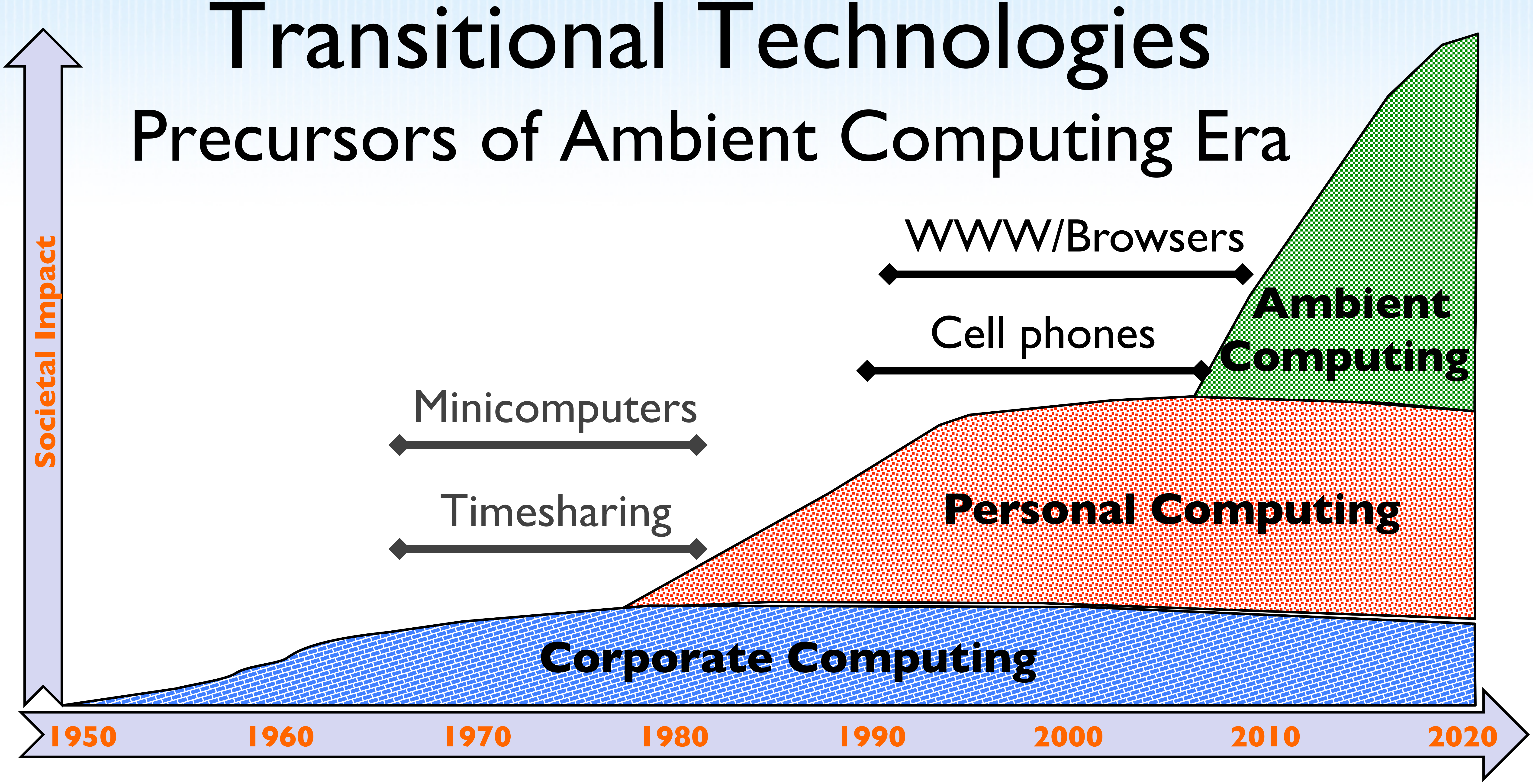
Precursors of PC Era



mozilla

Transitional Technologies

Precursors of Ambient Computing Era

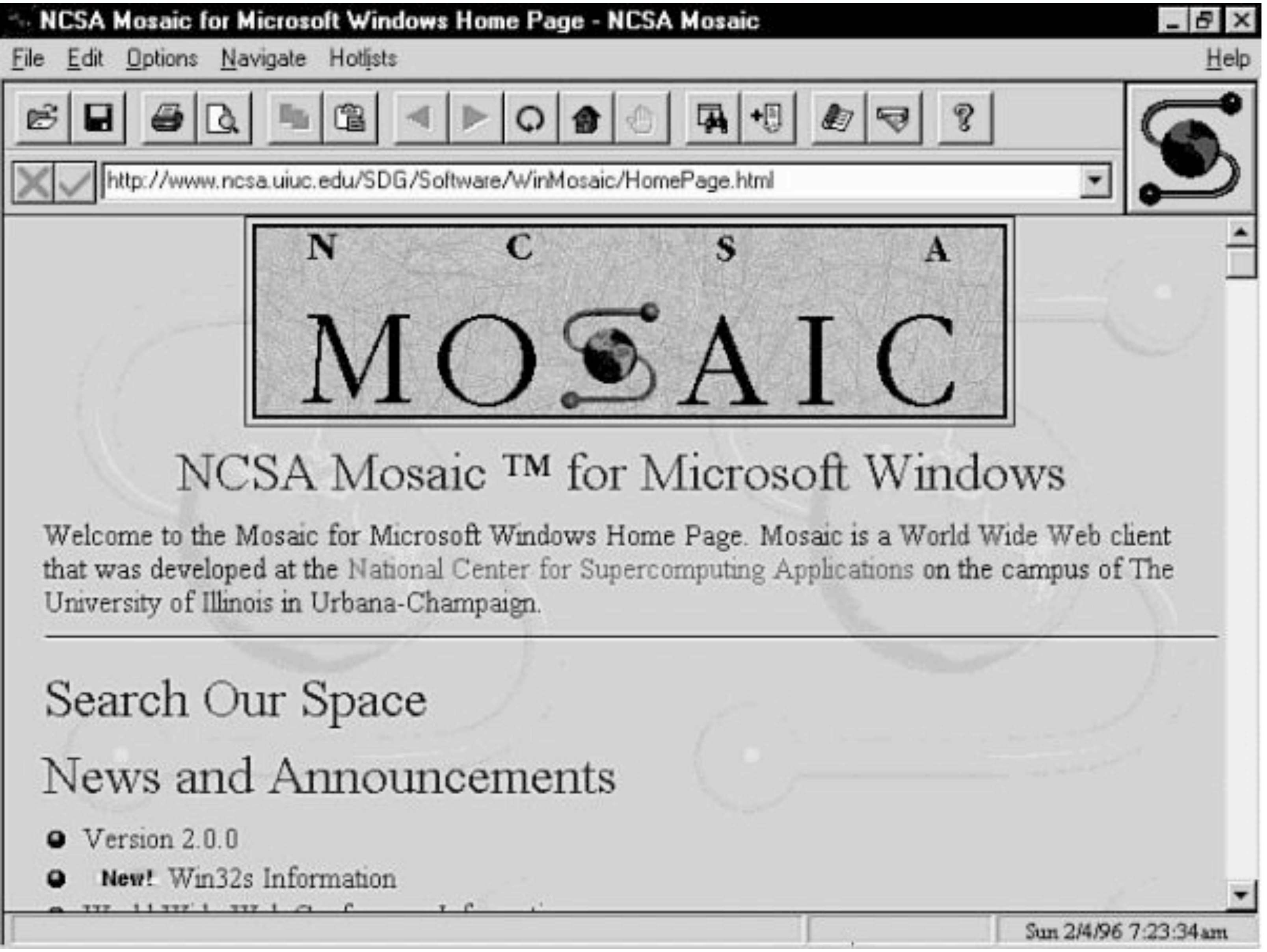


mozilla

The Cell Phone

- Always with you
- Always connected
- But initially dedicated to a single use case
- Initially not very smart

The WWW/Web Browser


NCSA Mosaic for Microsoft Windows Home Page - NCSA Mosaic
File Edit Options Navigate Hotlists Help
N C S A
MOSAIC

http://www.ncsa.uiuc.edu/SDG/Software/WinMosaic/HomePage.html

NCSA Mosaic ™ for Microsoft Windows

Welcome to the Mosaic for Microsoft Windows Home Page. Mosaic is a World Wide Web client that was developed at the National Center for Supercomputing Applications on the campus of The University of Illinois in Urbana-Champaign.

Search Our Space

News and Announcements

- Version 2.0.0
- New! Win32s Information

Sun 2/4/96 7:23:34am

- First real ubiquitous computer augmented availability of unstructured information
- Universal client/server computation model (HTTP/REST)
- Universal presentation platform
- Availability more important than performance
- A Web Browser is a classic PC application

Why are these only transitional

- Telephony is just a single application
- Wireless communications now common part of most devices
- WWW data access is an essential part of most applications
- Users don't need a "browser" to access the web any more
- Growing focus on solutions (apps) not tools (the browser)
- When something becomes ubiquitous, it disappears

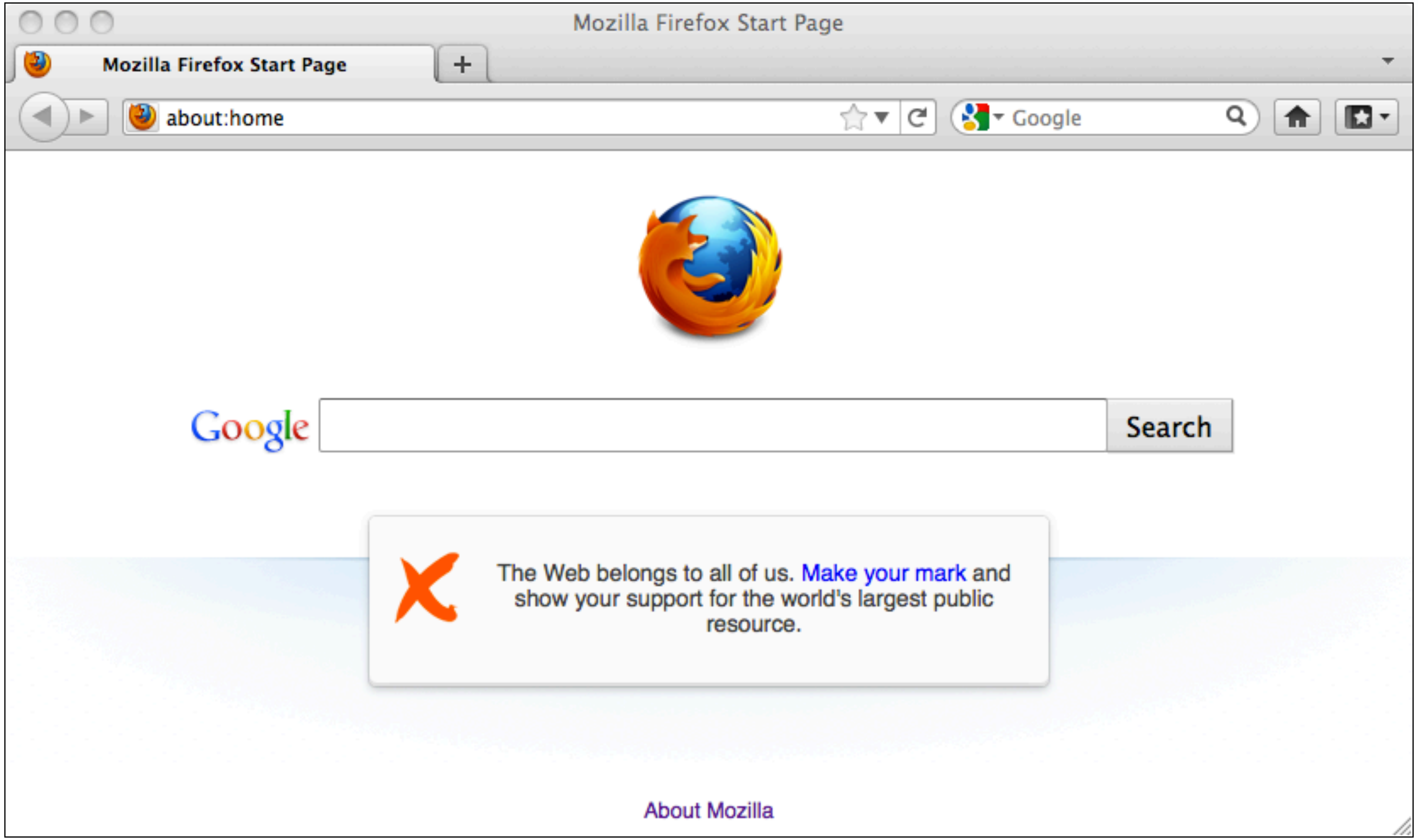
Every Computing Era Has a Dominant Application Platfrom

- Corporate Computing Era: IBM Mainframes
- Personal Conmputing Era: Microsoft/Intel PC
- Ambient Computing Era: T.B.D (or is it?)

Created by Market Demand,
"Good Enough" Technical Foundation,
and Superior Business Execution

mozilla

What do you have when you strip away the PC application part of a web browser?



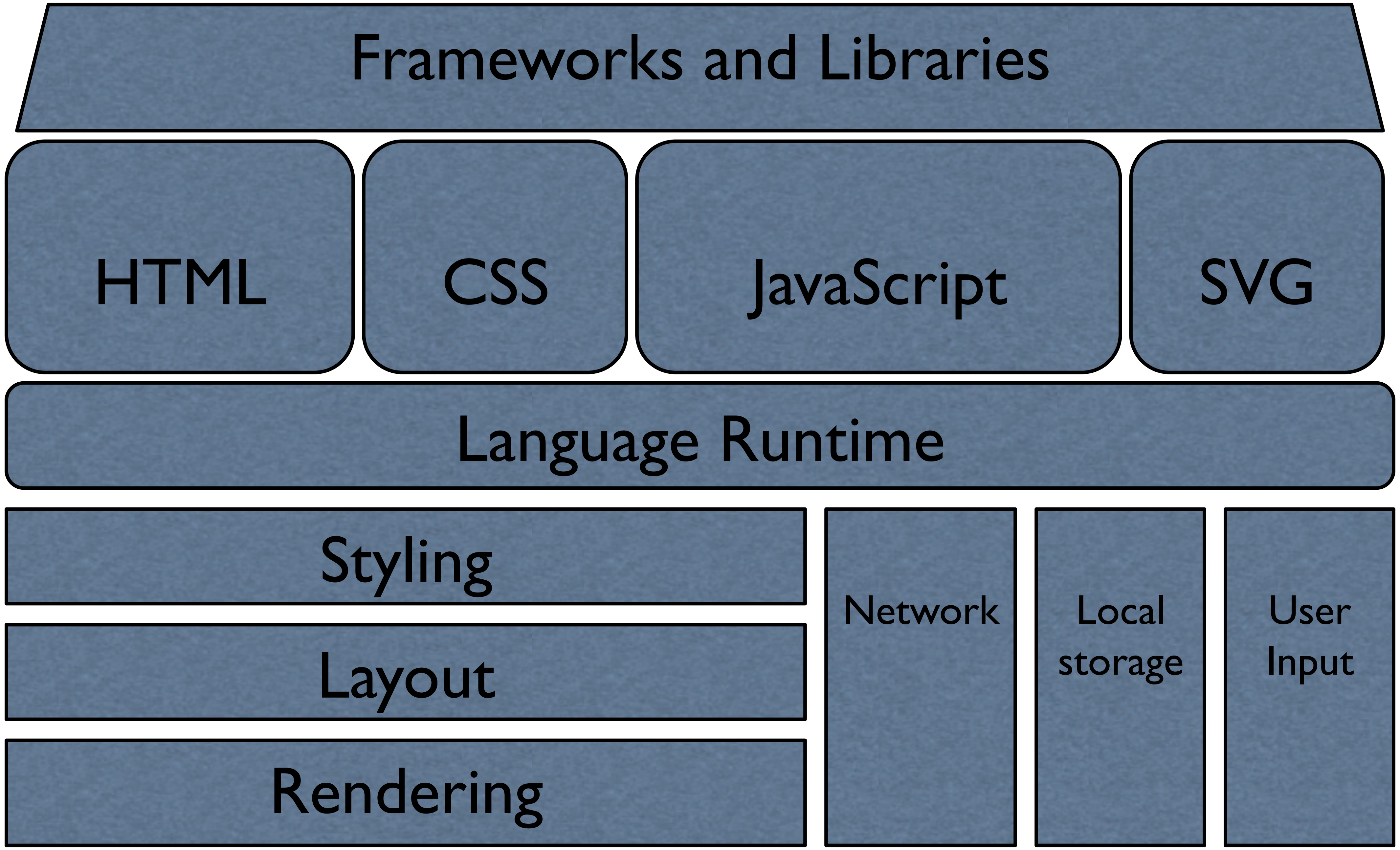
Mozilla Firefox Start Page

The Web belongs to all of us. Make your mark and show your support for the world's largest public resource.

About Mozilla

mozilla

What do you have when you strip away the PC application part of a web browser?

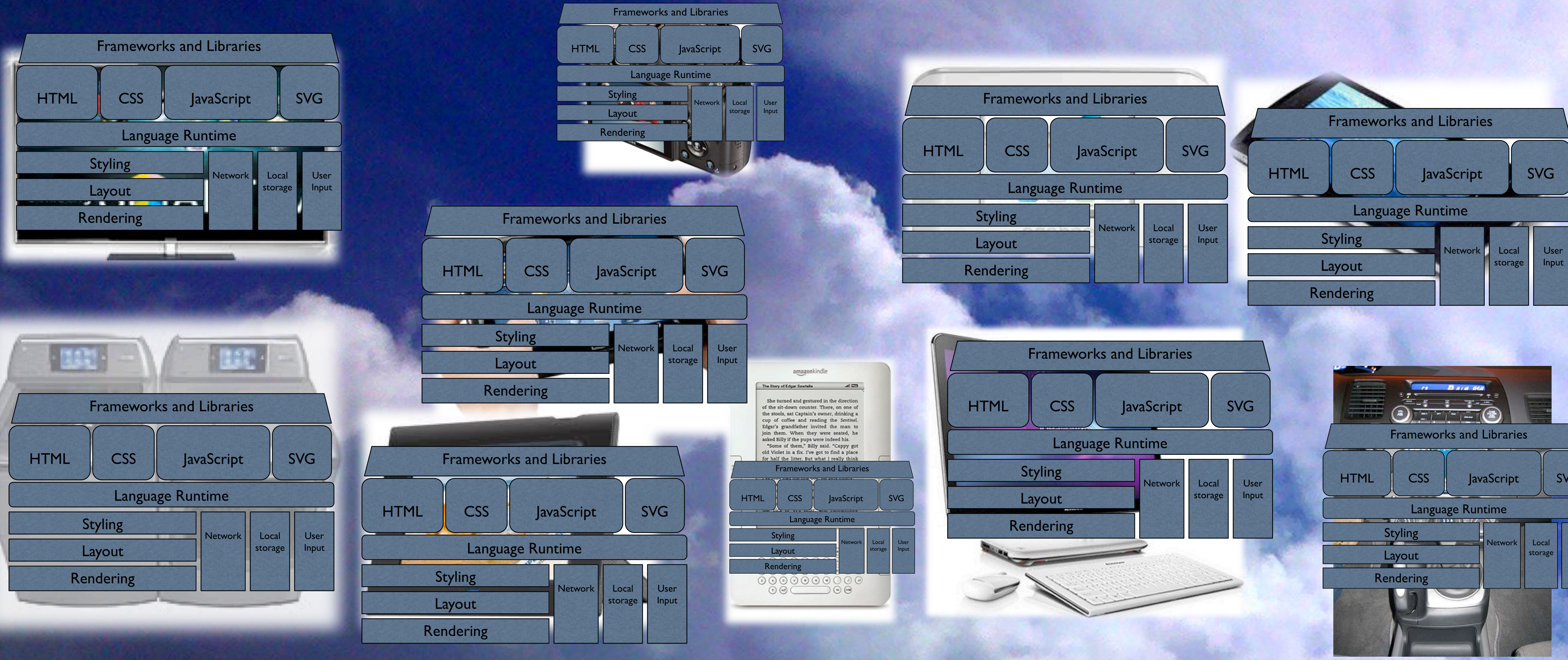


The Web Client Application Platform

mozilla

The Ambient Applications Platform

The Ambient Applications Platform



Frameworks and Libraries
HTML CSS JavaScript SVG
Language Runtime
Styling
Layout
Rendering
Network Local storage User Input

Frameworks and Libraries
HTML CSS JavaScript SVG
Language Runtime
Styling
Layout
Rendering
Network Local storage User Input

Frameworks and Libraries
HTML CSS JavaScript SVG
Language Runtime
Styling
Layout
Rendering
Network Local storage User Input

Frameworks and Libraries
HTML CSS JavaScript SVG
Language Runtime
Styling
Layout
Rendering
Network Local storage User Input

Frameworks and Libraries
HTML CSS JavaScript SVG
Language Runtime
Styling
Layout
Rendering
Network Local storage User Input

Frameworks and Libraries
HTML CSS JavaScript SVG
Language Runtime
Styling
Layout
Rendering
Network Local storage User Input

Frameworks and Libraries
HTML CSS JavaScript SVG
Language Runtime
Styling
Layout
Rendering
Network Local storage User Input

Frameworks and Libraries
HTML CSS JavaScript SVG
Language Runtime
Styling
Layout
Rendering
Network Local storage User Input

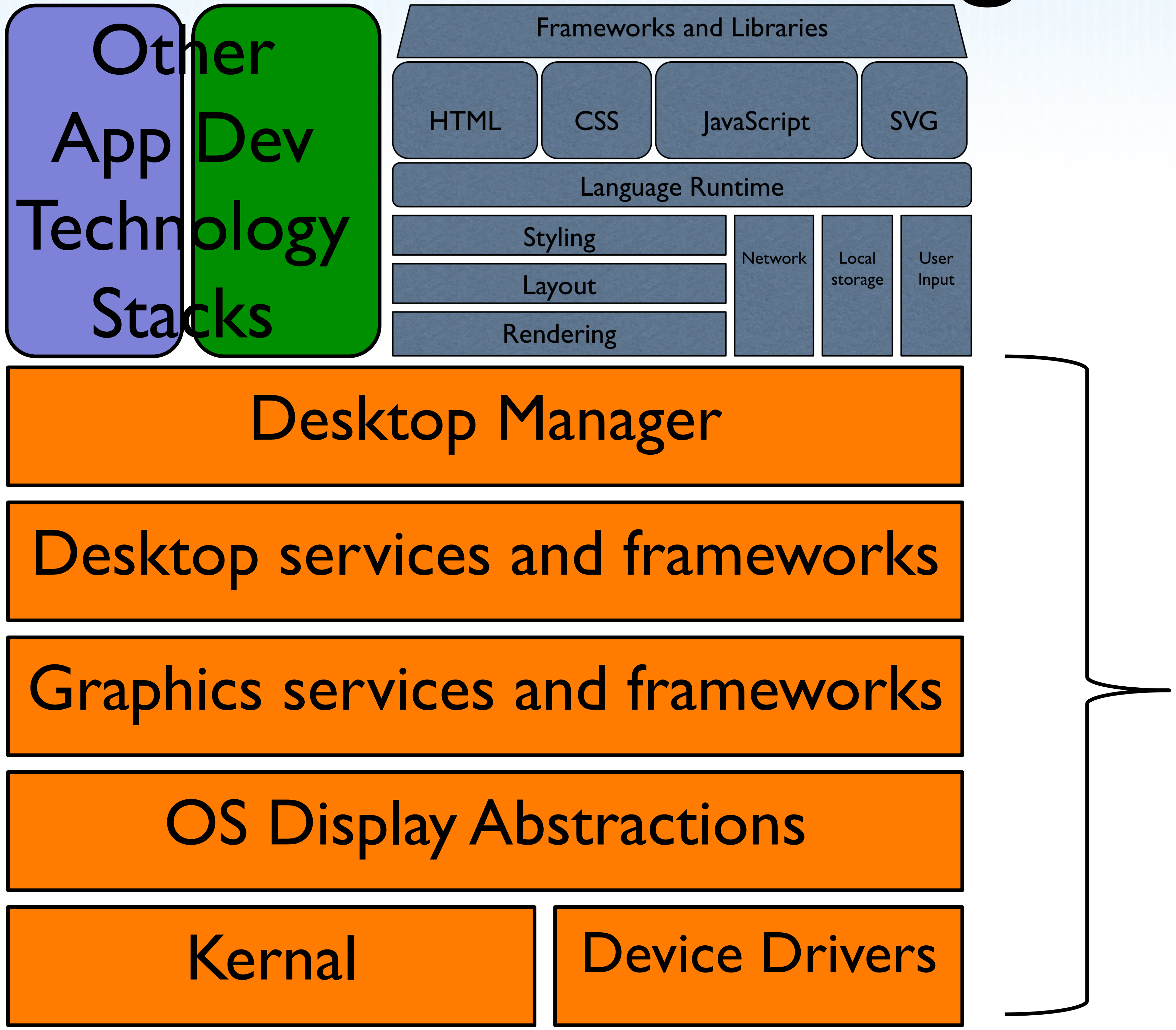
Frameworks and Libraries
HTML CSS JavaScript SV
Language Runtime
Styling
Layout
Rendering
Network Local storage

Web Apps vs. “Apps”

- HTML/CSS/JavaScript/WebApp APIs
- Server via: HTTP, JSON, XML, Sockets
- Works on any device
- Task specific “Chromeless” UIs emerging
- “App Stores” about to emerge

- Proprietary Language+Frameworks (Java, Objective-C, C#, etc.)
- Server via: HTTP, JSON, XML, Sockets
- Works only within a device family
- Task specific “Chromeless” UIs
- Tied to a specific “App Store”

New Platform Emergence



Other App Dev Technology Stacks

Frameworks and Libraries

HTML

CSS

JavaScript

SVG

Language Runtime

Styling

Layout

Rendering

Network

Local storage

User Input

Desktop Manager

Desktop services and frameworks

Graphics services and frameworks

OS Display Abstractions

Kernal

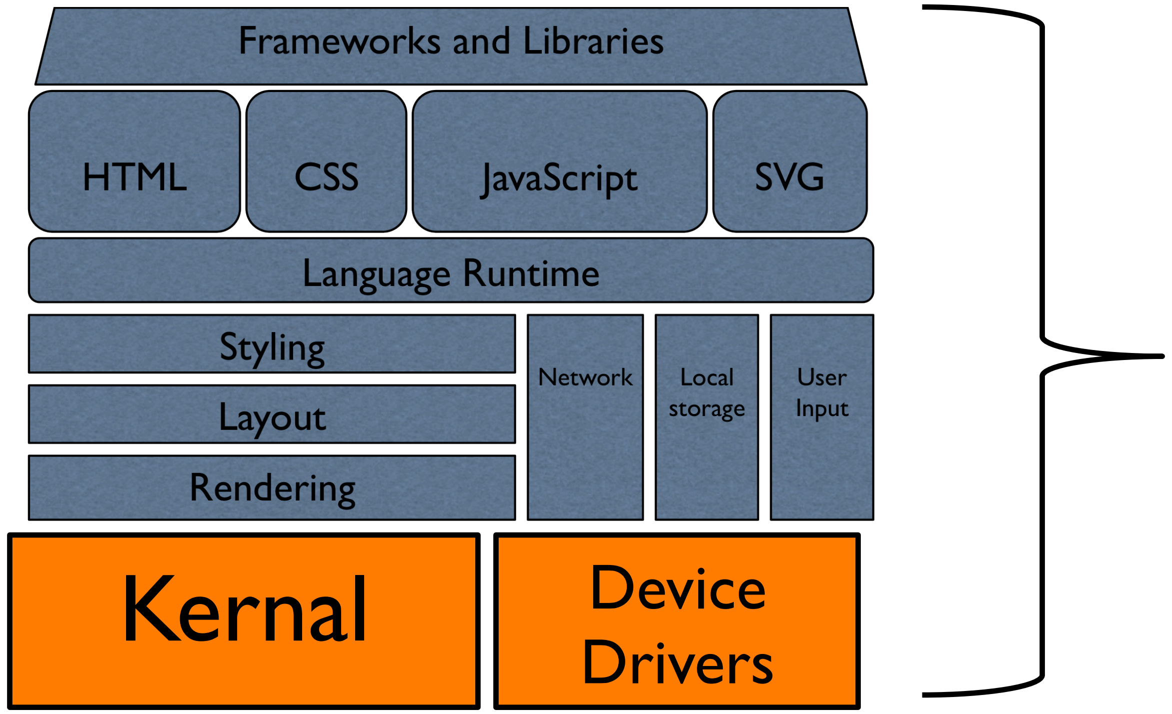
Device Drivers

“Modern” OS Circa 2005

mozilla

New Platform Emergence

- Mozilla Boot to Gecko
- Chrome OS
- Palm/HP WebOS
- Etc.



Frameworks and Libraries

HTML | CSS | JavaScript | SVG

Language Runtime

Styling

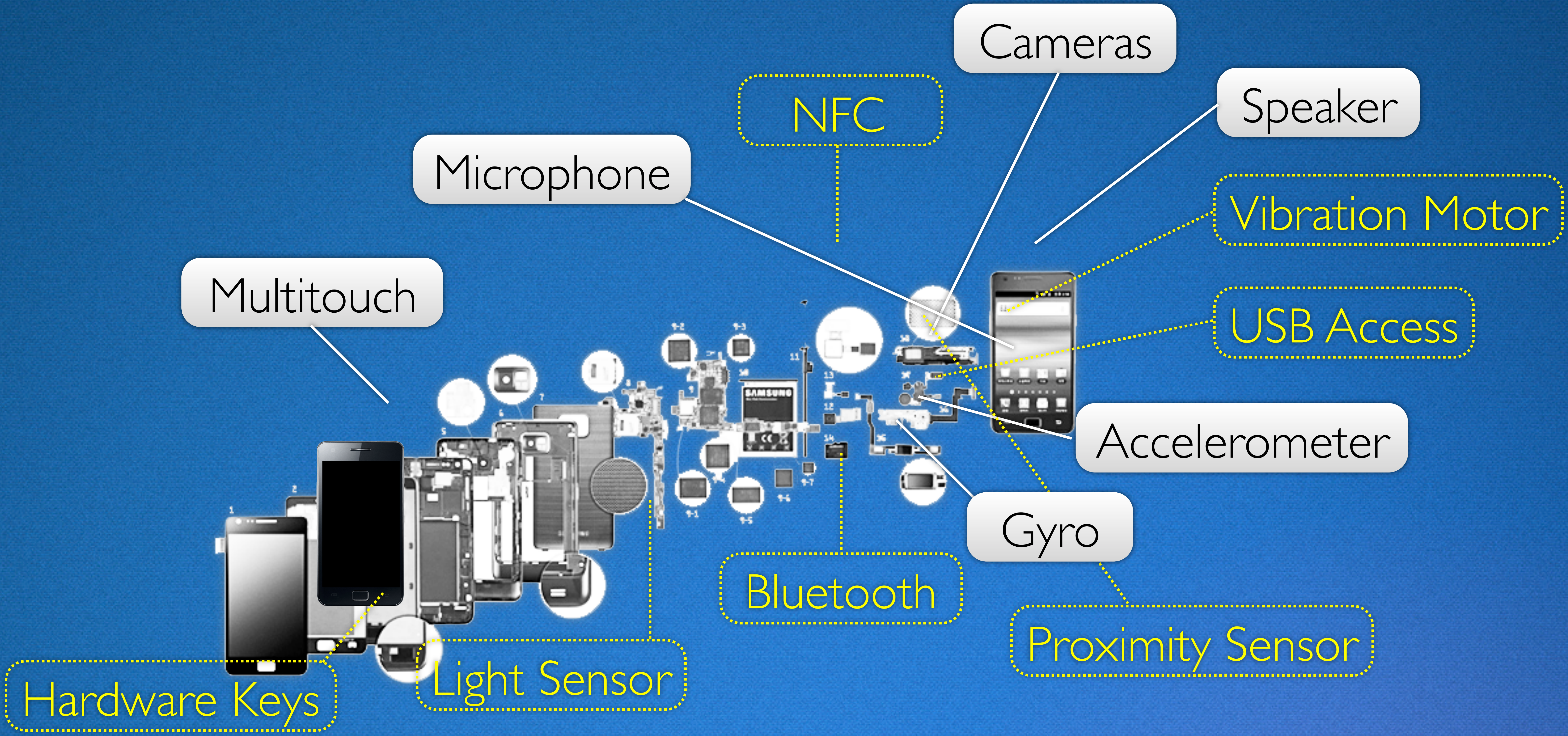
Layout

Rendering

Network | Local storage | User Input

Kernal | Device Drivers

Ambient
Device OS
Circa 2012



Cameras

NFC

Speaker

Microphone

Vibration Motor

Multitouch

USB Access

Accelerometer

Gyro

Bluetooth

Proximity Sensor

Light Sensor

Hardware Keys

Gray = Web APIs Exist

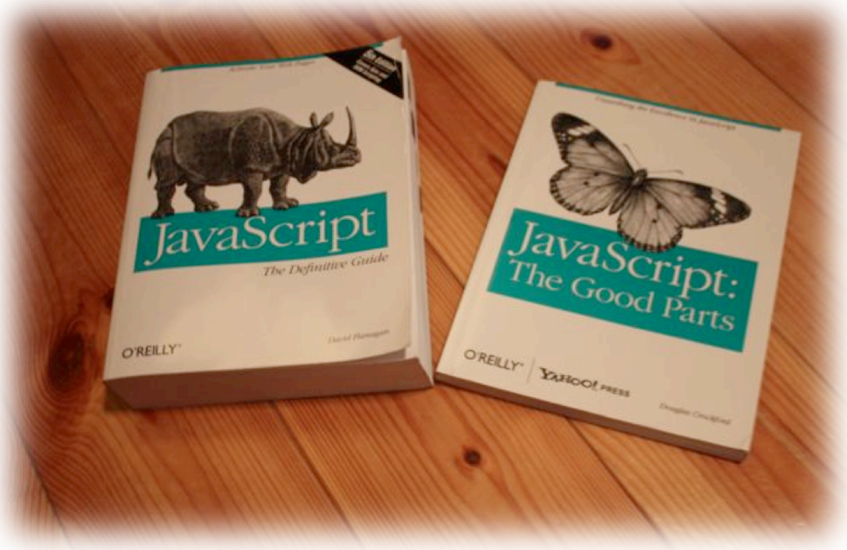
Yellow = Web APIs still needed

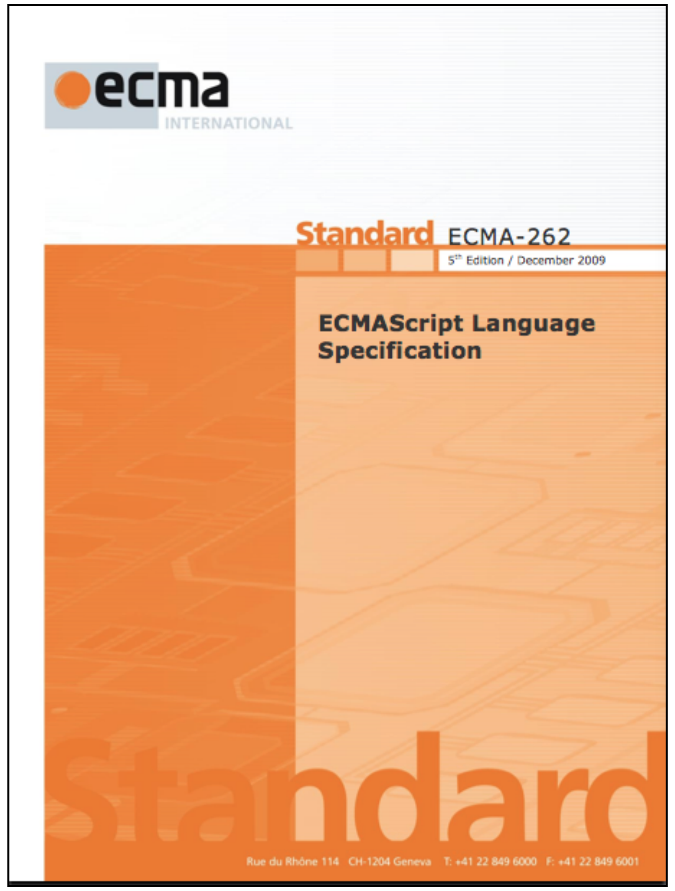
Areas of real concern

- Is it possible for a vender neutral standards-based open platform to achieve similar dominance?
- Cumbersome standardization processes
  - Really! W3C, Ecma, WhatWG, IETF, Kronos, …
- Who really drives innovation?
- Too slow rate of change?
- What actually gets implemented?

Each Computing Era has had Canonical Programming Languages

- Corporate Computing Era – COBOL/Fortran
- Personal Computing Era – C/C++ family
- Ambient Computing Era – JavaScript ??





ecma INTERNATIONAL

Standard ECMA-262

5th Edition / December 2009

ECMAScript Language Specification

Standard

Rue du Rhône 114 CH-1204 Geneva T: +41 22 849 6000 F: +41 22 849 6001

Why JavaScript?

Because "Worse is Better"

Dick Grabriel
http://www.dreamsongs.com/WorseIsBetter.html

- It's there – It's working
- It's good enough
- It's getting better
- What could replace it?
- How could that happen?

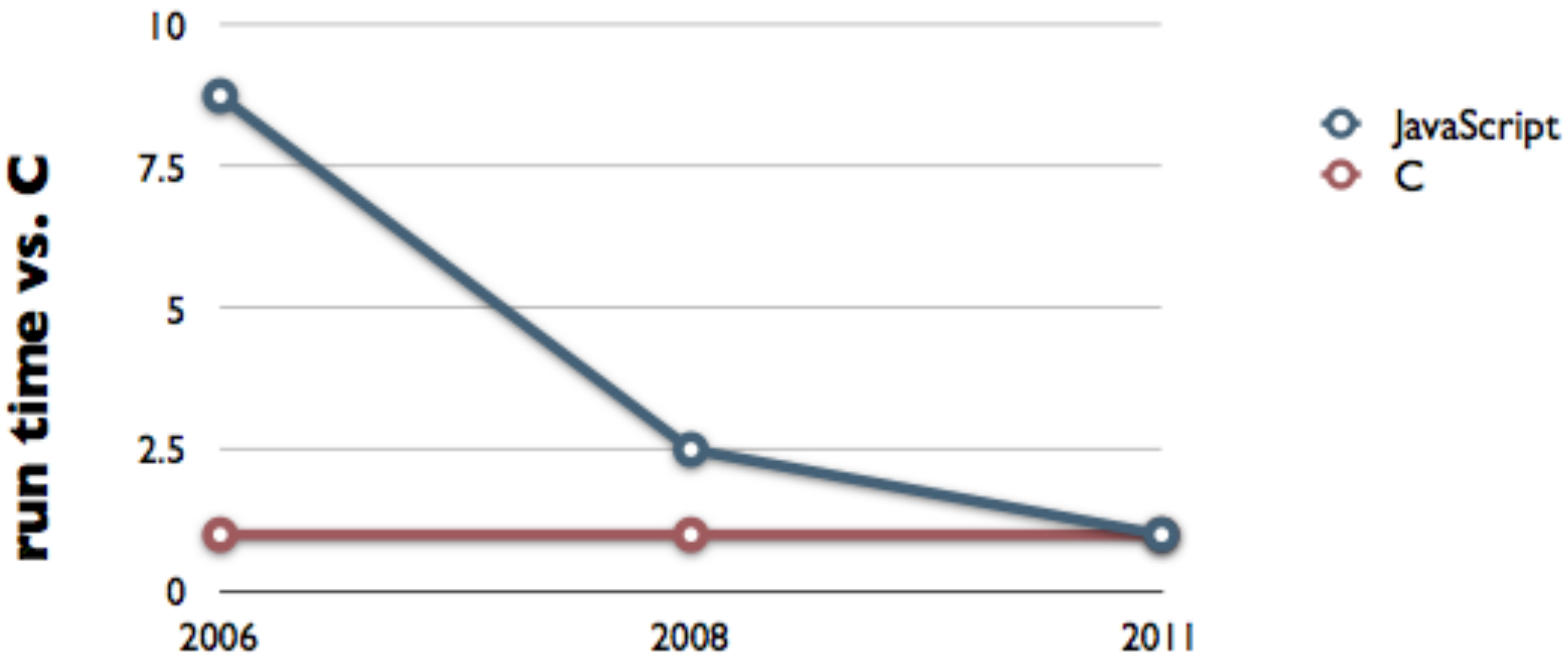
What is the world's most popular language?

A: French
B: English
C: Spanish
D: JavaScript

http://odetocode.com/Blogs/scott/archive/2009/03/18/signs-that-your-javascript-skills-need-updating.aspx

JavaScript Performance Over Time

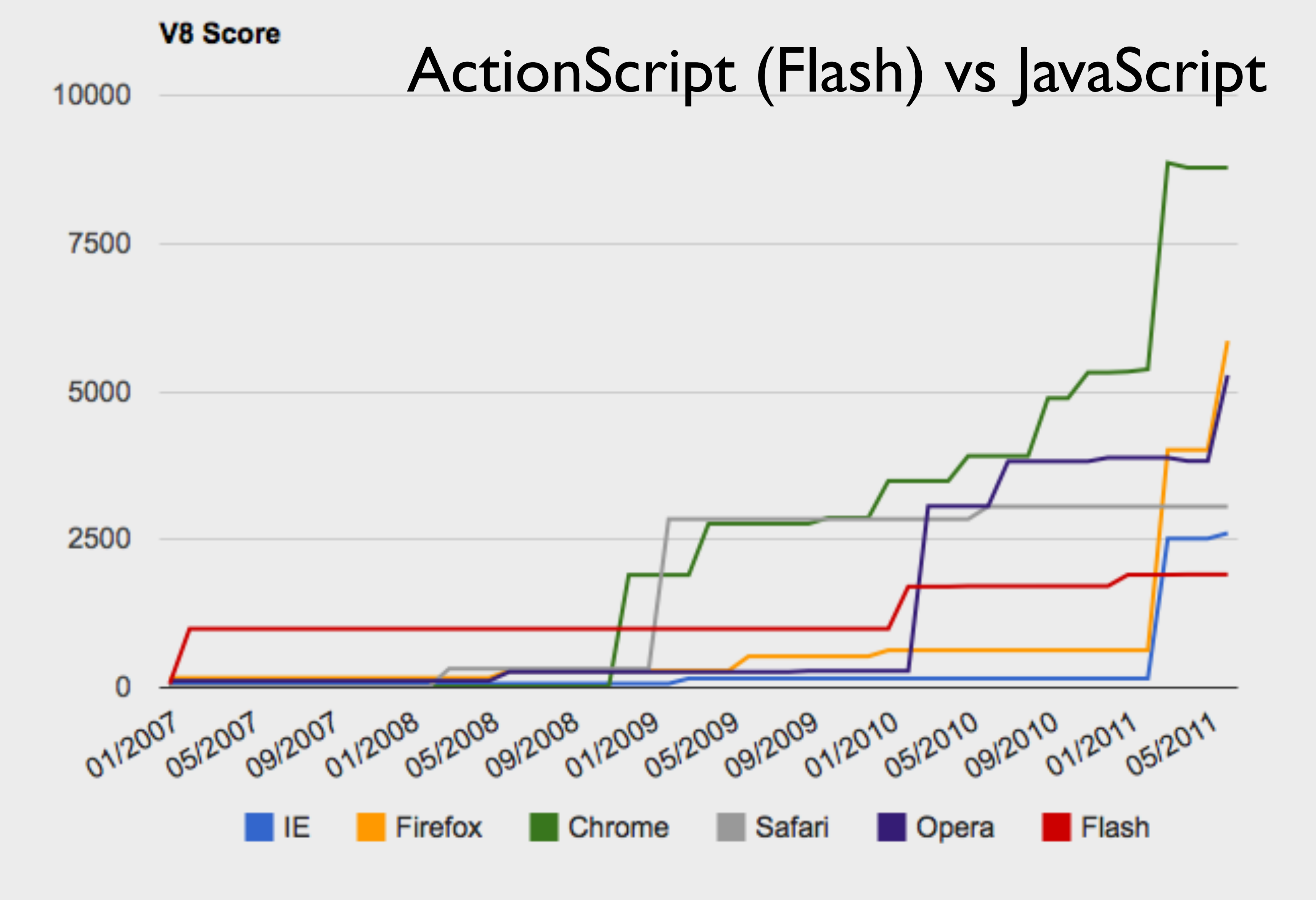
5 years of progress...



one program on one popular browser:

10x faster!

ActionScript (Flash) vs JavaScript



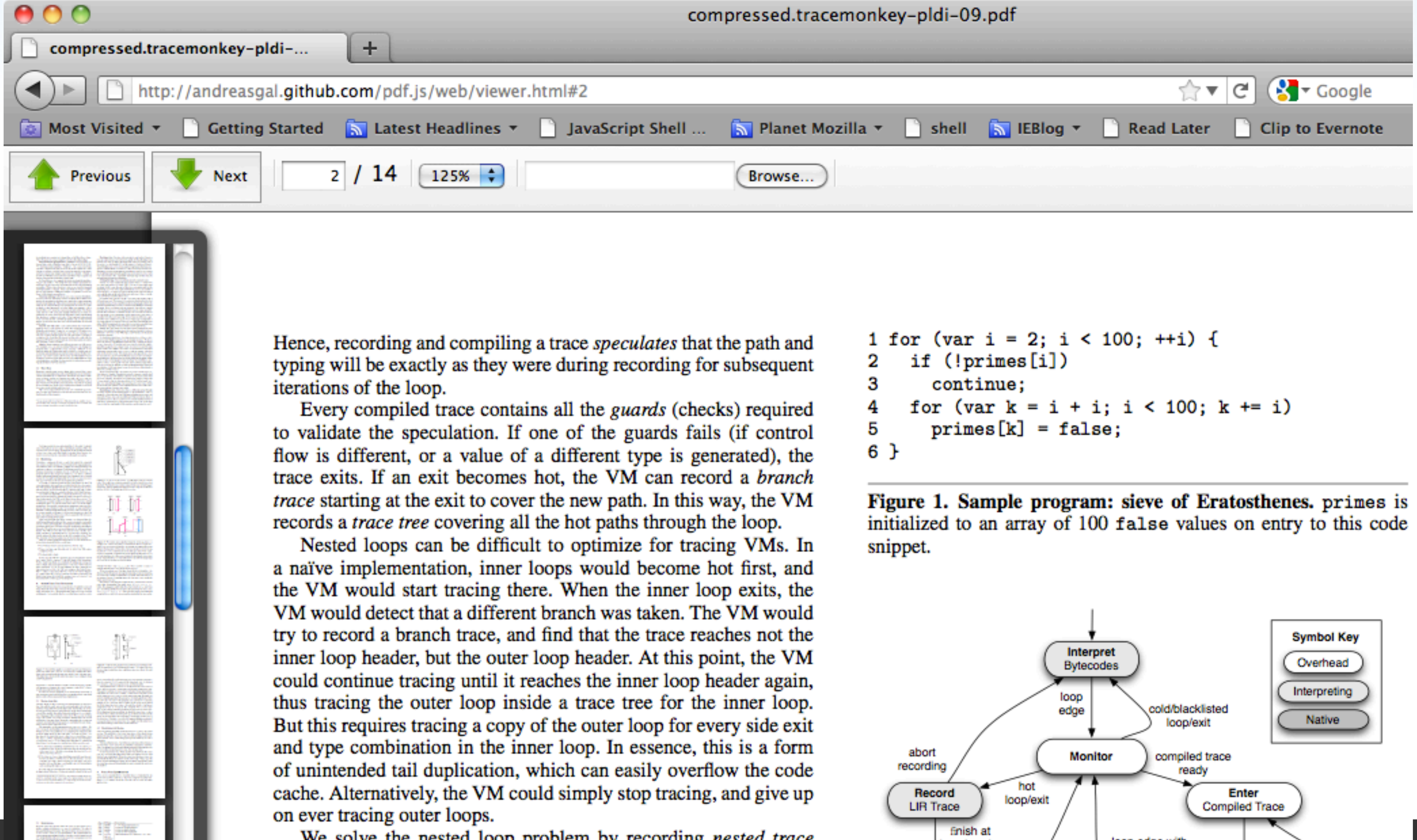
http://people.mozilla.com/~dmandelin/KnowYourEngines_Velocity2011.pdf

http://iq12.com/blog/as3-benchmark/

PDF Renderer in JavaScript

http://mozilla.github.com/pdf.js/web/viewer.html

https://github.com/mozilla/pdf.js



Hence, recording and compiling a trace *speculates* that the path and typing will be exactly as they were during recording for subsequent iterations of the loop.

Every compiled trace contains all the *guards* (checks) required to validate the speculation. If one of the guards fails (if control flow is different, or a value of a different type is generated), the trace exits. If an exit becomes hot, the VM can record a *branch trace* starting at the exit to cover the new path. In this way, the VM records a *trace tree* covering all the hot paths through the loop.

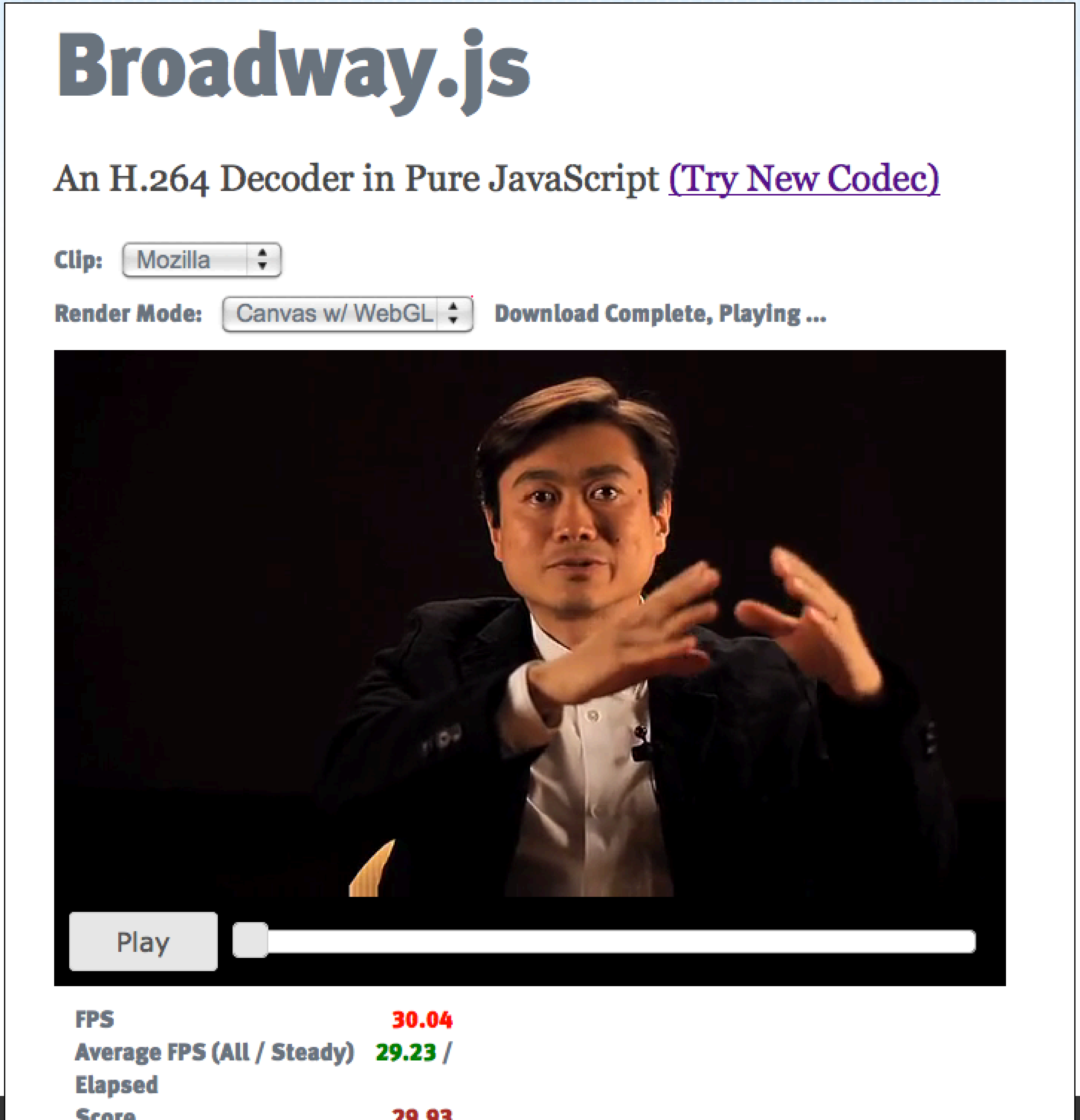
Nested loops can be difficult to optimize for tracing VMs. In a naïve implementation, inner loops would become hot first, and the VM would start tracing there. When the inner loop exits, the VM would detect that a different branch was taken. The VM would try to record a branch trace, and find that the trace reaches not the inner loop header, but the outer loop header. At this point, the VM could continue tracing until it reaches the inner loop header again, thus tracing the outer loop inside a trace tree for the inner loop. But this requires tracing a copy of the outer loop for every side exit and type combination in the inner loop. In essence, this is a form of unintended tail duplication, which can easily overflow the code cache. Alternatively, the VM could simply stop tracing, and give up on ever tracing outer loops.

We solve the nested loop problem by recording *nested trace*

```
for (var i = 2; i < 100; ++i) {
  if (!primes[i])
    continue;
  for (var k = i + i; i < 100; k += i)
    primes[k] = false;
}
```

Figure 1. Sample program: sieve of Eratosthenes. `primes` is initialized to an array of 100 `false` values on entry to this code snippet.

JavaScript Performance:



Broadway.js

An H.264 Decoder in Pure JavaScript (Try New Codec)

Clip: Mozilla

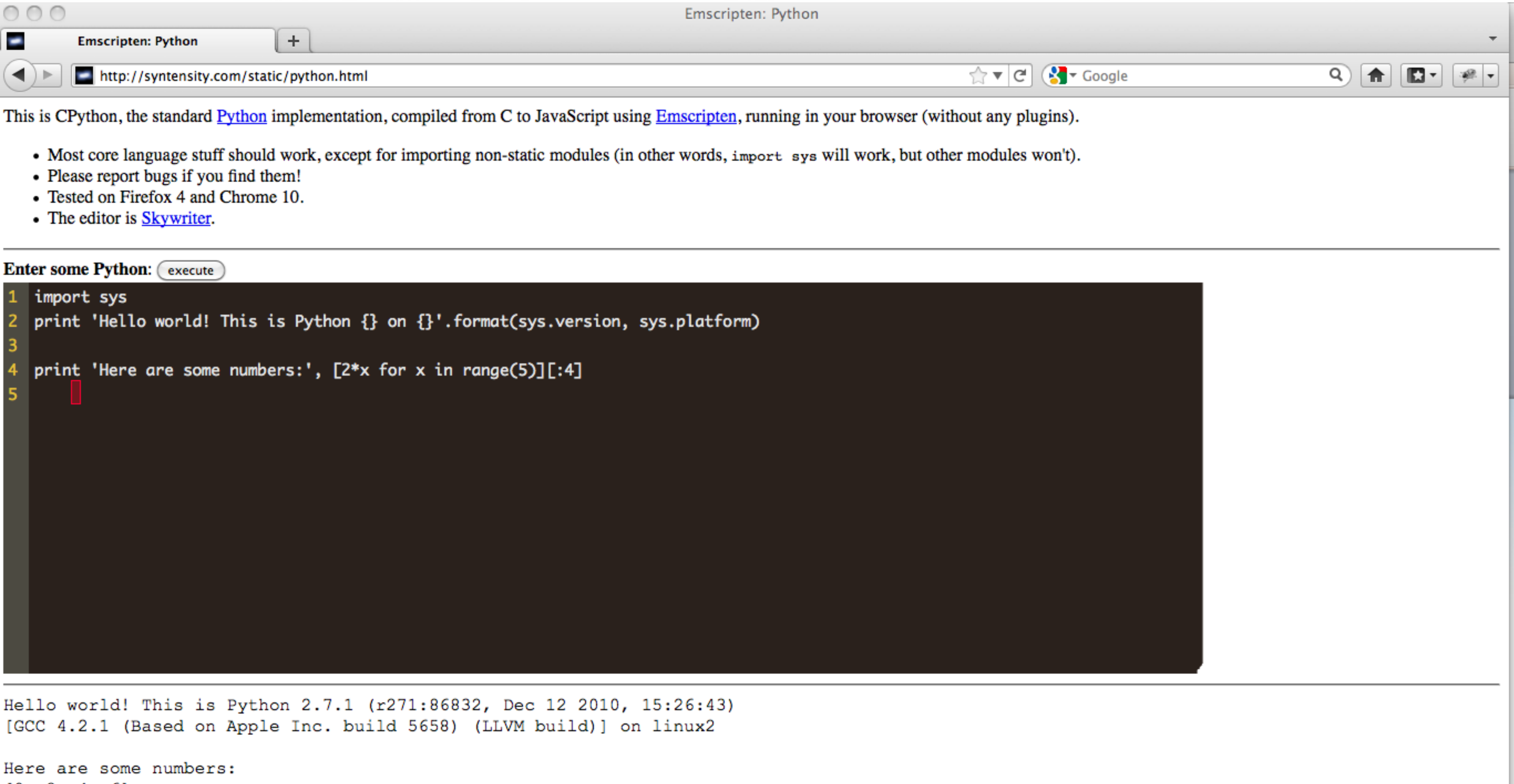
Render Mode: Canvas w/ WebGL   Download Complete, Playing ...

FPS	30.04
Average FPS (All / Steady)	29.23 /
Elapsed	
Score	29.93

http://mbebenita.github.com/Broadway/broadway.html

http://haxpath.squarespace.com/imported-20100930232226/2011/10/28/broadwayjs-h264-in-javascript.html

Emscripten
Compiling C/C++ to JavaScript



This is CPython, the standard Python implementation, compiled from C to JavaScript using Emscripten, running in your browser (without any plugins).

- Most core language stuff should work, except for importing non-static modules (in other words, `import sys` will work, but other modules won't).
- Please report bugs if you find them!
- Tested on Firefox 4 and Chrome 10.
- The editor is Skywriter.

Enter some Python: execute

```
import sys
print 'Hello world! This is Python {} on {}'.format(sys.version, sys.platform)

print 'Here are some numbers:', [2*x for x in range(5)][:4]

```

```
Hello world! This is Python 2.7.1 (r271:86832, Dec 12 2010, 15:26:43)
[GCC 4.2.1 (Based on Apple Inc. build 5658) (LLVM build)] on linux2

Here are some numbers:
```

Tranlates LLVM intermediate code to JavaScript

https://github.com/kripken/emscripten/wiki

http://syntensity.com/static/python.html

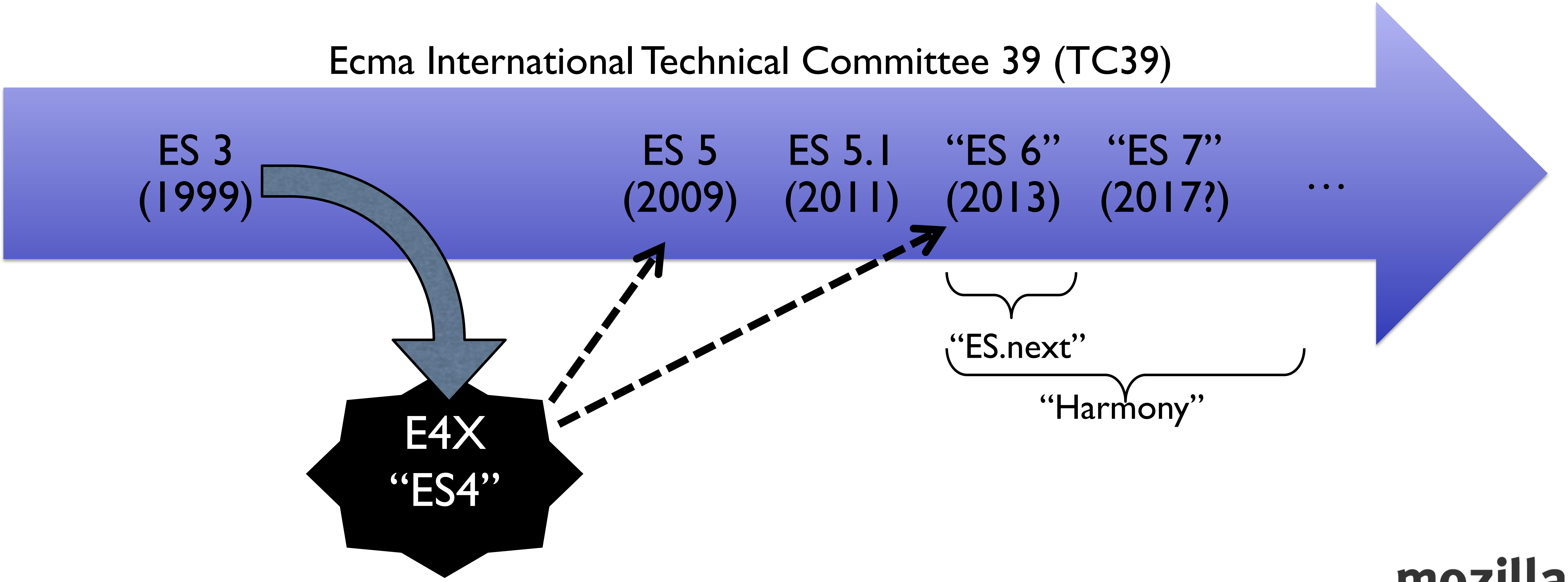
mozilla

Langauges That Compile To JavaScript

- CoffeeScript (“Improved” JavaScript syntax)
- Java (GWT)
- Script# (C# dialect)
- Dart
- ClojureScript (Lisp dialect)
- Dozens more listed at https://github.com/jashkenas/coffee-script/wiki/List-of-languages-that-compile-to-JS

JavaScript is the “virtual machine” of the Web Platform.

Evolving JavaScript
The ECMAScript Standards Process



Ecma International Technical Committee 39 (TC39)

ES 3 (1999)

ES 5 (2009)

ES 5.1 (2011)

“ES 6” (2013)

“ES 7” (2017?)

…

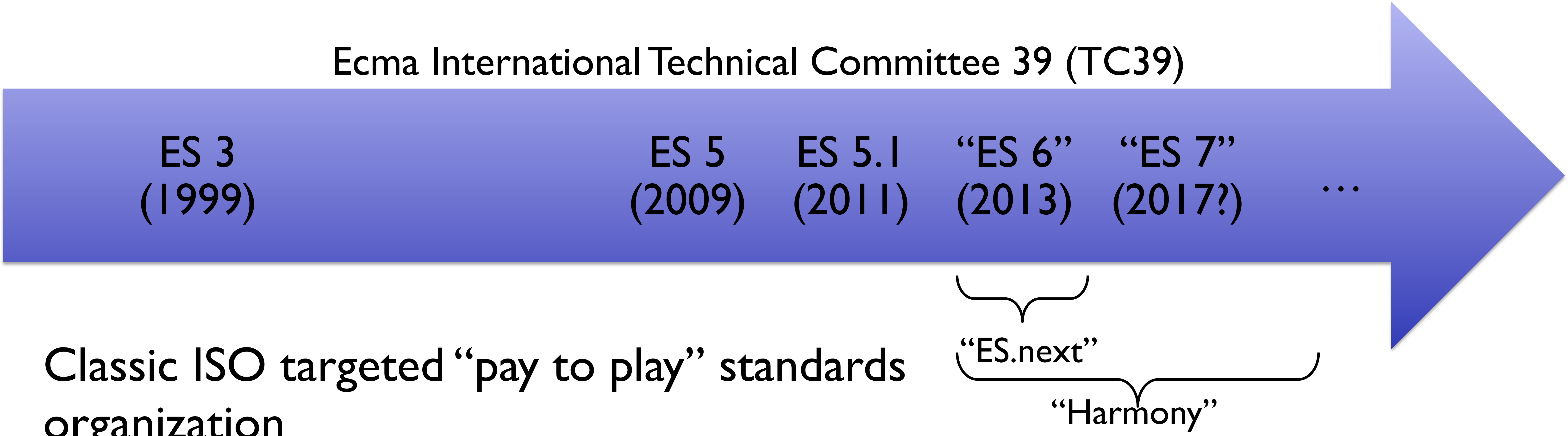
E4X “ES4”

“ES.next”

“Harmony”

mozilla

Evolving JavaScript
The ECMAScript Standards Process



Ecma International Technical Committee 39 (TC39)

ES 3 (1999)

ES 5 (2009)

ES 5.1 (2011)

“ES 6” (2013)

“ES 7” (2017?)

…

“ES.next”

“Harmony”

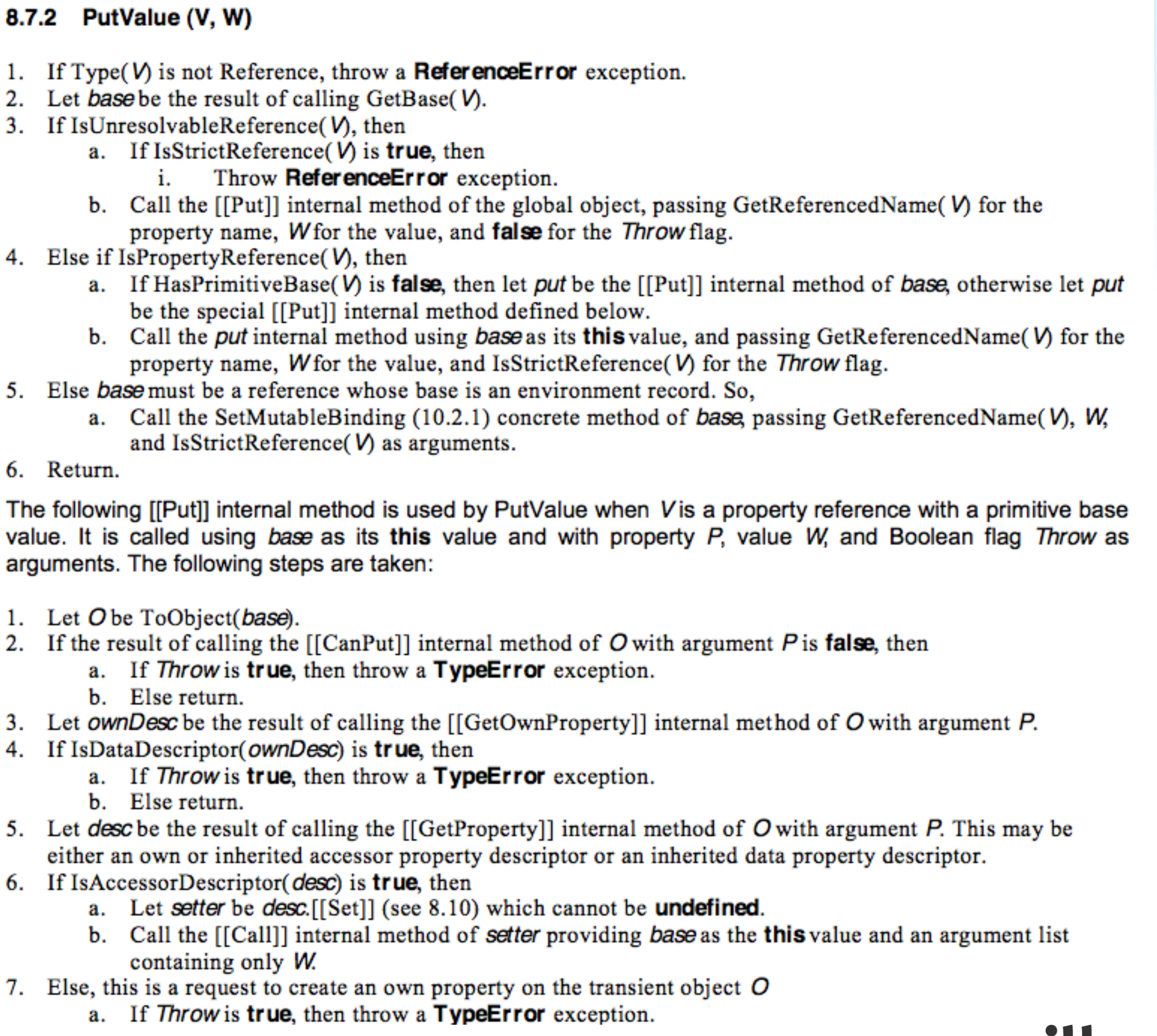
- Classic ISO targeted “pay to play” standards organization
- Open public discussion list es-discuss@mozilla.org
- Working documents all publically available at wiki.ecmascript.org

Interoperability is TC39’s highest priority

- A detailed and highly prescriptive specification
- Large, non-normative test suite for implementers

ecmascript test262

http://test262.ecmascript.org/



8.7.2 PutValue (V, W)

```
1. If Type(V) is not Reference, throw a ReferenceError exception.
2. Let base be the result of calling GetBase(V).
3. If IsUnresolvableReference(V), then
    a. If IsStrictReference(V) is true, then
        i. Throw ReferenceError exception.
    b. Call the [[Put]] internal method of the global object, passing GetReferencedName(V) for the
       property name, W for the value, and false for the Throw flag.
4. Else if IsPropertyReference(V), then
    a. If HasPrimitiveBase(V) is false, then let put be the [[Put]] internal method of base, otherwise let put
       be the special [[Put]] internal method defined below.
    b. Call the put internal method using base as its this value, and passing GetReferencedName(V) for the
       property name, W for the value, and IsStrictReference(V) for the Throw flag.
5. Else base must be a reference whose base is an environment record. So,
    a. Call the SetMutableBinding (10.2.1) concrete method of base, passing GetReferencedName(V), W,
       and IsStrictReference(V) as arguments.
6. Return.
```

The following [[Put]] internal method is used by PutValue when *V* is a property reference with a primitive base value. It is called using *base* as its **this** value and with property *P*, value *W*, and Boolean flag *Throw* as arguments. The following steps are taken:

```
1. Let O be ToObject(base).
2. If the result of calling the [[CanPut]] internal method of O with argument P is false, then
    a. If Throw is true, then throw a TypeError exception.
    b. Else return.
3. Let ownDesc be the result of calling the [[GetOwnProperty]] internal method of O with argument P.
4. If IsDataDescriptor(ownDesc) is true, then
    a. If Throw is true, then throw a TypeError exception.
    b. Else return.
5. Let desc be the result of calling the [[GetProperty]] internal method of O with argument P. This may be
   either an own or inherited accessor property descriptor or an inherited data property descriptor.
6. If IsAccessorDescriptor(desc) is true, then
    a. Let setter be desc.[[Set]] (see 8.10) which cannot be undefined.
    b. Call the [[Call]] internal method of setter providing base as the this value and an argument list
       containing only W.
7. Else, this is a request to create an own property on the transient object O
    a. If Throw is true, then throw a TypeError exception.
```

mozilla

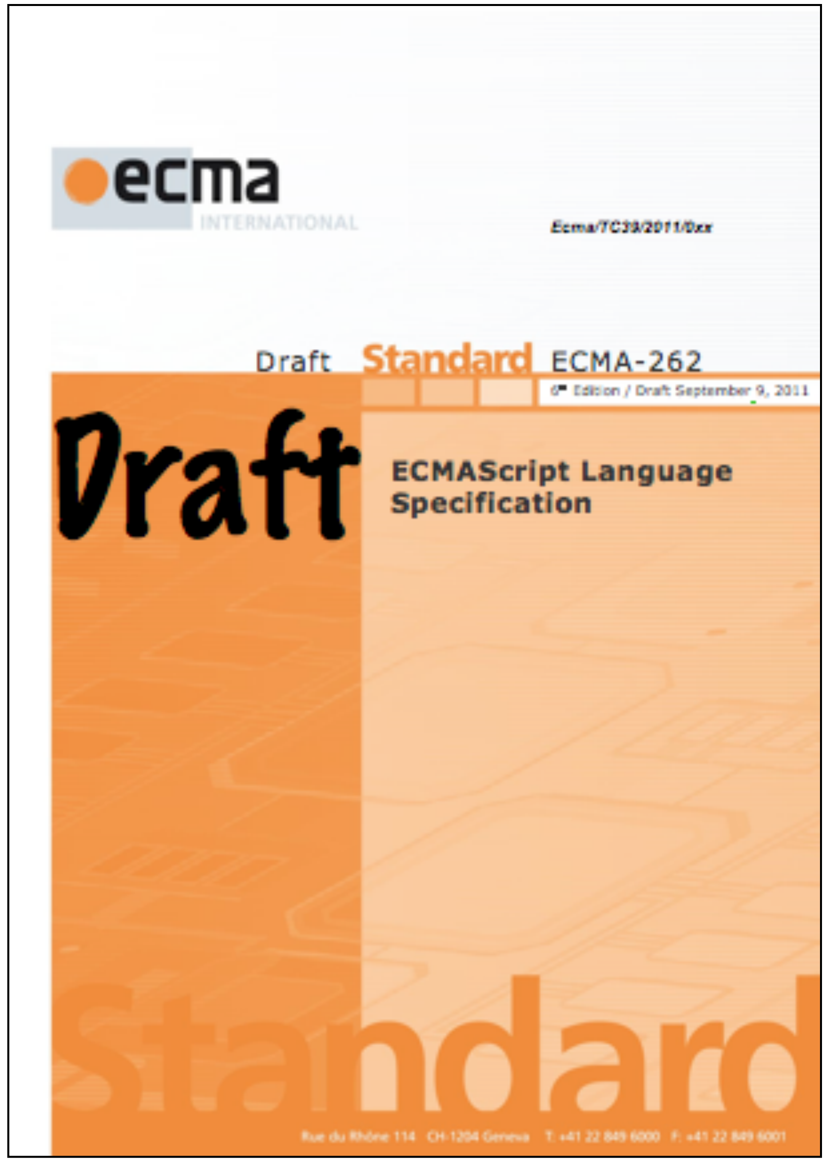
Evolving JavaScript ECMAScript Harmony Goals

1. Be a better language for writing:
   - A. complex applications;
   - B. libraries (including the DOM) shared by those applications;
   - C. code generators targeting the new edition.
2. …

http://wiki.ecmascript.org/doku.php?id=harmony:harmony

Major ES.next Enhancements

- Modules
- Sanding-boxing module loaders
- Proxy Objects – low level behavioral intercession
- Control abstraction via iterators and generators
- Array comprehensions, String interpolation
- super references
- Encapsulated state via gensym-like private names
- Better support for class-style inheritance



ecma INTERNATIONAL

Ecma/TC39/2011/0xx

Draft Standard ECMA-262

6th Edition / Draft September 9, 2011

Draft

ECMAScript Language Specification

Standard

Rue du Rhône 114 CH-1204 Geneva T: +41 22 849 6000 F: +41 22 849 6001

What about the server?

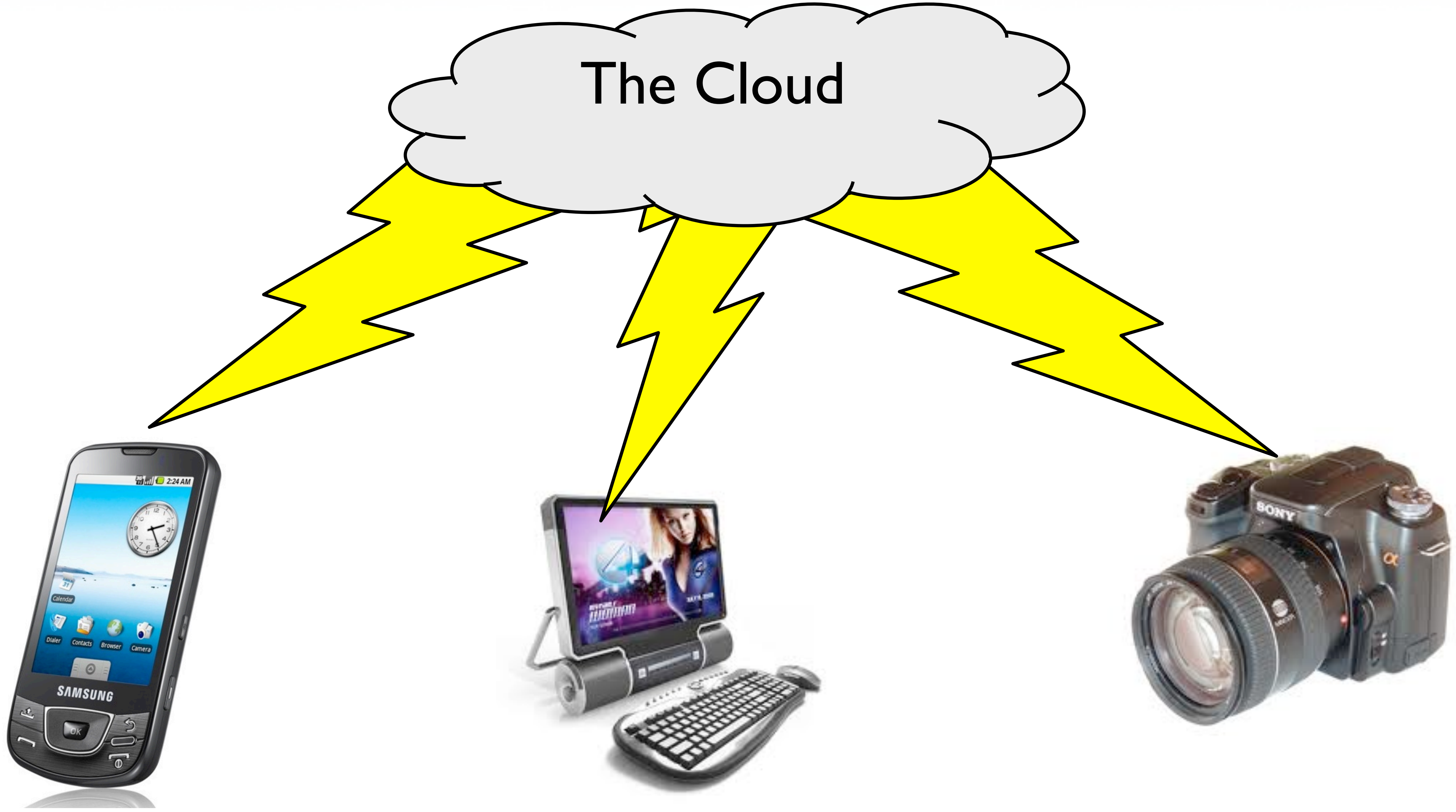
- The server-side of applications don’t have the the “single platform” imperative of clients: Java, .Net, Python, Ruby
- Growing interest in also using JavaScript for the server portion of applications

Developers want to use same skills and technical resources on both client and server

- node.js (http://nodejs.org/) has exploded on the scene

```
//hello world server
var http = require('http');
http.createServer(function (req, res) {
   res.writeHead(200, {'Content-Type': 'text/plain'});
   res.end('Hello World\n');
 }).listen(1337, "127.0.0.1");
console.log('Server running at http://127.0.0.1:1337/');
```

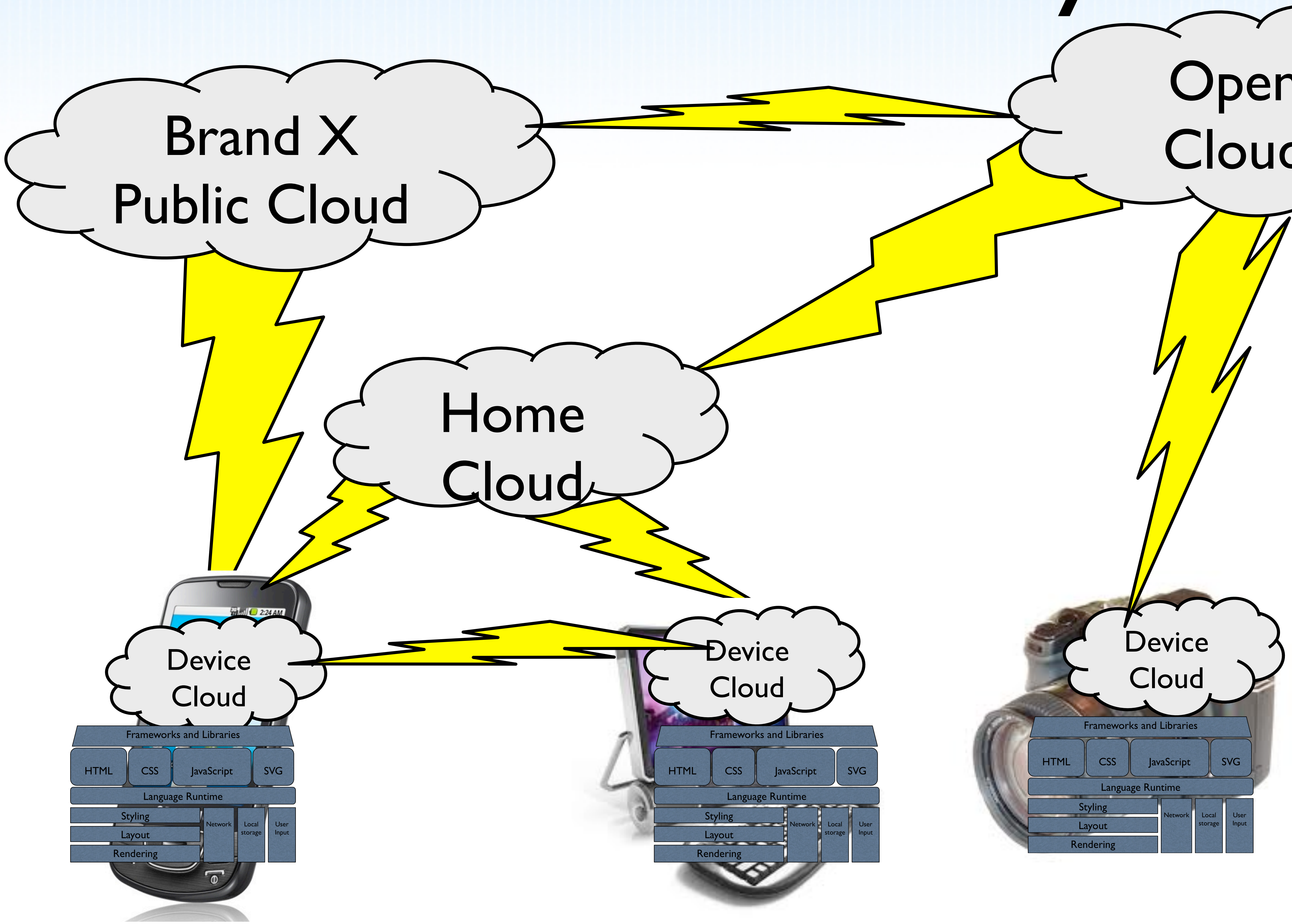
What About “The Cloud”



Concurrency/Using Multicores

- The Browser App Platform uses a very simple no shared state concurrency model.
  - Web Workers
  - PostMessage and JSON data records
  - Event loops / Continuation passing style
- Essentially the same model as browser ←→ webserver
- Can be fully distributed, not just client/server.

Clouds All the Way Down?

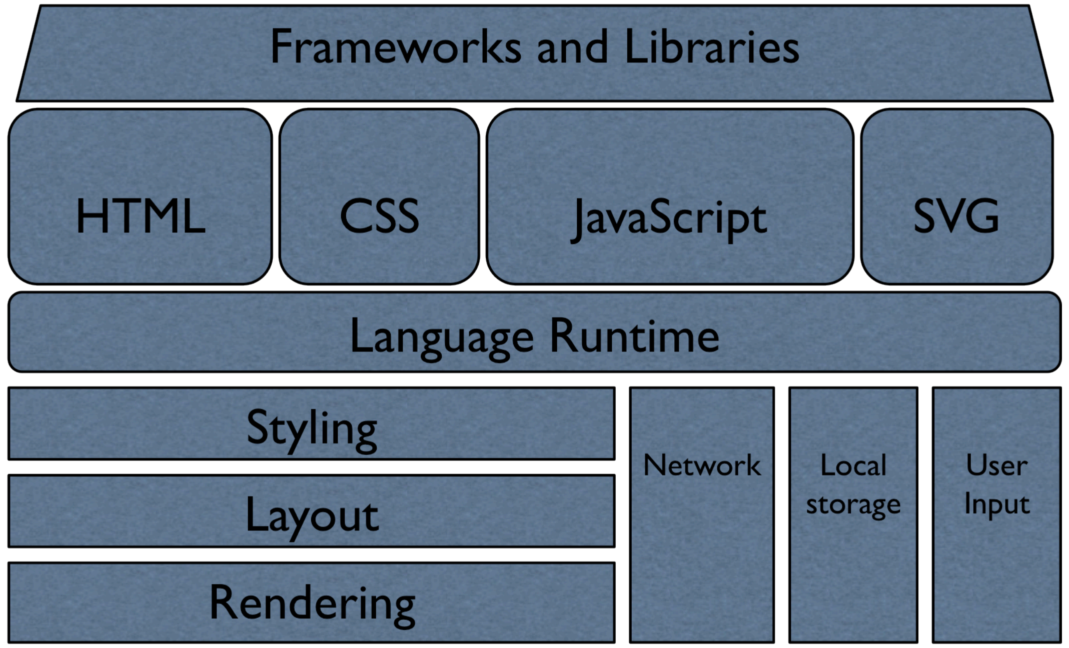


- Common programming models
- Common APIs
- Local/remote cloud transparency

Proprietary Silos

Walled Gardens

or a Standards-based Open Platform?



Users: Customers or Merchandise?

- Are we building devices and services that empower people?
- Or, are we just trying to attract their attention so we can turn them into monetizable assets?

User Sovereignty

- Can we keep the user in control of their computer augmented life?
- Some of the key issues
  - ❑ Identity
  - ❑ Privacy
  - ❑ Freedom to use what we choose
  - ❑ Reliable storage of our digital assets
  - ❑ Durability of data and applications

> "Mozilla's mission is to build user sovereignty into the fabric of the internet"
>
> Mitchell Baker
> Mozilla Chairperson

What's It All Mean?

- A twenty year vision is easy to craft,
  the path that will take us there is unknowable.
- The life of an innovator at the beginning of an era is
  very different from the waning days of an established era.
- The browser derived web client platform is a strong candidate to be the dominant application platform of the Ambient Computing Era.
- If you aren't all ready doing so, it's time to take JavaScript seriously.
- We are all users, let's create the world we want to live in

mozilla

This is the most exciting time for the computing industry since the 1980’s.

Enjoy it!

Follow up

http://www.wirfs-brock.com/allen/posts/category/post-pc-computing

allenwb@mozilla.com

@awbjs