

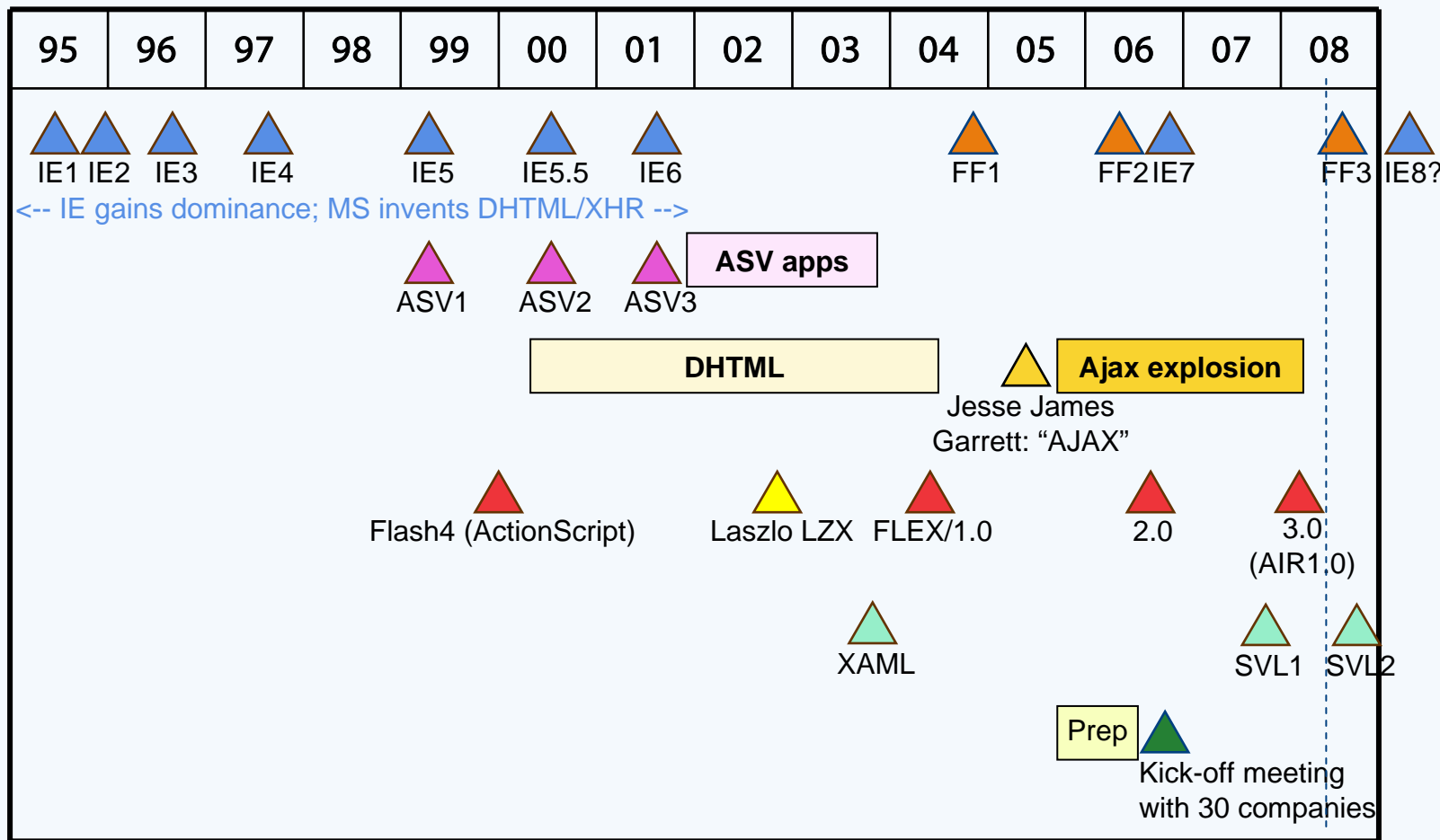
How Ajax Changes the Game for SVG

Jon Ferraiolo

IBM and OpenAjax Alliance

26 August, 2008

Desktop RIA Timeline



Agenda

- **SVG from Jon's perspective**
- **Ajax (and OpenAjax)**
- **How Ajax Changes the Game**

Conception

- Many saw that the browser needed 2D graphics
- Jonathan Gay and FutureSplash launched in 1993
 - Subsequently purchased by MACR and renamed to Flash
- Various other proprietary formats
- WebCGM
- PDF = PostScript as a viewable document
- Chris Lilley
 - 1994 – First pronouncements? (earliest date I found)
 - 1996/1997 – Requirements for “scalable vector graphics”

Formation of the SVG WG

- **By early 1998, several big companies concluded:**
 - 2D vector graphics in browser was next big thing
 - Must be expressed in XML
 - Must be developed at W3C
- **As a result:**
 - PGML Submission: Adobe, IBM, Netscape, Sun
 - VML Submission: Microsoft, Autodesk, HP, MACR, Visio
 - WebCGM Submission: Bob Hopgood and colleagues
- **W3C formed Scalable Vector Graphics WG**

Birth

- **SVG WG kick-off meeting on Aug 31, 1998**
 - ~40 companies, including the PGML and VML companies
- **SVG 1.0 was a huge effort**
 - Tricky politics
 - Technical complexities around graphics
 - Technical complexities around integration issues with other W3C initiatives (XML, XML Namespaces, DOM, DOM Events, CSS, SMIL, XLink, XPointer, ...)
 - W3C changed its rules a couple of times
- **W3C approves the SVG 1.0 Rec on Sept. 4, 2001**

Adobe: the main early driver behind SVG

- **Adobe's SVG strategy**
 - Strong involvement in SVG WG
 - Ubiquitous browser plugin
 - Monetize via SVG tools, such as Adobe Illustrator
- **For Ferraiolo, SVG was just the first phase of a broader strategy**

All of the ingredients for success

- **Everything set up for SVG success**
 - Spring 2001 – Acrobat Reader 5 released
 - *ASV2 bundled inside, installed automatically*
 - *ASV2 had auto-update logic (therefore, would update to ASV3)*
 - *Acrobat Reader 5 pre-installed on most desktop PCs*
 - 4 September 2001 – SVG 1.0 is a W3C Recommendation
 - Fall 2001 – ASV3 and Adobe Illustrator 10 released
 - Early 2002 – More SVG products from Adobe (Adobe GoLive, Graphics Server, Adobe Document Server)
- **Adobe puts SVG into maintenance**

Childhood

- **Strong adoption of SVG/ASV on desktop in 2002-2003**
 - Favorite technology for many Enterprise developers
 - *Rich feature set*
 - *Relatively easy programming (declarative XML plus JavaScript)*
 - *Integrates very nicely with back-end XML servers*
 - *Standards-based*
 - *Backed by big company (or so it appeared)*
 - *➔ Nearly every major Enterprise product had an ASV dependency*
 - Favorite technology for visionary developers
 - *Allowed rich web features that were only possible at the time in Flash*
 - *But Flash was proprietary and required developers to program within an animation tool with all logic in frame zero*
- **But with no future releases of ASV, and other alternatives emerging, many desktop developers moved away from ASV (and desktop SVG)**

Early Adulthood

- As desktop SVG declined, mobile SVG grew
- SVG Mobile becomes a W3C Recommendation in Jan. 2003
 - SVG Tiny 1.1 designed by mobile industry to fit on volume phones
 - *Could be built with a ~100K footprint*
- Other mobile standards organizations adopt SVG Tiny
 - MMS, OMA browser standards, J2ME/JSR226, MPEG LAsER, 3GPP DIMS
 - J2ME/JSR 226 (and JSR 248) probably the most significant
- Multiple reliable suppliers of SVG Tiny
 - Particularly Bitflash and Ikivo
- As a result, SVG Tiny now is deployed on hundreds of millions of phones

SVG Tiny – Mixed Success

■ Good news

- Checklist requirement on various mobile standards
- Deployed on hundreds of millions of phones
- Real commercial applications based on SVG, such as mobile video

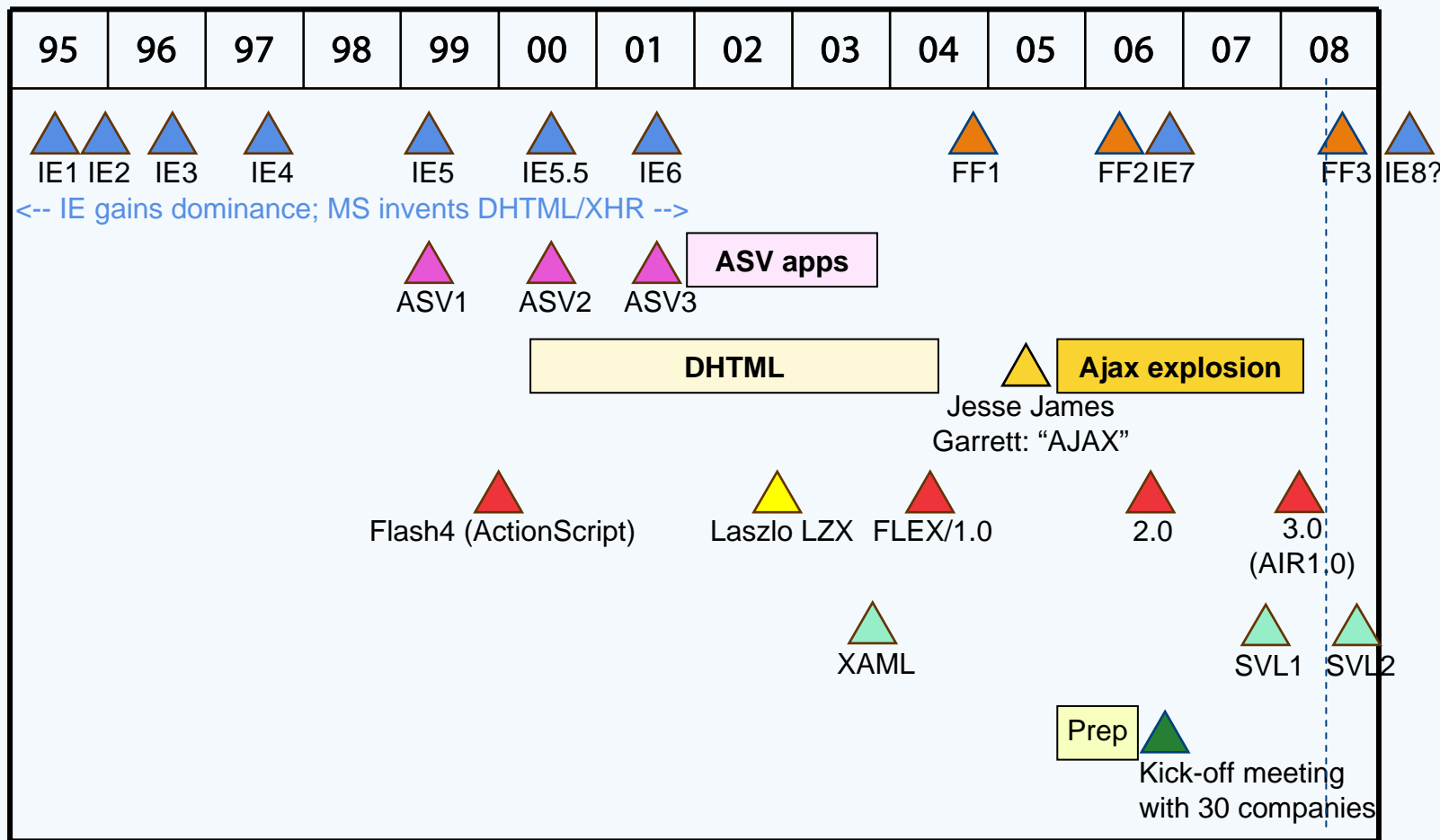
■ Bad news

- SVG Tiny is available on a large percentage of cell phones, but only some phones view SVG from the Web (i.e., limited browser integration)
- SVG Tiny 1.2 (with scripting) not yet approved
- Marketing shortcomings: little awareness of SVG Tiny as an option
- Desktop technologies (e.g., full Web browsers) are displacing the mobile standards (e.g., OMA browser standards) that have checklist requirements for SVG Tiny

Maturity

- Starting with 2005, major changes in the industry
- RIA platforms
 - Macromedia coins the term RIA, then says RIA=Flash
 - AJAX “discovered” in Feb. 2005
 - (Later) Microsoft adapts XAML into the Silverlight browser plugin

Desktop RIA Timeline



Agenda

- SVG from Jon's perspective
- Ajax (and OpenAjax)
- How Ajax Changes the Game

Quick background on Ajax

- **Originally “AJAX” (Jesse James Garrett, Feb. 2005)**
 - “Asynchronous JavaScript and XML”
 - Leveraging XMLHttpRequest to talk to server in background
 - Incremental screen updates (POST) instead of full page updates (GET)
- **At first, AJAX seemed too difficult for average programmer**
 - Yes, Google’s geniuses could achieve Google Suggest and Google Maps
 - But how could others achieve the engineering power and deep pockets to use these techniques
- **Almost immediately, countless AJAX toolkits appeared**
 - Commercial and open source
 - Provided nice JavaScript APIs. Took care of browser differences under the hood.

The broader meaning of “Ajax”

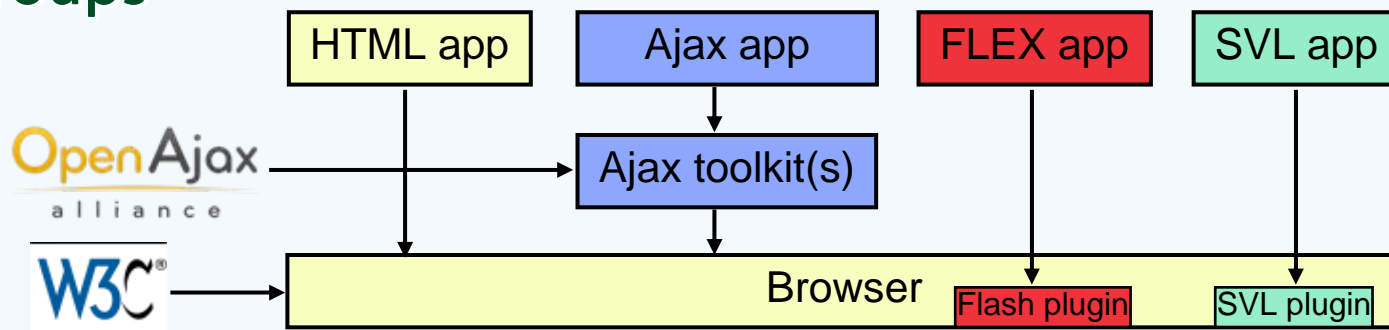
- **Ajax definition:**
 - A set of programming techniques that deliver
 - Desktop-like user interfaces
 - Within the browser
 - Using open standards formats
 - *Natively implemented in the browser*
 - *Without plugins (i.e., no Flash, Silverlight, or Java)*
- **Key technology piece is AJAX (XMLHttpRequest)**
- **But also includes many other techniques**
 - Widget libraries
 - Animation effects
 - Layout managers
 - Data binding, web services, local storage, server push, ...
- **In a nutshell - rich user interfaces using HTML and JavaScript**

The Ajax explosion (2005-2007)

- Huge amount of hype
- ~200 Ajax toolkits appear almost instantly
 - Microsoft Atlas (now called ASP.NET AJAX)
 - *Helped to contribute to legitimacy*
 - Several other major “proprietary” Ajax frameworks
 - *Adobe/Spry, Backbase, ICEsoft, Nexaweb, Sun/jMaki, TIBCO, ...*
 - Several major open source projects
 - *Prototype/Scriptaculous, Dojo, Yahoo, DWR, Google GWT, jQuery, ...*
 - Why so many so quickly? Simply rebrand DHTML to Ajax

Why OpenAjax Alliance?

- **A rich but fragile ecosystem**
 - Interoperability shortcomings
 - Educational/marketing shortcomings
 - Dependence on browsers companies for technical advances
- **Ajax vendors motivated to work together from fear of Adobe and Microsoft platform dominance in Web 2.0**
- **Ajax technology layer not addressed by other industry groups**



Membership

■ Members

- 109 members
- Big guys: Cisco, ESRI, Google, Oracle, SAP
- Ajax toolkits: Backbase, Dojo, DWR, jMaki, jQuery, Laszlo, Nexaweb
- Tools: Adobe, Eclipse, Microsoft, Sun, Tibco, WaveMaker
- Mobile: FranceTelecom/Orange, Openwave, Opera, Vodafone, Volantis
- Other standards organizations: Eclipse, HR-XML, W3C
- Ajax “users”: American Greetings, Fidelity, Gemini Systems, MITRE

OpenAjax Alliance Areas of Focus

- **Interoperability**
 - Ajax runtime libraries
 - Ajax IDEs
 - Mashups and widgets
 - Mobile Ajax
- **Marketing, education and evangelism**
- **Future browsers**

IDE Interoperability

- IDEs = Developer tools (code assist, debug, visual layout)
- Problem
 - NxM permutation problem (~200 Ajax toolkits, ~15 Ajax IDEs)
 - Each Ajax toolkit documents their APIs and widgets in their own way
- Key deliverable: OpenAjax Metadata
 - Industry standard XML for:
 - *JavaScript APIs*
 - *UI controls*
- OpenAjax Alliance's IDE WG members
 - Adobe Dreamweaver
 - Eclipse (Aptana, JSDT, ATF)
 - Microsoft Visual Studio

Mashup and Widget Interoperability

- **Mashups = Ability to assemble pre-built components into a composite application**
- **Problems**
 - Security
 - Widget interoperability
- **Key deliverables:**
 - OpenAjax Hub 1.1
 - OpenAjax Metadata for Widgets
 - Open source widget transcoders
 - Open source mini-mashup tool

Mobile Ajax

- **Mobile Ajax = Full Ajax running on mobile phones**
- **Problems**
 - Special challenges (screen size, latency, keypad, battery, ...)
 - Special opportunities (portability, GPS, camera, voice, ...)
- **White paper: Introduction to Mobile Ajax for Developers**
- **Mobile Device APIs**
 - Open source JavaScript to access CPS, camera, address book, SMS, ...
 - Targets both existing proprietary, system-dependent APIs (via plugins) and emerging industry standards
 - Collaborating with OMTP, tracking W3C (e.g., geolocation)

Shaping the Future of Ajax Ecosystem

- **Ajax industry browser wishlist initiative**
 - April – July, 2008
 - 222 participants, including many industry leaders
 - 55 separate feature requests described and discussed on wiki
- **Top feature requests**
 - 2D Graphics
 - Security (better prevention of XSS and CSRF)
 - Improved low-level DOM hooks for visual layout
 - DOM performance
 - Rich text editing
 - Server push (Comet)
 - Video and audio

Agenda

- SVG from Jon's perspective
- Ajax (and OpenAjax)
- **How Ajax Changes the Game**

Resurrection of SVG on the desktop

- **Native SVG support in Mozilla, WebKit and Opera**
 - Opera: all of SVG 1.1, some of SVG Tiny 1.2, passes Acid3
 - WebKit: nearly all of SVG 1.1, passes Acid3
 - Mozilla: nearly all of static SVG 1.1, working hard to add animation and fonts, should soon pass Acid3
 - *Also adding SVG video and leveraging SVG features within HTML*
- **However, IE still holds veto power**
 - IE8 doesn't include SVG support
- **Ajax community has stepped up, particularly Dojo**
 - dojo.gfx implements static SVG model in JavaScript
 - Renders to SVG on most browsers, but on IE uses either VML or Silverlight, and Canvas on iPhone (some dojo.gfx features disabled)

Ajax is the horse to ride

- SVG can't compete in the platform wars by itself
- Ajax solves many problems SVG has faced
 - Ubiquitous distribution
 - Industry momentum
 - Flowable text streams
 - Native UI elements
 - Grid layout
- Ajax is, in fact, the only horse to ride
 - Ajax is the only major RIA platform that embraces open standards
 - 3 out of 4 browsers already implement SVG natively
 - Acid3 test is strong leverage

JavaScript vs XML

- Ajax guys continually explore what works best in today's browsers
- **JSON – data subset of JavaScript**
 - Datatypes: Strings, Numbers, Booleans, Arrays, Objects
 - Executable logic disallowed: no assignments, no functions
 - Root of a JSON object is either an array [] or an object {}
 - Faster than XML
- **Most Ajax toolkits use JSON notation as their “declarative markup”**
 - ```
var myData = [
 ['Apple',29.89,0.24,0.81,'9/1 12:00am'],
 ['Ext',83.81,0.28,0.34,'9/12 12:00am'],
 ['Google',71.72,0.02,0.03,'10/1 12:00am'],
 ['Microsoft',52.55,0.01,0.02,'7/4 12:00am'],
 ['Yahoo!',29.01,0.42,1.47,'5/22 12:00am']];
```

# How to increase synergy with Ajax (1)

## ■ Higher-level goals

- PostScript-like vector graphics in the browser (paths, text, images, transformations, etc.)
- Open standards based
- Support for interactivity, scripting, animation, effects
- Don't fall victim to "Marketing Myopia", where it has to be one particular technology (i.e., SVG as it stands today); instead, remember the higher-level goals

## ■ Constraints

- Microsoft still holds the SVG trump card
- Even if they ship SVG in IE9, there will still be a lot of desktops still running IE6 (36% today)

# How to increase synergy with Ajax (2)

## ■ Possible high-level strategies

- Ubiquitous SVG plugin for IE
- Wait for WebKit, Opera, Mozilla, etc. to make SVG happen on mobile:
  - *SVG will gain marketplace momentum and MS will finally surrender*
- Double-down on Ajax techniques like dojo.gfx
  - *Jon's favorite*

## ■ Proposed shorter-term tactics

- Give top priority to HTML/SVG integration
  - *SVG WG needs to work closely with SVG people on Mozilla/WebKit/Opera*
  - *Make sure the HTML5 guys look at SVG as theirs*
  - *If possible, refocus SVG WG on HTML/SVG integration questions*
- Double-down on the Ajax toolkit front
  - *We need a next-generation graphics toolkit beyond what is in dojo.gfx*
  - *280 North's Cappucino showcases some interesting ideas*

# Thank you!