Modern web applications based on open standards

Huibert Aalbers
Senior Certified Software IT Architect

IT Insight podcast

- This podcast belongs to the IT Insight series
- You can subscribe to the podcast through iTunes.
- Additional material such as presentations in PDF format or white papers mentioned in the podcast can be downloaded from the IT insight section of my site at http://www.huibert-aalbers.com
- You can send questions or suggestions regarding this podcast to my personal email, huibert_aalbers@mac.com

Agenda

- A new generation of Web applications
- A technical proposal based on standards developed by the W3C
  - The new <canvas> tag in HTML 5
  - The SVG standard
- Conclusion

A new generation of web applications

- The speed at which data can be exchanged over the net in almost every corner of the world is such that it is now reasonable to create web applications designed to replace even the most complex desktop programs
- There are many companies that want to convince us that this cannot be achieved by solely relying on current W3C standards like HTML and CSS to cite just a few. They argue that proprietary technologies need to be adopted by developers in order to allow a new generation of web apps to run on current browsers
- This is simply not true.Web standards have evolved substantially over the years and today it is possible to create really sophisticated web applications based exclusively on open standards

A new generation of web applications

- It is not the goal of this presentation to cover all the relatively new standards-based technologies that allow developers to create web applications that look more and more like desktop applications
- Instead, we will focus exclusively on two technologies, the <canvas> tag and SVG, with the stated goal of generating interest among Web developers who want to create revolutionary applications and believe that this cannot be achieved by using exclusively open technologies that run on any browser

The origin of the <canvas> tag

- In 2005, Apple launched Mac OS X 10.4,Tiger. Among the new features it included was the ability to use desktop Widgets easily written in HTML and JavaScript.
- However, in order to enable the development of powerful, good looking Widgets, a new component in which applications could draw was required.That is why Apple created the <canvas> tag, originally only supported by Safari, the browser they also develop.
- Simultaneously, Apple proposed to add that new tag to the HTML 5 standard

Which browsers support <canvas>?

- The <canvas> tag is now part of the Web Applications 1.0 specification created by the WhatWG organization, also known as HTML 5, and therefore is supported by most modern browsers
  - Safari 2.0 and later
  - Firefox 1.5, and other browsers based on Gecko 1.8 and later
  - Opera 9 and later
- Despite the fact that Microsoft is part of the effort to develop HTML, no IE version currently supports this tag natively
- This problem can easily be solved using a JavaScript library

<canvas> support in IE

- Despite the fact that IE does not support the <canvas> tag, it supports VML (Vector Markup Language) an obsolete XML format used to represent vector graphics
- Neither VML nor PGML (a similar format proposed by Adobe, Sun and others) were actually adopted as W3C standards. However they both are at the origin of SVG, which supersedes them.
- Google has developed a JavaScript library that usesVML in order to implement <canvas> support on IE 5.5+
  - http://excanvas.sourceforge.net/
  - Only one line of code needs to be added to the HTML page
    - <!--[if IE]><script type="text/javascript" src="excanvas.js"></script><! [endif]-->

A simple example on how to use <canvas>

```
<html>
<body>
<canvas id="animated" width="500px" height="100px"></canvas>
<script>
var animatedctx = document.getElementById('animated').getContext('2d');
var r,g,b,alpha;

function addBox() {
  animatedctx.beginPath();

  // borra la pantalla aleatoria
  if (Math.random() > 0.99) animatedctx.clearRect(0,0,500,100);
  r = Math.round(Math.random()*255); g = Math.round(Math.random()*255);
  b = Math.round(Math.random()*255); alpha = (Math.round(Math.random()*100))/100;
  animatedctx.fillStyle='rgba('+r+','+g+','+b+','+alpha+')';

  animatedctx.rect(Math.random()*500,Math.random()*100,Math.random()*100, Math.random()*100 );
  animatedctx.fill();

  setTimeout('addBox()', 50 );
}
addBox();
</script>
</body>
</html>
```

Drawing graphics in a Canvas

- Having been created by Apple, the API designed to manipulate a Canvas is very similar to Core Graphics (Quartz), the 2D imaging API included in Cocoa and used by OS X applications
- The API features support for
  - Complex paths Regions
  - Bitmap images
  - Gradients
  - Transformations
  - Shadows

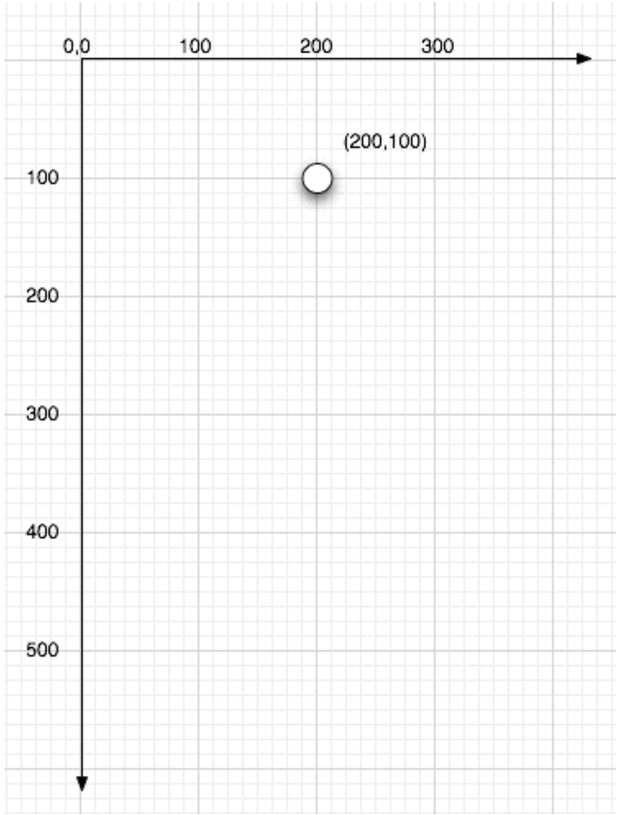
How to use <canvas>?

- The <canvas> element can be accessed using JavaScript through the DOM (Document Object Model)
- In order to easily obtain a reference to our canvas, the best way is to include a unique identifier within the tag
- The canvas object contains a “context” attribute which is an object that performs the drawing primitives

```
<html>
<body>
<canvas id="canvas1" width="500px" height="100px"></canvas>
<script>
  var context = document.getElementById('canvas1').getContext('2d');
  context.fillrect(10,10,150,100);
</script>
</body>
</html>
```

Coordinate system

- Just like with PDFs, the page coordinates are measured from the upper-left corner
- It is possible to move the origin of the coordinates
  - ctx.translate(x,y);
- The ‘sheet’ can also be rotated
  - ctx.rotate(angle);
  - Angles are measured in radians



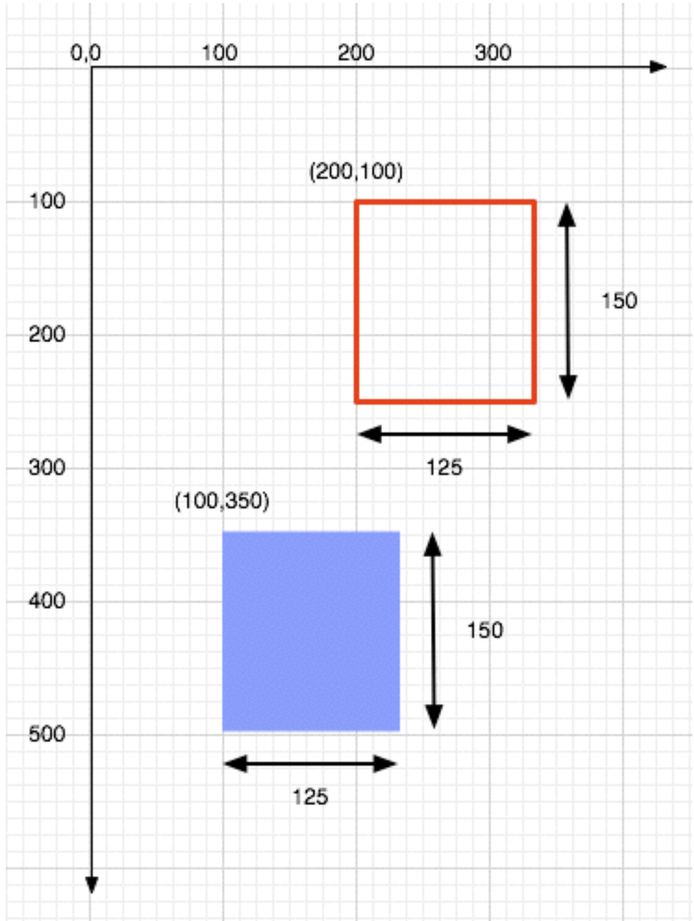
Coordinate system

```
<html>
<body>
<canvas id="canvas2" width="250px" height="250px"></canvas>

<script>
  var ctx = document.getElementById('canvas2').getContext('2d');
  var i,steps,angle,alpha;

  steps = 20;
  angle = 2*3.14/steps;
  ctx.translate(125,125);

  for (i=0;i<steps;i++) {
    ctx.beginPath();
    ctx.rotate(angle);

    alpha = (1/steps)*(i+1);
    ctx.fillStyle='rgba(255,0,0,'+alpha+')';

    ctx.rect(0,0,25,100);
    ctx.fill();
  }
</script>
</body>
</html>
```

Drawing a rectangle

- Rectangles
  - ctx.fillRect(x, y, width, height);
  - ctx.strokeRect(x, y, width, height);

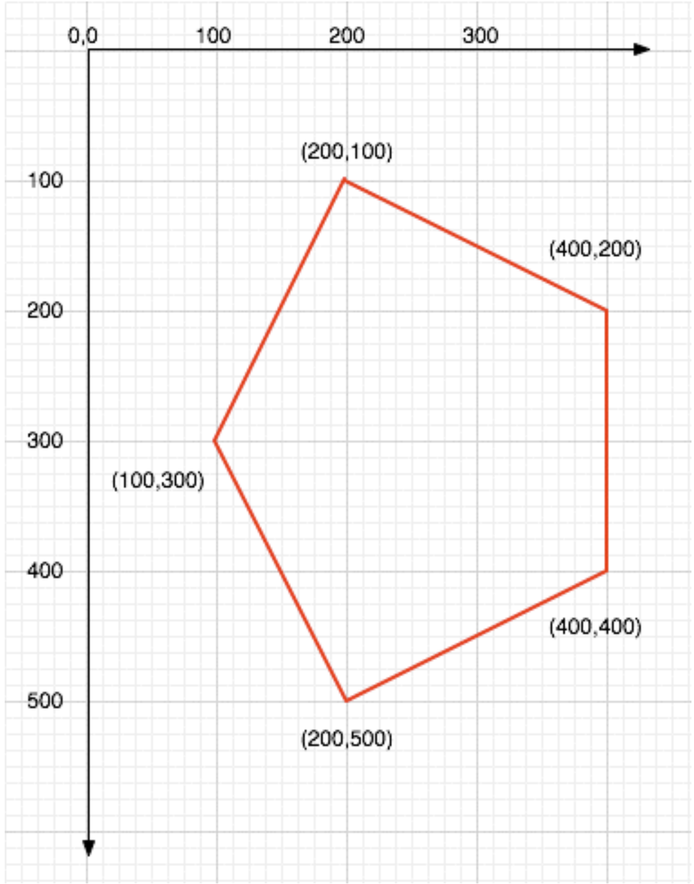
```
context.strokeStyle = ‘rgba(255,0,0,1.0);
context.strokeRect(200,100,125,150);

context.fillStyle = ‘rgba(115,135,255,1.0);
context.fillRect(100,350,125,150);
```



Drawing a polygon

- Polygons are created by drawing their frame using simple graphics primitives

```
context.beginPath();
  context.moveTo(200,100);
  context.lineTo(400,200);
  context.lineTo(400,400);
  context.lineTo(200,500);
  context.lineTo(100,300);
context.closePath();

context.stroke();
```



Drawing complex shapes

- It is also very easy to draw complex shapes using Bézier paths

```
ctx.beginPath();
  ctx.moveTo(100,100);
  ctx.lineTo(200,100);
  ctx.bezierCurveTo(200,150,250,200,300,200);
  ctx.lineTo(300,300);
  ctx.lineTo(100,300);
ctx.closePath();

ctx.stroke();
```

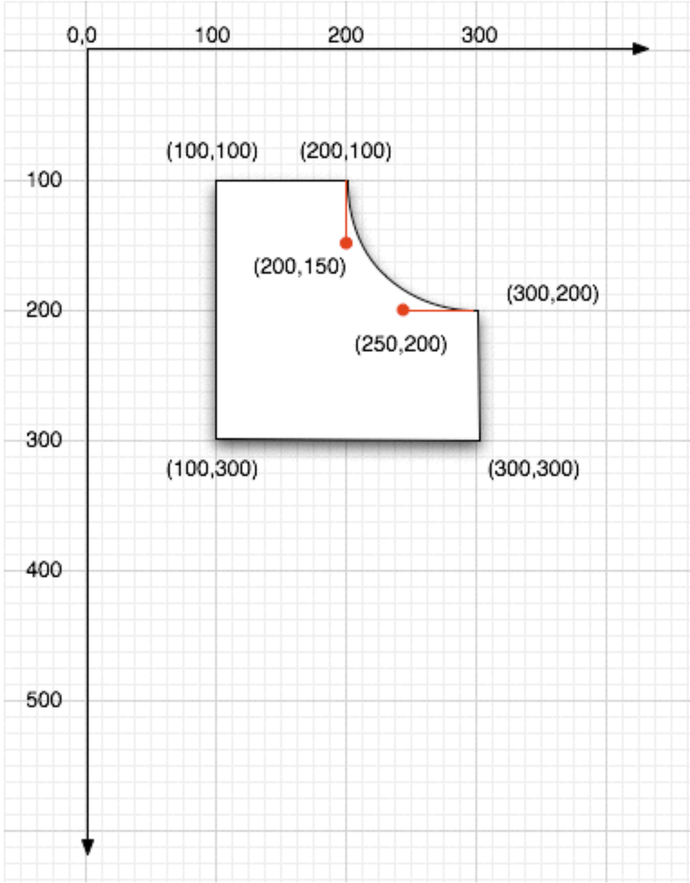
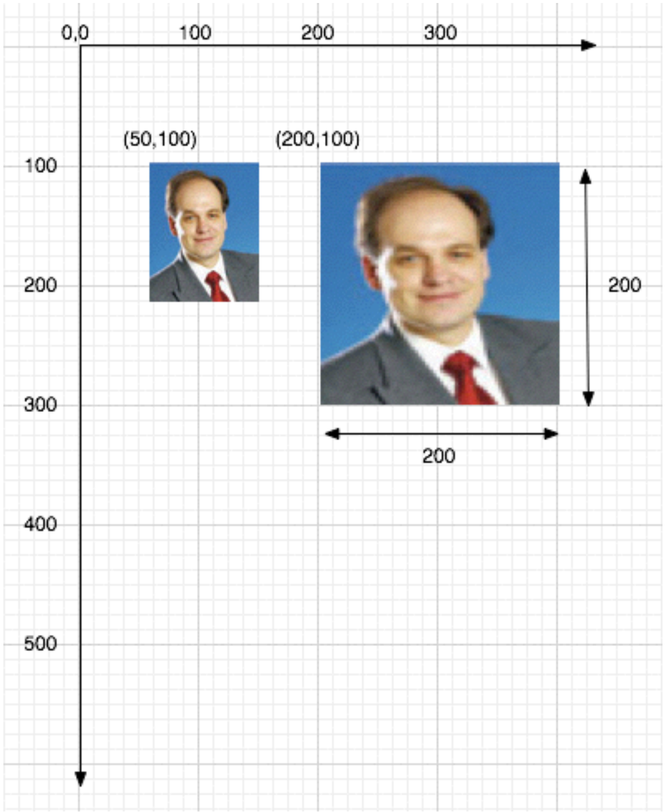


Image manipulation

- Drawing bitmap images in a canvas is extremely easy

```
var ctx = document.getElementById('canvas1').getContext('2d');
ctx.drawImage(img,50,100);
ctx.drawImage(img,200,100,300,300);
```



Where to get more information?

- Specification
  - http://www.w3c.org
  - http://developer.apple.com/documentation/AppleApplications/Reference/SafariJSRef/Classes/Canvas.html
- Examples
  - http://developer.mozilla.org/en/docs/Canvas_tutorial
  - http://www.abrahamjoffe.com.au/ben/canvascape/
  - http://the.fuchsia-design.com/2007/06/wwdc-2007-webkit-svg-demos.html
  - http://www.starxpert.fr/mozilla/demos/MouseMove/canvas.html

SVG

- SVG (Scalable Vector Graphics) is an XML standard format developed by the W3C (World Wide Web Consortium) to represent 2D vector graphics
  - SVG is the result of the standardization of two competing formats,VML (favored by Microsoft and Macromedia) and PGML (proposed by Sun Microsystems and Adobe)
  - SVG 1.0 became a W3C recommendation in September 2001
  - SVG 1.1 (the most popular so far) was introduced in 2003
  - SVG 1.2 was launched in August 2006

Which browsers support SVG?

- SVG is an emerging standard that allows to display vector graphics. Since web designers have had to use bitmap images since the invention of the web, many haven't felt the need for vector scalable graphics and therefore adoption has been relatively slow. However, there is great potential for this technology.
  - Safari 2.0.2 and later
  - Firefox 1.5, and browsers based on Gecko 1.8 and later
  - Opera 9 and later
- No IE version currently supports SVG natively
  - Fortunately, Adobe provides an IE plug-in that allows it to handle SVG documents and there is another one called Renesis

SVG

- SVG was designed to represent vector graphics through an XML file

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 20010904//EN" "http://www.w3.org/TR/2001/REC-SVG-20010904/DTD/svg10.dtd">
<svg xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink" xml:space="preserve" width="600" height="100" viewBox="0 0
1200 200">
  <!-- Pattern Definition -->
  <defs>
    <pattern id="checkerPattern" patternUnits="userSpaceOnUse" x="0" y="0" width="20" height="20" viewBox="0 0 10 10">
      <rect x="0" y="0" width="5" height="5" fill="lightblue" />
      <rect x="5" y="5" width="5" height="5" fill="lightblue" />
    </pattern>
    <linearGradient x1="10%" x2="90%" id="myFillGrad" >
      <stop offset="5%" stop-color="red" />
      <stop offset="95%" stop-color="blue" stop-opacity="0.5" />
    </linearGradient>
    <linearGradient id="myPad" spreadMethod="pad" xlink:href="#myFillGrad" />
  </defs>
  <!-- Background -->
  <rect x="0" y="0" width="100%" height="100%" fill="url(#checkerPattern)" />
  <!-- Gradient Example -->
  <rect x="20" y="20" width="1160" height="160" fill="url(#myPad)" stroke="black" stroke-width="2" />
   <line x1="136" y1="0" x2="136" y2="320" stroke="black" stroke-width="2" />
   <line x1="1080" y1="0" x2="1080" y2="320" stroke="black" stroke-width="2" />
</svg>
```

Including a SVG graphic in a HTML file

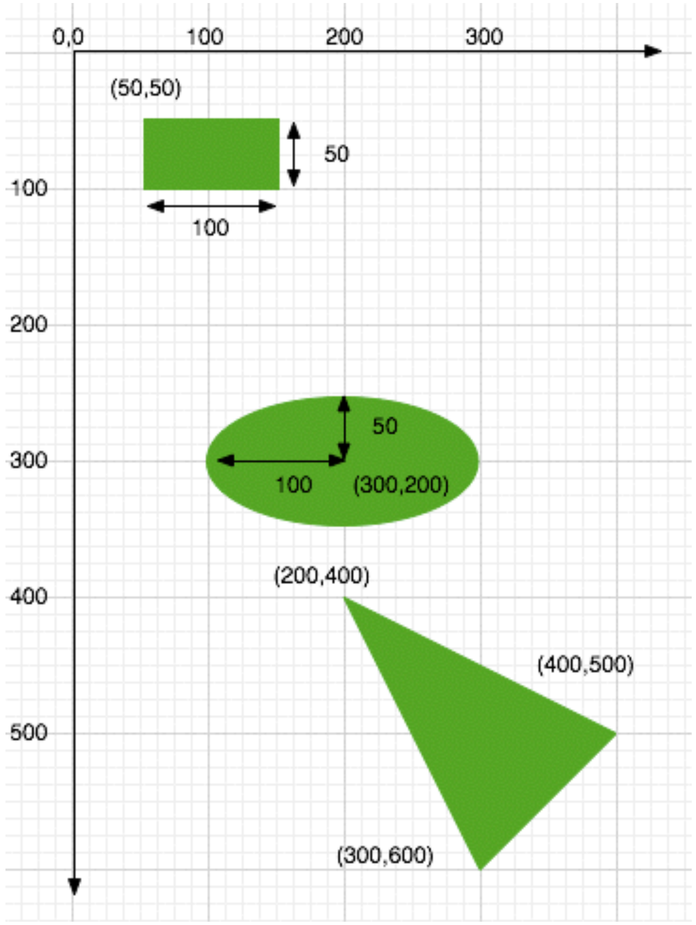
- The SVG XML data is usually kept separate from the HTML code, in its own file
- Inside the HTML file, the following code is used to import the SVG file

```
<object data="graph.svg" type="image/svg+xml" width="800" height="600">
```

Introduction to SVG

- SVG provides tags that allow to represent all the basic shapes in XML

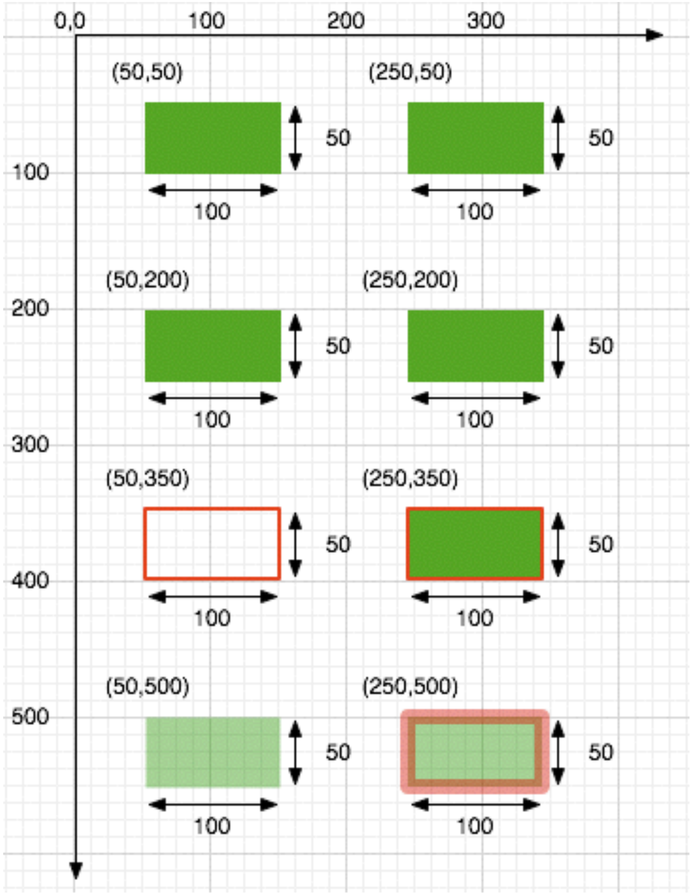
```
<rect x=”50” y=”50” width=”100” height=”50” />
<ellipse cx=”300” cy=”200” rx=”100” ry=”50” />
<polygon points=”200,400 300,600 400,500” />
```



Introduction to SVG

- SVG offers multiple ways to control how objects are draw

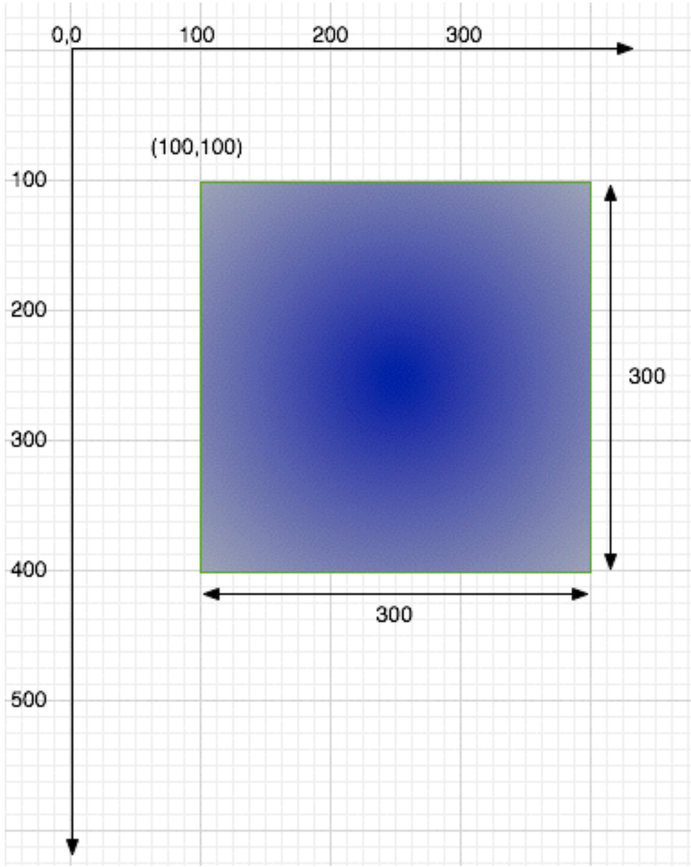
```
<rect ... fill=”green” />

<rect ... fill=”rgb(0,255,0)” />

<rect ... fill=”rgb(0,100%,0)” />

<rect ... fill=”#00FF00” />

<rect ... stroke=”#FF0000” />

<rect ... fill=”#00FF00” stroke=”#FF0000” />

<rect ... fill=”green” fill-opacity=”0.5” />

<rect ... fill=”green” fill-opacity=”0.5”  stroke=”#00FF00” stroke-width=”8” stroke-opacity=”0.5” />
```



Introducción a SVG

- SVG makes it very easy to use gradients to fill objects

```
<radialGradient id=”grad2” cx=”0.5” cy=”0.5” r=”0.5” />
  <stop offset=”0” stop-color=”#000099” />
  <stop offset=”1” stop-color=”#8C93AA” />
</radialGradient>

<rect x=”100” y=”100” width=”300” height=”300” fill=”url(#grad2)” />
```



SVG and CSS

- Just as HTML pages, SVG files can contain styles that can be applied to objects

```
<svg xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink" viewBox="0 -51 894 195"
preserveAspectRatio="xMidYMid meet">
  <title>Prueba</title>
<defs>
    <style type="text/css"><![CDATA[
      rectstyle {
        fill:blue;
        stroke:black;
        stroke-width:4;
        stroke-dasharray:10,5;
      }
    ]]></style>
</defs>

<rect width="100" height="100" class="rectstyle" />
</svg>
```

- By using pseudo classes like :hover o :focus it is possible to add basic interactive effects to vector graphics

SVG and Javascript

- An SVG file can also contain Javascript code that can be used to build interactive vector graphics
  - This works like in any HTML page where you start by navigating the DOM to find the element that needs to be modified by using any of the following methods
    - .parentNode
    - .childNodes
    - .firstChild
    - .getElementById(id)
    - .getElementsByTagNameNS(ns, nombre)
- Then you can modify the value of any of the element's attributes using any of the following methods
  - .getAttribute(nombre)
  - .getAttributeNS(ns,nombre)
  - .setAttribute(nombre,valor)
  - .setAttributeNS(ns,nombre,valor)

SVG and Javascript

- This simple function creates a rectangle

```
function crearRectangulo (x, y, ancho, alto) {
   var rectangulo = document.createElementNS('http://www/w3/org/2000/svg,'rect');

   rectangulo.setAttribute('x', x);
   rectangulo.setAttribute('y', y);
   rectangulo.setAttribute('width', ancho);
   rectangulo.setAttribute('height', alto);

   return rectangulo;
}
```

SVG and Javascript

- Through the use of Javascript it becomes possible to add advanced interactivity to SVG graphics
  - Standard event handlers can be used within the XML file
    - onclick

```
•<rect x="20" y="40" width="50" height="30" fill="red" onclick="showEvtData(evt)" />
```

    - onmouseover
    - onmousemove
    - etc.
  - Event handlers can also be added dynamically at execution time
    - .addEventListener('click', myfunction, false);

When to use SVG or <Canvas>?

	SVG	Canvas
Static Image	✓	
Large number of objects		✓
Interactivity	✓	
Animation	✓	
Data model		✓

Frameworks

- Lets face it, the development of web applications based on the use of the latest open standards can be complex. That is why the open-source community has been working hard on new frameworks that simplify the development of ambitious web applications.
  - DOJO

    http://dojotoolkit.org
  - SproutCore

    http://www.sproutcore.com
  - Google Web Toolkit

    http://code.google.com/webtoolkit/

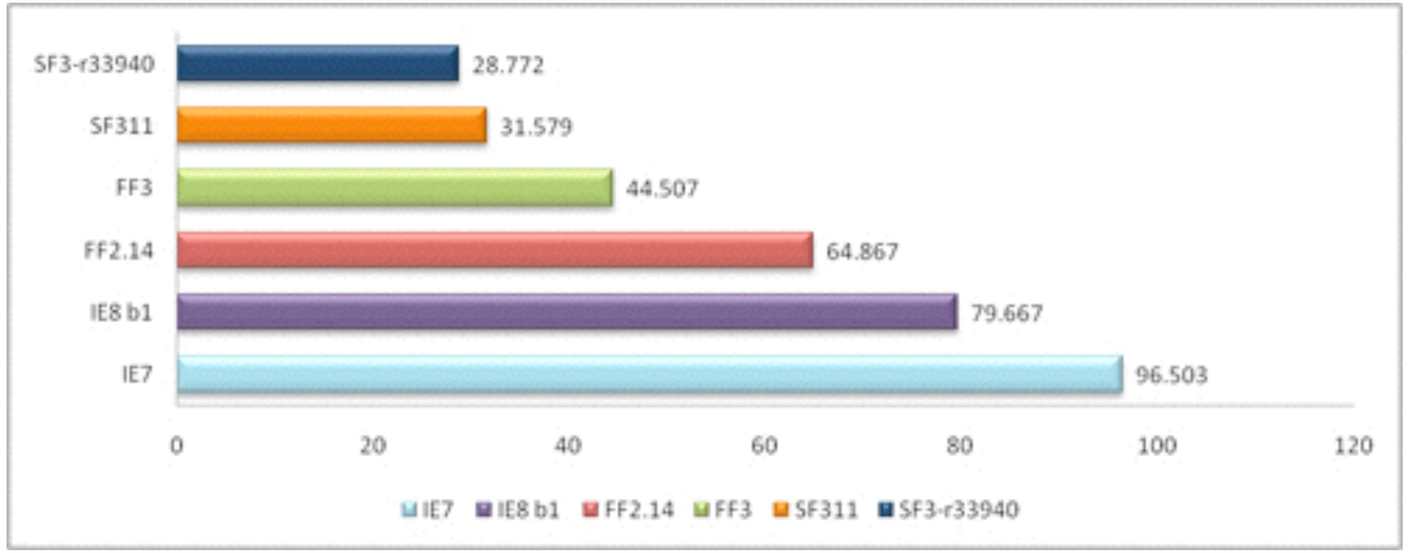
Examples

- In this presentation we have covered only a small part of the recent advances in standard web technologies. In order to understand the complete potential of the capabilities offered by modern browsers, the best way is to see some of the latest applications that use them
  - 280 Slides (Web-based PowerPoint competitor)

    http://280slides.com/
  - MobileMe (e-mail, contacts, agenda and photo gallery)
  - http://mobileme.com

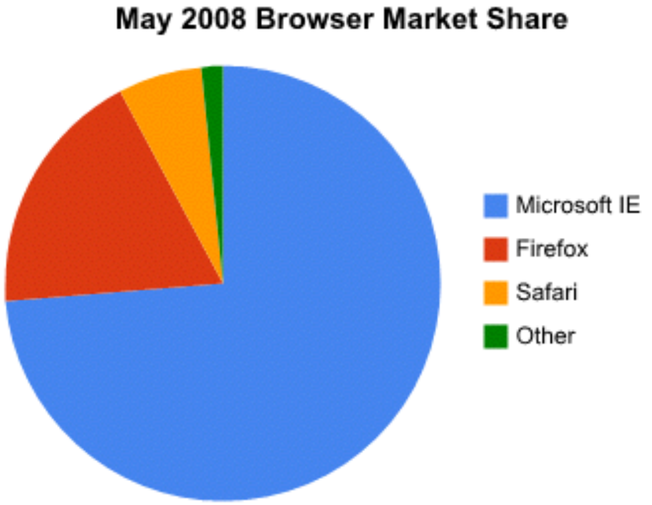
The importance of using a good browser

- The main factor affecting performance in modern web applications is the browser, and in particular the quality of its Javascript engine
- Despite what many believe, not all web browsers are created equally, the differences can be significant as shown in this chart



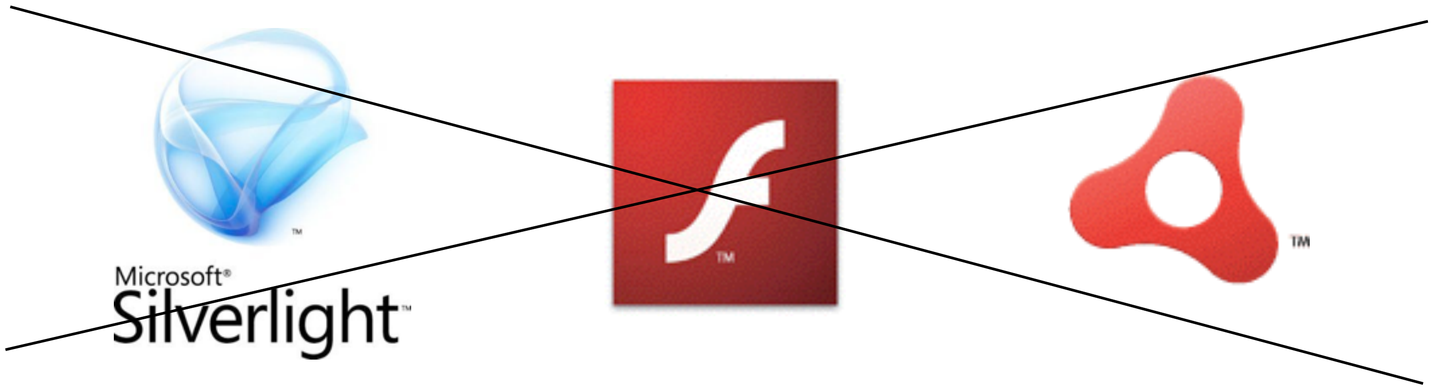
Conclusion

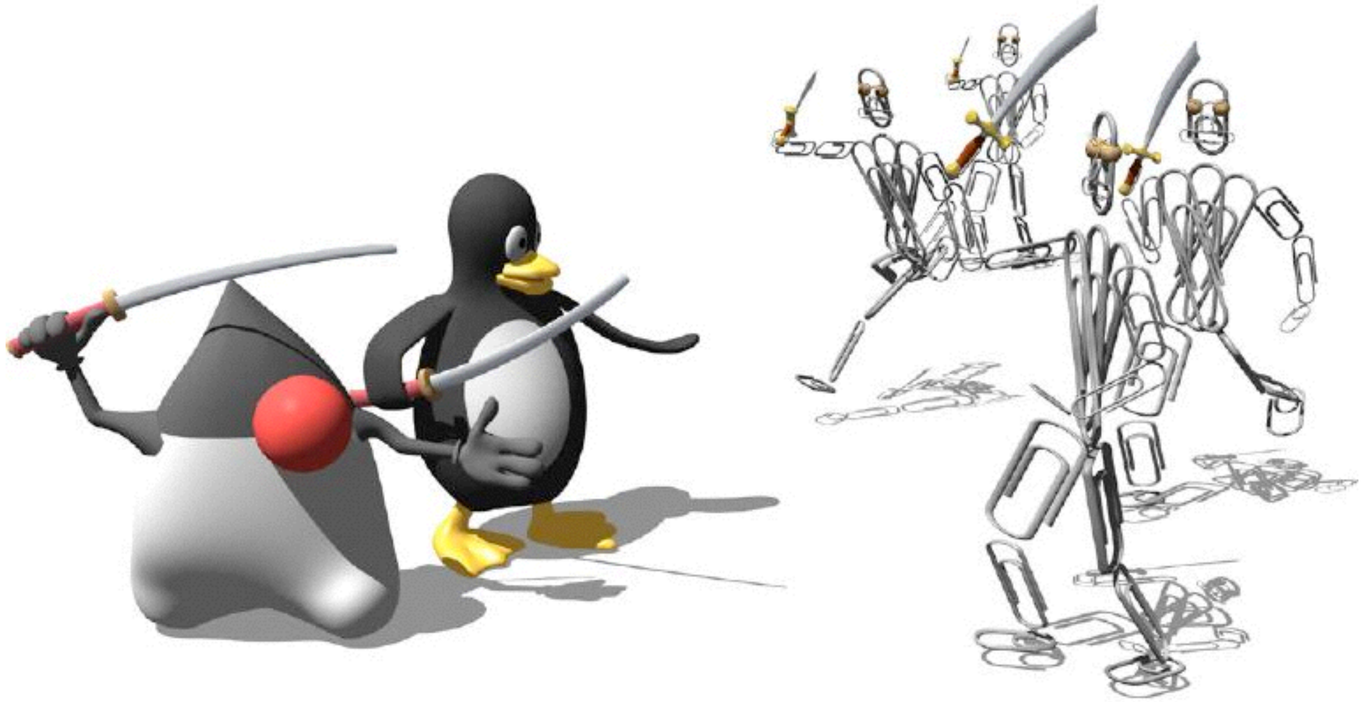
- The Internet landscape is changing quickly. The days when IE, with a market share of 95%, reigned undisputed are well behind us.The latest numbers show that IE has currently less than 75% market share and that it continues to lose adepts every day.
- As a result, developers that bet on proprietary technologies must assume the risk of losing a significant number of potential users



Conclusion

- Despite what many believe, standard web technologies have improved significantly in recent years
- This means that it is less and less necessary to turn to proprietary technologies in order to create attractive dynamic web applications
- This is good news for end-users who will be able to use a new generation of exciting web applications without having to install plug-ins





For more information, please contact me at
huibert_aalbers@mac.com