



WEBfactory 2010

Silverlight vs. HTML 5

Version 1.0

July 2012

This whitepaper is a product of the company WEBfactory GmbH.

WEBfactory GmbH
Hollergasse 15
74722 Buchen
Germany

Tel: (+49) 6281 5233 0
Fax: (+49) 6281 5233 333

<http://www.webfactory-world.de>

© Copyright 2008, WEBfactory GmbH. All rights reserved.

This whitepaper is meant to present an objective comparison between Silverlight and HTML 5. The following chapters will guide you through a confrontation of functionalities, ease of implementation and performance.

No part of this document may be copied or forwarded in any other way without the explicit written approval of WEBfactory GmbH.

All information and descriptions were selected with special care, following careful investigations. However, application errors or changes leading to a series of changed commands or other results may not be entirely excluded. WEBfactory GmbH cannot undertake any legal or other liability for such changes or any errors deriving from them.

Microsoft® and MS -Internet Explorer® are registered trademarks, property of the Microsoft Corporation. All other logos and product names are registered logos or trademarks of their respective owners.

WEBfactory GmbH denies any liability for increased costs of phone calls as a consequence of using the respective communications modules, even in the case of accidental connections.

WEBfactory GmbH has drawn up this document according to the most advanced knowledge but it does not undertake any responsibility for the programs / systems generating the results expected by the user.

WEBfactory GmbH reserves the right to make changes to the document without undertaking the responsibility of informing third parties as to such changes.

We reserve the right to make technical changes!

1. User Interface	5
1.1. HTML and XAML	5
1.2. Controls	6
1.3. Layout	6
1.4. SVG and Shapes	6
1.5. Canvas and WriteableBitmap	7
1.6. Brushes.....	7
1.7. Animations	8
2. Code	9
2.1. Languages	9
2.2. Data binding.....	10
2.3. Threading.....	10
3. Communication	11
3.1. Web requests.....	11
3.2. Web services	11
3.3. WebSockets and System.Net.Sockets	11
3.4. Isolated Storage.....	12
4. Media.....	12
5. Devices.....	13
6. Business impact	14
6.1. Developer skills.....	14
6.2. Sustainability.....	14
6.3. Ease of implementation	14
6.4. Portability	14
6.5. IP Protection	15

1. User Interface

1.1. HTML and XAML

Both HTML and XAML are markup languages and can make use of attributes. Unlike HTML 5, Silverlight allows the addition of custom attributes to existing tags, using attached properties. HTML 5 can also use custom attributes, but it will need a Javascript hack because HTML 5 ignores attributes that are not recognized.

When using Silverlight, each tag can be accessed from code behind using the tags name (x:Name). HTML 5 can offer similar functionality, accessing a tag is possible by using the tags ID. Unfortunately this is available only in certain browsers. In order to overcome this, Javascript's location service by ID can be used, but this can also slightly complicate the development.

However, HTML 5 offers a query tool in order to compensate the drawback: Selectors API. Using Selectors API, one can easily search through the objects on the page using selection queries.

Another notable difference would be the code separation. While Silverlight separates the code from markup, in HTML 5 is possible to leave the code in the HTML page or to outsource it.

Regarding the tools, Silverlight is clearly ahead of HTML 5 for now. Microsoft Visual Studio empowers Silverlight, as well as Microsoft Expression Blend, while HTML 5 has no business tool up to date. Having such powerful tools makes Silverlight development easier: XAML design is easy using Expression Blend, while debugging and having access to all services (like syntax highlights, Intellisense, etc.) is what Visual Studio puts on the table.

HTML 5 benefits from the browser development toolbars. They allow debugging and code profiling inside the browser. While having the advantage of lightweight and speed in development, the HTML 5 development toolbars for browsers are no match to Visual Studio.

1.2. Controls

When dealing with user interface, Silverlight exposes its powers: it offers the possibility to create controls from scratch or by aggregation (UserControl and CustomControl) or the possibility to inherit from basic controls or modify templates to change their appearance.

On the other hand, HTML 5 has virtually no other integrated solution for creating controls than copying and pasting the code. This represents a major drawback in the business of industrializing projects.

1.3. Layout

Silverlight has rich options for layout management. The objects can be positioned in various ways:

- In a grid with cell size in absolute, relative or even adaptable to the content
- With vertical or horizontal stacking in a StackPanel
- In a canvas that enables positioning through absolute coordinates
- With the margins and alignments
- With overflows exposed by the new control RichTextboxOverflow

This system can be used to define own layouts by adding user controls (like WrapPanel).

1.4. SVG and Shapes

Both Silverlight and HTML 5 use vector graphics (drawings composed of geometric shapes). HTML 5 uses SVG (Scalable Vector Graphics) while Silverlight uses the classes inherited from the base class Shapes. Both vector technologies have the same geometrical elements:

- Rectangle
- Ellipse
- Line
- Polyline
- Polygon
- Path (with control system: MoveTo, LineTo, ...)

In terms of transformation systems, both Silverlight and HTML 5 use matrices as transformation base. The difference can be observed only in terms of performance, but it is not notable. The only advantage the Silverlight has over HTML 5 in terms of vector drawing and transformations is the more readable code.

1.5. Canvas and WriteableBitmap

Both Silverlight and HTML 5 allow working on 2D surfaces at pixel level. In Silverlight, the two dimensional surface is called WriteableBitmap, while in HTML 5 it is called Canvas.

Both surfaces are capable of drawing basic shapes. In Silverlight, the access to the WriteableBitmap contents is made as ARGB, handling 1 integer for each color, each element occupying 8 bits. In HTML 5, the access to the canvas contents is made as RGBA, handling 4 integers per color. It is easy to observe that manipulating pixels in Silverlight is more efficient than HTML 5.

Performance wise, the test will state that Silverlight is winning, due to its improved pixel storage system. But in real world usage, considering that browsers are capped at 60 frames per second, the performance of Javascript gets very close to the Silverlight performance.

Drawing shapes on the two dimensional surfaces will have the same speed in both Silverlight and HTML 5.

1.6. Brushes

Silverlight and HTML 5 make user of brushes (used to fill interface elements with color, gradients and patterns) using the same concept: in HTML 5, the brushes can be used with CSS, on the 2D canvas or in the vector SVG. Images and videos can be used as brushes too. In Silverlight, the same concept is used, with the sole difference that other controls can be rendered as base texture for brushes (VisualBrush).

1.7. Animations

In Silverlight, animation is done by an autonomous system that has the ability to manage events, simple properties like double or integer, or even complex properties like color, thickness, etc. The control activities are executed in a storyboard which handles the timing.

The animations can be created in XAML or code, and it is possible to use functions for non-linear animations.

Managing translation animations in Silverlight is done by VisualStateManager. It cannot define animations when changing a value but only when changing the state of a control. To launch animations when a value is changed, additional code is required.

In HTML 5, animation is done manually. This implies using the `setInterval` and `setTimeout` to animate DOM elements. Though managing animations is up to the user, there are no limits in HTML 5 animations capability.

Using the new technology like CSS Animation Module Level 3, it is possible to animate CSS styles. Like always, this advanced and new technology is poorly supported by browsers. Also, by using CSS Transitions Module Level 3, it is possible to define animation on a transition (when an attribute receives a new value). It is also possible to use transition-timing-function, which means it is possible to define how animation will evolve over a time period.

In HTML 5, similar solutions can be found:

- Grids with `<table>` `<td>` `<tr>`
- New system for placing and sizing grids: Alignment Grid CSS Level 3. You will find that this mechanism is almost exactly identical to that of Silverlight (With good reason, as it is a Microsoft proposed specification)
- System integrated with CSS layout (absolute or relative placement, absolute or relative size, etc..)
- New CSS3 Flexbox system for which you can see a demo right here (if your browser supports it). This specification allows management of alignments and placement relative to boxes (yes that's exactly the same thing for XUL and it's not a coincidence)
- CSS3 Multi-column layout which is a rendering system with multi-column (example here).

- CSS3 Media Queries that define different layouts based on display sizes with an example here (besides Samuel Blanchard's had fun doing the same thing in Silverlight here).

Both technologies are able to create adaptable and scalable layouts, but, needless to say, HTML 5 layout system depends on the used browsers.

2. Code

2.1. Languages

Silverlight and HTML 5 have different approaches on languages. While Silverlight uses C# or Visual Basic, strongly typed languages with JIT (Just in Time) compiler, HTML 5 uses Javascript, which is an interpreted dynamic language. Javascript can also be compiled on the go in some browsers.

Debugging a dynamic language like Javascript proves to be more difficult, as Javascript accepts everything that is typed. It is possible, for example, to assign two different values for a variable. More, if a mistake is made in assignment writing, for example the targeted property does not exist, Javascript will create the property instead of raising an error. This can complicate debugging but also can increase the languages capabilities. With strongly typed languages, like C# and Visual Basic, it is difficult to mess up code without getting a warning.

It is important to know that the new versions of C# have adopted some features of the dynamic code (lambda expressions, dynamic).

Performance wise, Javascript is faster at calculations and executing code, but again, it's speed varies from browser to browser. Overall, the Silverlight speed is close to the average speed of Javascript, considering the fastest and slowest browsers. It is very important also to note that HTML 5 is makes its first steps, so improvements in speed should be expected.

2.2. Data binding

In Silverlight, databinding is the key feature that allows the usage of modern design patterns like Model/View/ ViewModel. Besides the advantage of modern and effective architectures, databinding opens new possibilities like defining dynamic interfaces without any line of code.

HTML 5 has no similar technology, making HTML 5 more consumer webdesign oriented, while Silverlight targets the business applications.

Even with not direct methods of databinding in HTML 5, there are workarounds: using Javascript code to manually assign values to controls and subscribe to events can create similar results to databinding. There are even some libraries (jQuery or Knockout) that will help HTML 5 in the databinding direction, but nothing comparable in terms of implementation with Silverlight.

2.3. Threading

Silverlight has good threading support via the class `System.Threading.Thread` and the provided access to the thread pool.

In Silverlight, communication with other threads is possible via calling methods and passing parameters. A large set of tools is available for managing concurrency and timing for thread control.

The conventional Silverlight threading empowered by Visual Studio and the right tools is no match for HTML 5 in modern applications development.

HTML 5 handles threading by using Web Workers, where threads are independent from the main thread, which manipulates and displays the DOM. This Web Workers communicate with the main thread via a complex messaging system. The compartmentalization and the message based communication allow the user not to think so much about shared data. The Web Workers cannot access the DOM.

Unfortunately, the Workers communications passes through strings, so the serialization of objects to JSON is needed and can be very complicated.

HTML 5 has also implemented the concept of Shared Workers, which can communicate with other scripts. Due to this fast development of HTML 5, the downsides on HTML 5 threading may soon disappear.

3. Communication

3.1. Web requests

Web Requests are requests made by the client to a server. These requests can query a remote source with no impact over the user interface. If needed, the query can be processed to impact the UI.

In Silverlight, two classes handle these requests: `HttpWebRequest` and `WebClient`.

HTML 5 uses the Javascript `XMLHttpRequest` for querying. HTTP POST is also possible using the same system.

3.2. Web services

Using web services in Silverlight (and Visual Studio) is easy. Visual Studio allows the user to add a reference. It generates the local proxy to call the web service.

In HTML 5 (Javascript), the process of using web services is more complicated due to the lack of proper tools. This means that the user must handle manually the queries and the protocol. Alternatively, AJAX framework can be used.

3.3. WebSockets and System.Net.Sockets

Sockets on TCP and Multicast are supported in Silverlight. These classes can be constrained by security measures for preventing any soft spot for Silverlight. This is a great security plus for network communication in Silverlight.

WebSocket is a new concept in HTML 5 for network communication. More evolved than TCP/HTTP, WebSocket will allow a bi-directional connection between the client and the server. Unfortunately, WebSocket is difficult to use for transient or push communication, because the HTTP protocol is stateless and not connected.

WebSocket's biggest issue is the continuous development. Because of this, the browsers either do not support this standard or block the feature. Due to some recently discovered WebSocket safety problems, some browsers may even leave the system open vulnerable.

Note that WebSocket does not stand for sockets for web. It cannot be used to connect to any TCP server.

3.4. Isolated Storage

Local data storage is made available in Silverlight using the Isolated Storage API. Isolated Storage is treated like a file system, accessible only by the application and the current user. The usage of the storage space is handled similar to that of a standard file system.

HTML 5 provides two different solutions for this matter. The first one is the localStorage, which is used to save pairs of keys/values for long term (until the browser is closed). This local storage is accessible only by the current window object.

The second solution is the sessionStorage. It can store key/value pairs for the entire session. Like the localStorage, it is only accessible by the current window object.

4. Media

Silverlight handles video and audio with not much effort. Mainly because there are no multiple codecs, so there is no need for multiple sources for a single video file. The basic video manipulation tools (Play/Pause/Stop/Seek) are available, and besides, Silverlight allows the user to use the video as brush to other controls.

Additionally, Silverlight supports smooth streaming, meaning broadcasting streams that adapt to the connection and the resource power of the client. For smooth streaming, the associated control SmoothPlayer must be used.

Regarding the digital media rights, Silverlight supports the concept of DRM (Digital Rights Management) and can deliver copyright protected content. Unfortunately this feature lacks in HTML 5, rendering the new HTML 5 technology useless in copyright media distribution scenarios.

Unfortunately for HTML 5, the support for audio and video is complicated by the use of non-standard codecs. Depending on the browser, different formats must be sent. However, HTML 5 provides the possibility to define several sources in a single video or audio tag. Even Silverlight can be added as source in case the browser cannot handle HTML 5.

The conventional controls such as Play/Pause/Stop/Seek are available in HTML 5 and are easy to implement. Also, HTML 5 is able to capture an image from the video and use it in other places of the application. This feature is not unique to HTML 5, Silverlight being able to do just the same.

5. Devices

Access to camera and microphone is provided in Silverlight. Also, printing is supported, Silverlight having control over the drawing of each printed page.

In HTML 5, the access to camera and microphone is problematic, no browser supporting the necessary specifications: Media Capture API and HTML Media Capture.

In printing, HTML 5 manages to use the method window.print. This means that even if the printing works, there is no control over the behavior of each page.

6. Business impact

6.1. Developer skills

Choosing one or another technology depends on the skills of the development team. If the developers have .NET experience, Silverlight would fit in perfectly. On the other hand, if the developers are experienced in Javascript, HTML 5 would be an easier choice.

6.2. Sustainability

Even if Silverlight is currently supported and benefits from a large set of tools, the HTML 5 technology is still at its beginnings. This means that, most probably, many tools will emerge focusing on HTML 5, thus making this technology at least as sustainable as Silverlight.

However, HTML 5 is evolving slowly compared to Silverlight. In this case, HTML 5 cannot afford to miss any feature that Silverlight has. If moving to slow, HTML 5 might be too far behind Silverlight in terms of features and tools, in a not-so-far-away future.

6.3. Ease of implementation

Regarding implementation, Silverlight is clearly the winner. Visual Studio, Expression Blend and all the debugging, profiling, frameworks and patterns deal a lot of weight in favor of Silverlight; at least for the time being.

The HTML 5 ease of implementation can evolve quickly, especially with the future versions of Visual Studio. Also, the browsers can provide great tools for debugging, profiling or analyzing code.

6.4. Portability

Both Silverlight and HTML 5 seem to have almost the same level of support. But taking a closer look, we can easily state that if the application targets Windows or Mac OS clients,

Silverlight will do just fine (so would HTML 5, depending on the browser). But if the targets are iOS, Android or other Linux/Unix based OS, HTML 5 would be the better option.

One advantage that Silverlight has over HTML 5 is that the Silverlight projects can work “outside the browser”, similar to a normal application (having elevated rights). This can give Silverlight an edge in some situations.

Considering that HTML 5 will not be totally supported until 2014, at best, developing in HTML 5 will imply writing more code while trying to solve the compatibility issues.

6.5. IP Protection

In terms of intellectual property protection, there are obfuscators for both technologies. Still, Silverlight reverse engineering is more difficult. Besides that, the XAML is not exposed, unlike HTML, which is directly sent to the browser.