Flash Vs Silverlight

Author	: Riccardo Simon Corradin
Date	: 18-09-2008
Version	: 0.1
Educational institute	: Vrije Universiteit
Study	: Computer Science - Multimedia
Coordinator	: Dr. A. Eliëns

Contents

1. Introduction

Nowadays there is a growing demand for Rich Internet Applications (RIAs). RIAs are characterized by the fact that their features and functionality resemble those of (offline) desktop applications. In contrast with traditional HTML powered online applications, RIAs offer much more user interface functionality. In this way, more intuitive and interactive user interfaces can be developed.

Realizing RIAs can be done by using different technologies. Currently, the most used technology to develop and interact with RIAs is Adobe Flex and Flash. Adobe claims that 98 percent of US users and 99.3 percent of all internet desktop users have the Adobe Flash player installed to view RIA content.

In this study, two RIA technologies (Adobe Flex/Flash and Microsoft Silverlight) will be investigated from a neutral perspective. Their strengths, weaknesses and prospects will be summarized. Finally, a conclusion will be given about which technology can be identified as the most promising one, considering current features and looking at features that will be added in the near future.

2. Rich Internet Applications

Before starting with the different technologies that realize RIAs, we will provide a summary of what RIAs are and why the demand for RIAs is high at the moment.

Initially, the internet was designed for transporting documents and information. To realize this functionality, HTML-based applications were written, since the cost of deployment was low. Furthermore, HTML has a simple architecture and is easy to learn. Since the priority was communicating between computers and accessing information across the globe, no considerable effort was spend in the improvement of a user interface.
When the internet became more popular and accessible, the need for richer user interfaces grew. As more people gained access to the internet and became aware of the possibilities the internet could offer them, new applications had to be developed. Certain complex applications are difficult to navigate through and to develop considering the static behavior of HTML. For example, users may want to change data and immediately see the changes the just made in that particular area of the page, instead of reloading the page as a whole.

The term RIA was introduced my Macromedia in 2002, but its success became visible several years later up until recently. According to [Noda and Helwig, 2005], there are five major factors why RIAs gained popularity several years after their introduction.

1. Broadband

2. Computing power shift

3. Better response to user actions
4. Leading tech companies make their move
5. Web services and SOA

The first factor refers to the fact that most RIAs require the download of an application. In 2002 a lot of internet users still used a dial-up connection to gain access to the internet. Since a lot of interactive content has to travel over a connection, the performance (as experienced by the user) was rather low. After several years the majority of internet users possessed a broadband connection, making RIAs more accessible.

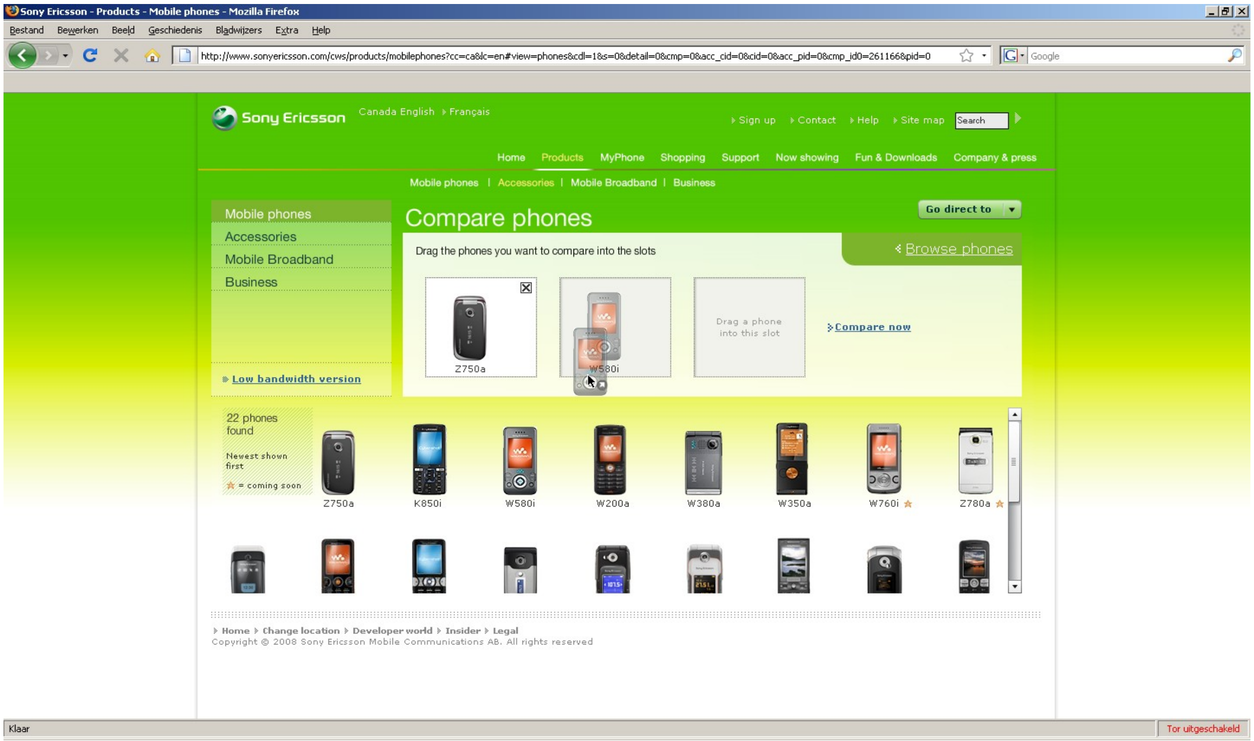
The second factor refers to the power shift from server to client. Nowadays both desktop computers and mobile devices offer enough power to successfully run RIAs. In the past, applications had to rely on the power supplied by the server.

The third factor describes the increase in demand for complex and responsive applications. Different businesses require rich applications with intuitive user interfaces. The static nature of HTML is in most cases unable to realize this.

More and more companies are aware of the benefits RIAs have to offer. Companies such as Google, Volkswagen and Sony Ericsson as described earlier use RIA technologies. The fourth factor describes this behavior.

The fifth factor refers to the fact that web services and Service Oriented Architecture (SOA) enabled a new approach of developing applications. The main idea is that presentation layers of an application are independent of the business logic layers. This new approach has also affected RIAs, since most RIA platforms consist of presentation layers on top of logic layers.

The difference between RIAs and traditional static internet applications are emphasized by an example. Two applications are considered next. The Sony Ericsson mobile phone products website and the Dutch T for Telecom website. Both are mobile phone resellers of the brand Sony Ericsson. In the screenshots below, the differences are visible.


Sony Ericsson - Products - Mobile phones - Mozilla Firefox
Bestand Bewerken Beeld Geschiedenis Bladwijzers Extra Help
Sony Ericsson
Canada English › Français
› Sign up › Contact › Help › Site map Search
Home Products MyPhone Shopping Support Now showing Fun & Downloads Company & press
Mobile phones | Accessories | Mobile Broadband | Business
Mobile phones
Accessories
Mobile Broadband
Business
» Low bandwidth version
Go direct to
« Browse phones
Z750a
W580i
» Compare now
Z750a
K850i
W580i
W200a
W380a
W350a
W760i
Z780a
› Home › Change location › Developer world › Insider › Legal
Klaar
Tor uitgeschakeld

http://www.sonyericsson.com/cws/products/mobilephones?cc=ca&lc=en#view=phones&cdl=1&s=0&detail=0&cmp=0&acc_cid=0&acc_pid=0&cmp_id0=261166&pid=0

Compare phones

Drag the phones you want to compare into the slots

Drag a phone into this slot

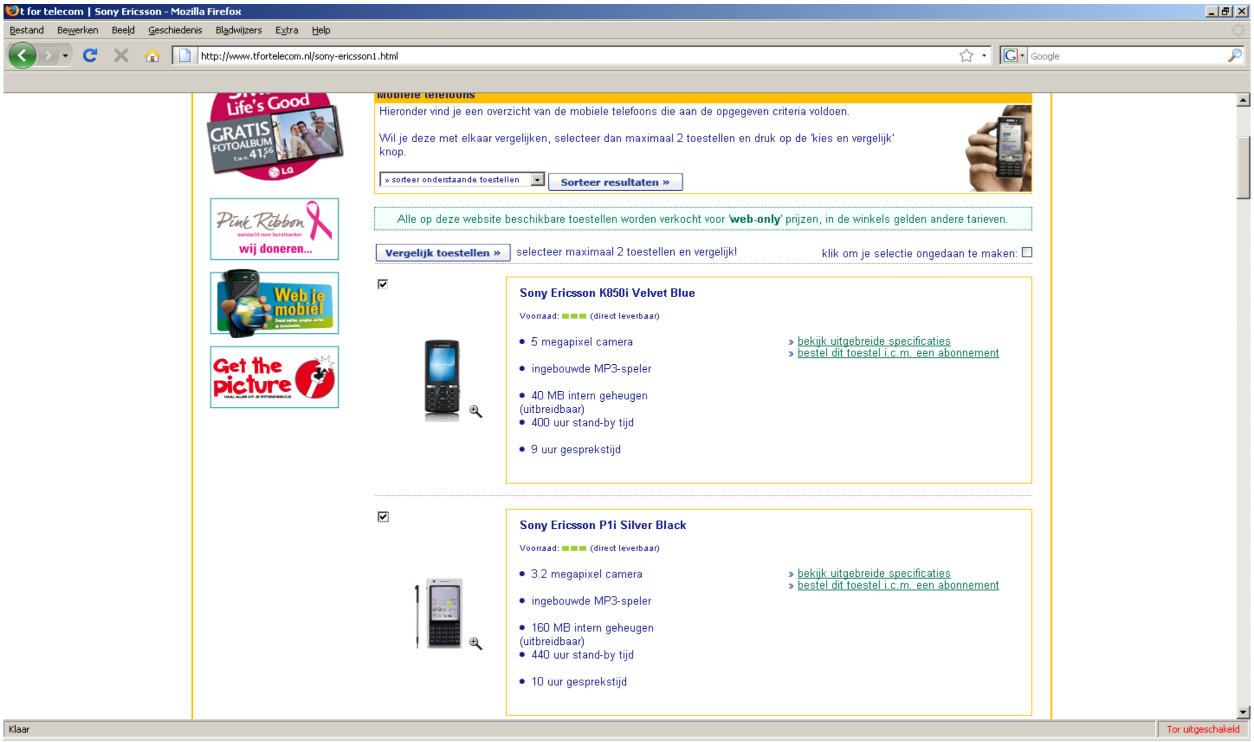
22 phones found

Newest shown first

☆ = coming soon

Copyright © 2008 Sony Ericsson Mobile Communications AB. All rights reserved

Figure 1: Sony Ericsson mobile phones


t for telecom | Sony Ericsson - Mozilla Firefox
Bestand Bewerken Beeld Geschiedenis Bladwijzers Extra Help
Life's Good
GRATIS FOTOALBUM t.w.v. 41,56
LG
Pink Ribbon
wij doneren...
Web je mobiel
Get the picture
» sorteer onderstaande toestellen
Sorteer resultaten »
Vergelijk toestellen »
» bekijk uitgebreide specificaties
» bestel dit toestel i.c.m. een abonnement
Klaar
Tor uitgeschakeld

http://www.tfortelecom.nl/sony-ericsson1.html

Mobiele telefoons

Hieronder vind je een overzicht van de mobiele telefoons die aan de opgegeven criteria voldoen.

Wil je deze met elkaar vergelijken, selecteer dan maximaal 2 toestellen en druk op de 'kies en vergelijk' knop.

Alle op deze website beschikbare toestellen worden verkocht voor '**web-only**' prijzen, in de winkels gelden andere tarieven.

selecteer maximaal 2 toestellen en vergelijk!

klik om je selectie ongedaan te maken: □

☑ **Sony Ericsson K850i Velvet Blue**

Voorraad: 🟩🟩🟩 (direct leverbaar)

- 5 megapixel camera
- ingebouwde MP3-speler
- 40 MB intern geheugen (uitbreidbaar)
- 400 uur stand-by tijd
- 9 uur gesprekstijd

☑ **Sony Ericsson P1i Silver Black**

Voorraad: 🟩🟩🟩 (direct leverbaar)

- 3.2 megapixel camera
- ingebouwde MP3-speler
- 160 MB intern geheugen (uitbreidbaar)
- 440 uur stand-by tijd
- 10 uur gesprekstijd

Figure 2: T for Telecom mobile phones

In both figures, the option to compare two cell phones to each other is displayed. In Figure 1, a user can drag different cell phones to a compare area and click on “compare now” to open a new window in which the dragged cell phones are placed next to each other for comparison. This is an example of a RIA, since the user interface has been carefully developed in order to reach a high degree of intuitionism for the user. In Figure 2, the opposite is shown. In order to compare two cell phones to each other, a user has to click once on the textbox in the upper left corner of the cell phone image. Furthermore, only up to two cell phones can be compared to each other, while the user can select more boxes. If this is done, the user will get a message that too much boxes are selected. This is less intuitive than the previous case, in which the user sees that there are three drag boxes available and cannot add more cell phones. In addition, the user does not have a complete view of all the cell phones in Figure 2. The user has to scroll down in order to navigate through different cell phones. When this is done, the user has to scroll up again to click on “Vergelijk toestellen”. In other words, using the traditional static approach means that users will spend more time in navigating through the application.

3. RIA technologies

As described earlier, different technologies can be used to realize RIAs. In this chapter Adobe Flex and Microsoft Silverlight will be studied.

3.1 Adobe Flex

Adobe Flex is a free open source framework for building rich internet applications. These applications are able to run on all major web browsers, desktops and operating systems. The Flex SDK enables an application developer to write source code for the application. This can be done in any plain text editor, but to accelerate the development of applications, Adobe advices programmers to use the Adobe Flex Builder software. This acceleration is done by using intelligent coding, interactive step-through debugging and a rich visual layout.

Adobe Flex is not the same as Adobe Flash, as the main difference is that the purpose of Flash is to design interactive content easily, while with Flex the focus lies more on the design of an application. The two technologies, however, can be coupled easily to each other, since interactive content in flash can be converted with just a couple of mouse clicks to a flex component. This component can then be loaded by Flex Builder and code can be added to use the component as part of the application.

The current version of Flex and Flex Builder is 3 and there exist two types of application deployment a programmer can write:

- Browser
- Desktop

All applications developed for the internet fall under the first type and require the Adobe Flash player to display the content correctly. This content can be viewed with a compatible web browser.

The second type includes applications designed for desktop usage. With Adobe AIR (the technology behind the development of desktop applications and formerly code-named Apollo) installed, the user can run an Adobe AIR application. This application has to be developed specifically for Adobe AIR and runs independently of a web browser. Examples of desktop applications are games, multimedia players and design programs.

Both types of solutions have their benefits and flaws. For example, desktop solutions have access to local resources and can store an unlimited amount of data. Browser solutions have restricted access to resources and usually limited storage. On the other hand, less effort is required to access a browser application from anywhere in the world, since a simple plug-in will run on most internet capable computers. No heavy system requirements are needed. A more complete list of browser versus desktop solutions is available at [BVSD].

The above discussed two types of solutions are purely software oriented. Nothing about the underlying hardware has been mentioned. Adobe offers different technologies to develop browser and desktop solutions according to the platform for which it is used.

The following two types of platforms exist:

- Fixed
- Mobile

With fixed platforms, we refer to desktop computers, notebooks, netbooks and any other computer that has a desktop oriented operating system installed. Fixed platforms are capable of running different types of web browsers and usually are more powerful than mobile solutions. Next to the earlier discussed Adobe AIR, Adobe offers Flash Player 9 for many different web browsers and operating systems. Adobe Flex can be used to develop Flash or AIR applications for fixed platforms.

Mobile platforms on the other hand include cell phones, smart phones and PDAs. These platforms usually have less powerful system specifications and are dependent on a limited amount of battery power. Developing a runtime engine for these platforms to display rich applications is a serious challenge. Adobe Flash Lite 3 currently is the latest version of a runtime engine for mobile devices. Although it offers fewer possibilities than the fixed platforms, impressive rich applications can be made that take current mobile application standards to the next level. Adobe Flash Lite can be compared to the Adobe Flash player and has to be installed on the mobile device in order to view rich applications. As with a fixed platform, these applications can be run in a browser or offline. Earlier, it was mentioned that a user can install the flash player to view browser content and Adobe AIR to view desktop content. These are both runtime engines as is Adobe Flash Lite. However, Adobe Flash Lite permits a user to view both browser and desktop content on a mobile device. I.e.: One solution instead of two separate ones. From a programmer’s point of view, Adobe Flash Creative Suite is needed to develop rich applications and Adobe Devices Central Creative Suite to preview and test a developed application for a specific mobile device.

In the figure below, the different Adobe software discussed in this section is displayed.

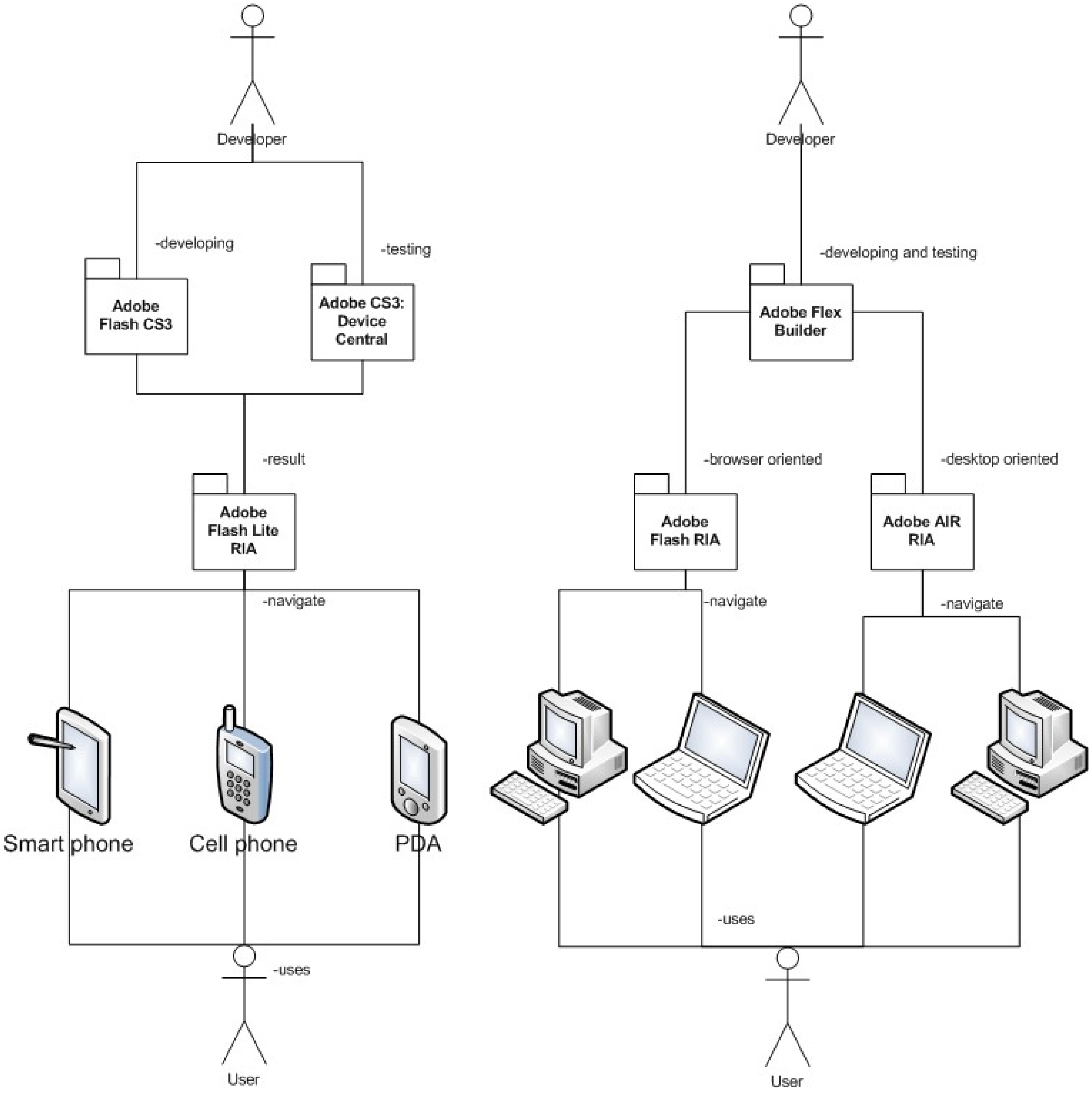


Figure 3: Relation of Adobe SW/HW

The figure is split up into two parts. The left part relates to the mobile and the right part to the fixed platform implementation. Starting with the top left, a developer for mobile RIAs has to use Adobe Flash CS3 for writing the application and the device central for testing the written application.

This will result in an Adobe Flash Lite application that can be viewed by different users with compatible mobile devices, such as a smart phone, cell phone and PDA.
The right part of the figure starts with Adobe Flex Builder as developing environment. The programmer

has to tell Flex Builder if the application to be developed is for a browser or desktop environment. When this indication has been given users can view the content and navigate through the Flash or AIR application with their computers. Note that in both cases the developed applications have to be deployed to a server, before they can be accessed by a device.

3.2 Microsoft Silverlight

In contrast to Flex, Microsoft Silverlight refers only to the interactive part of a RIA, so in a sense it can be best compared to Adobe Flash. To develop a full blown RIA however, Silverlight can be coupled with ASP.NET. Currently Silverlight is still in Beta testing (Silverlight 2 Beta 2), which means that no final stable version has been released. I.e.: Using and developing Silverlight technology should be carried out at own risk. However, this does not mean that Silverlight cannot be used. Microsoft ensures that every Beta release reaches a high quality bar. For mission critical applications, developers should use other technologies that have a stable release.
Implementing Silverlight applications can be done by using Microsoft Visual Studio and the Silverlight tools plug-in or Microsoft Expression Studio. Silverlight is able to run on different operating systems (Windows and MAC OS) and browsers (Internet Explorer, Firefox and Safari). While Adobe offers a wider range of cross-platform and browser compatibility Microsoft promises that it will extend the list of the above mentioned platforms and browsers.

Unlike Adobe, the focus of Microsoft Silverlight lies on web browsers. This does not mean that a Silverlight application cannot be viewed on the desktop independent of a web browser. Sidewinder (by swcube) is an application framework that allows users to take a web browser application (in this case a Silverlight application) and convert it to a desktop application. The application remains a Silverlight application and contains all the properties of the browser version, but without being bound to the browser window. Furthermore functionality may be added, such as the automatic docking of an application to the side of the screen. Sidewinder
Adobe AIR is developed specifically to allow applications to use local resources and to profit from the computing power of a desktop computer. Sidewinder actually ports the Silverlight application to the desktop environment. Since the browser application is not written for a desktop environment, neither is the ported version of the browser application. Sidewinder is actually an approach in general to view web applications. These can be Silverlight applications, but also Flash or AJAX.
The windows utility Desklighter however, is specifically designed for converting Silverlight project to .exe files so that they can be viewed in a desktop environment. Although Desklighter may produce only one .exe file and is easy to use, Sidewinder offers additional desktop options that allow more customization. Both solutions however are windows only and still not as powerful as Adobe AIR with Flash.

Continuing our discussion analogous to the previous section, fixed versus mobile platform possibilities for Silverlight will be investigated. Since Silverlight is fairly new, currently only two types of plug-ins exist. The before mentioned Silverlight 2 Beta 2 and Silverlight 1.0 for mobile, which refer to fixed and mobile platforms respectively. In the figure below, the relations between the different hardware and software components of Microsoft Silverlight are shown.

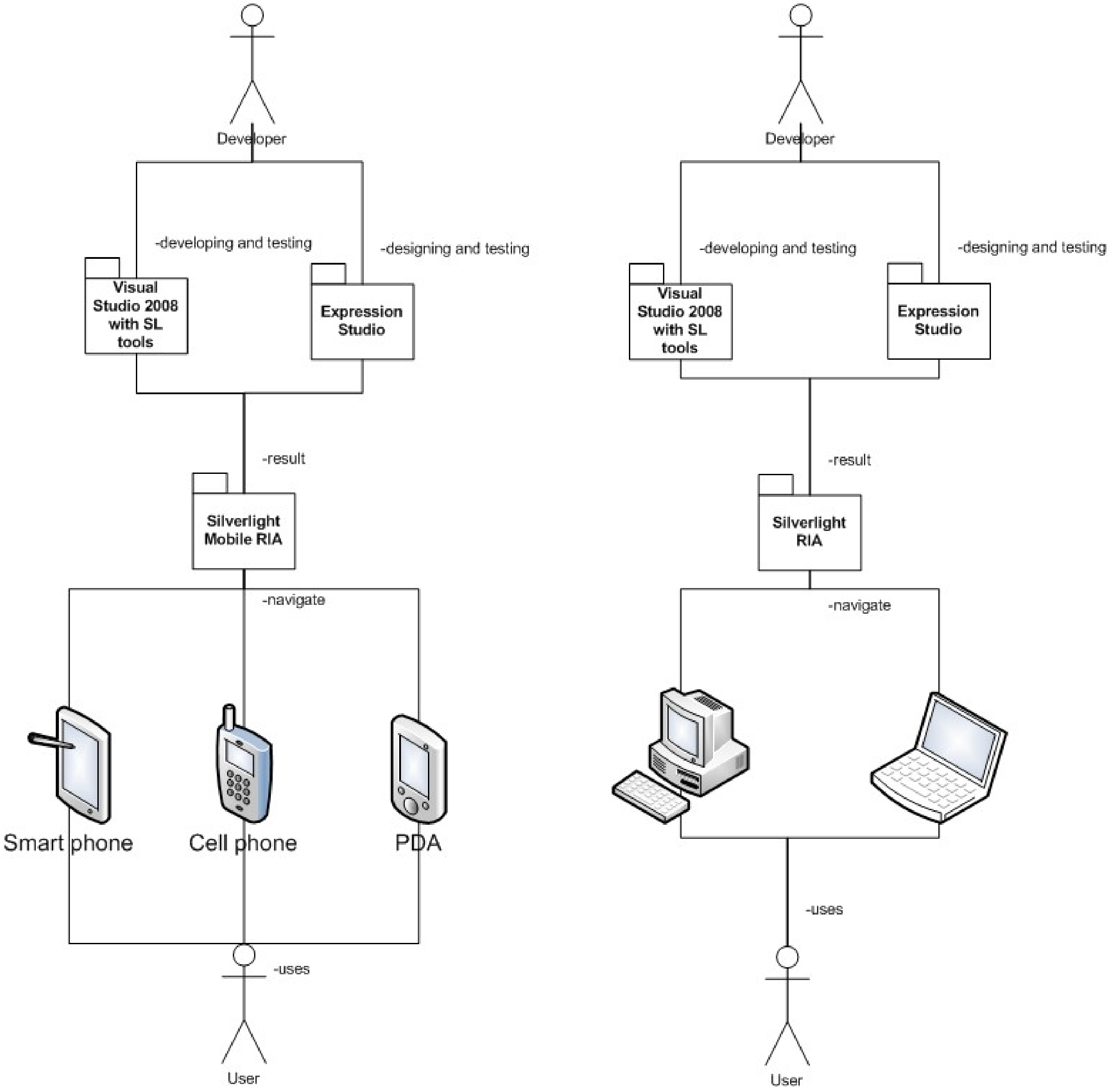


Figure 4: Relations of Microsoft SW/HW

Again, the figure has been split up into two parts. There are not many differences between a mobile and fixed application in Silverlight in terms of development environments. In both cases a developer can choose to use Visual Studio and/or Expression Studio. The main difference between the two is that Expression Studio is more focused towards the design of a RIA and Visual Studio more towards the programming part. Combining the two developing environments allows a developer to create high quality RIAs. The remainder of the figure is analogous to Figure 3 and is self explanatory.

4. Comparison

In earlier sections current Adobe and Microsoft RIA technologies have been studied and explained. In this section a comparison will be made, considering different features of each technology. In particular 3D acceleration and its possibilities within the Adobe and Microsoft environment will be studied, because this is a new feature that adds 3D environments to RIAs, making them more interactive and appealing. 3D acceleration can be divided into two types:

- 3D software acceleration
- 3D hardware acceleration

With the first type, 3D acceleration is done by a general purpose CPU (Central Processing Unit). With this type of acceleration limited 3D environments can be generated, since the CPU has not been specifically designed for heavy 3D calculations.
With second type however, a specific hardware device is allocated to make complex 3D calculations. This device is more commonly known as a graphics card and is often a separate piece of hardware on a computer. This graphics card contains a GPU (Graphics Processing Unit) which is responsible for the 3D calculations. By shifting the calculations to a separate GPU, the CPU can focus on more general tasks and the overall performance increases.
Analogous to chapter 3, the 3D acceleration feature is first discussed for Adobe and next for Microsoft.

4.1 3D acceleration and Adobe

Currently there are several 3D engines to use for designing flash environments. Since hardware acceleration is only possible from Adobe Flash Player 10 (which has currently not reached its final state), the engines that are mentioned in this section apply to 3D software acceleration. There are a lot of different 3D engines for Flash, but we will restrict our study to three well known engines as listed below.

- PaperVision3D
- Sandy 3D
- Alternativa3D

PaperVision3D is an open source Flash 3D engine and has a large community on the web. A lot of tutorials and examples exist for the beginning developer to design rich 3D environments for Flash. Furthermore, since the engine is open source, it can be used for commercial use without acquiring a license. Sandy 3D is open source as well, but the community is much smaller. Because of this, less tutorials, demos and applications are developed using Sandy 3D.

Alternativa3D is a closed source engine, so it cannot be modified or used for commercial use without a valid license. The advantage however is the advanced environments that can be created using this engine. In contrast with the previously discussed two engines, Alternativa3D has collision detection, allowing a more realistic simulation of environments. Furthermore, lighting and shadows are implemented in a more realistic way. Another big advantage is that the company Alternativa offers also different environment compatible with Flash Player 10. This means that the development for hardware 3D acceleration is active within the organization. With the previously discussed two engines this is not the case. Flash Player 10 is currently focused on fixed platforms. There are currently no indications for a mobile compatible version.

4.2 3D acceleration and Microsoft

With Silverlight, only 3D software acceleration is supported. Microsoft has no intentions to include support for 3D hardware acceleration. Since Silverlight is fairly new compared to Flex and Flash, there are not sophisticated 3D engines yet. Silverlight actually has to prove itself first to be adult enough for RIAs before entering the 3D market. There are however two simple engines available that simulate the effect of 3D rotating shapes. The following engines exist.

- Balder
- Kit3D

Again these engines are not as sophisticated as the engines mentioned in the previous section, but show that displaying 3D content is possible with Silverlight. There is however a way to develop rich applications that use 3D hardware acceleration with Microsoft technology. XBAP (XAML Brower Application) is a full blown application running on the .NET framework of Microsoft. Rich applications can be created and loaded into a web browser with XBAP, while hardware 3D acceleration is fully supported. The downsides are that it cannot access local storage, performance does not seem to be optimal when rendering complex 3D scenes and XBAP is windows only.

As can be seen there is still no advanced solution for 3D acceleration within Silverlight that runs on most computers. As with Adobe Flash Lite, there are no indications Silverlight will add native 3D acceleration support for mobile devices.

4.3 Other features

Having discussed the possibilities of 3D acceleration in previous sections it can be concluded that Silverlight does not offer much possibilities compared to Flash. This is not surprising, since the first version of Silverlight was released in December 2006 while the first version of Flash (named FutureSplash Animator) was released in 1995. In this section, other features of Flash and Silverlight will be compared to each other.

Cross-platform
Currently, Flash supports more platforms and browsers than Silverlight. In particular, Linux is supported, which is not supported by Silverlight. However, the developers of Silverlight indicate that they are working on a release for Linux. Statistics on the website of Adobe show that in June 2008 96.5% of internet enabled PCs had Flash player 9 installed in Europe. In Japan this was 98.8%. These high percentages are related to previous versions of the Flash player. Before Silverlight, a lot of RIAs were written with Flex and the choice for installing the Flash player was a straightforward one. This led to more and more Flash RIAs and an increase in Flash player installations. The development within Adobe is also constantly evolving and an early version of Flash player 10 is available. As can be concluded, as a newcomer it is hard to compete with Adobe Flash. In Appendix A, tables are shown that list the platforms and operating systems compatible with Adobe Flash and Microsoft Silverlight. This is also done in Appendix B, but for the development environments of Adobe and Microsoft. Note that the second column in Table 6 includes the Flex SDK. I.e.: As an alternative a developer can also install the SDK and write the code in any text editor. Flex Builder, although, makes tasks like debugging and testing a lot easier.

Looking briefly at the tables in Appendix A and Appendix B, it can be seen that Adobe simply covers a broader area of PCs compared to Microsoft. For example, older PCs, which are incapable of installing the Silverlight client, still can view some Flash enabled content. This wide coverage may assist Adobe in achieving the earlier percentages.
What is not listed in the table, but is worth to mention is that Adobe supports over 350 mobile device s with Flash Lite. Microsoft currently plans to release Silverlight for Nokia's Series 60 (S60), Series 40 and Nokia internet tablets.

Learning curve
From a programmers point of view the learning curve of Flash is somewhat higher than with Silverlight. For developers that are used to an object oriented programming language, elements in ActionScript 3 (the scripting language of Flash) may not seem intuitive. Silverlight elements however can be easily defined and coupled to a main application with c# code. C# is easy to learn and is not bounded to Silverlight only, while Actionscript is to Flash.

Animation

Adobe Flash is very powerful when it comes to animation possibilities. It supports for instance motion tweening. Motion tweening is the process of generating intermediate frames between to images. This will result in a fluent transition between the start image and the end image. Silverlight unfortunately does not offer this built in functionality. There are, however, some independent developers that have added motion tweening to Silverlight, but tweening any random shape is not possible yet. Other built in animation effects of Flash include bitmap effects, such as blurring and glowing. Silverlight does not support these effects either.

Video quality

When Silverlight was introduced a strong argument for its survival was the video quality. Silverlight initially supported 720p high definition video. With the addition of the latest Flash version, a video quality of 1080p is supported.

Input

Silverlight has the possibility to read pressure sensitive input from styluses and touchpads, which is a nice little extra. Flash does not have this functionality.

Readability

The Flash format SWF (ShockWave Flash), which is used for Flash applications on the internet, is not human readable. This makes it hard for application developers to take some live examples from the web and inspect the code to see how other developers handled some similar issues.
Silverlight XAML (eXtensible Avalon Markup Language) code is perfectly human readable and code can be more easily shared.

Indexing

More importantly, readability affects the indexing of search engines. SWF not being readable makes it hard for search engines, such as Google and Yahoo, to find keywords in an SWF application. However, in [Schonfeld, 2008] it is said that Adobe has created a customized Flash player for search engines that scans each Flash application and translates the SWF content to a readable content. Unfortunately, this customized player is not able to gather the information that search engines use to rank pages. A lot still has to be done in order to successfully implement this behavior.
Silverlight does not have this problem, since the applications that are run across the web are readable by search engines and can be indexed more easily. This is because XAML looks a lot like XML and is a declarative language.

Deep Zoom

Deep Zoom is a feature of Silverlight that allows user to zoom deeply into an image. This functionality is also realizable in Flash, but the main advantage in Silverlight is that Microsoft offers a deep zoom composer. This is a program that allows developers to import images and save them as a deep zoom image or a collection that can be used to display the deep zoom image within a Silverlight application. Not much effort has to be made in order to enable deep zoom Silverlight applications.

4.4 Overview

This chapter will be concluded with a brief overview of the previously mentioned features in Flash and Silverlight. The following table lists the features and to what extent they are supported by Flash and Silverlight.

Features	**Adobe Flash**	**Microsoft Silverlight**
Cross-platform	●●●●●	●○○○○
Easy to learn	●●●○○	●●●●○
Animation	●●●●●	●●●○○
Video quality	●●●●●	●●●●○
Input	●●●○○	●●●●○
Readability	●●●○○	●●●●●
Indexing	●●●●○	●●●●●
Deep Zoom	●●●●○	●●●●●
3D effects	●●●●○	●○○○○

Table 1: Flash and Silverlight features

When summing up the amount of black dots, Flash scores 37 and Silverlight 32 points. Although this difference may seem small, the fact that Adobe Flash is much more used than Silverlight at the moment has been left out, since this will clearly be in favor of Adobe. The purpose is to give a neutral overview and compare the strengths and weaknesses of both technologies.

Conclusion

While being relatively young, Microsoft Silverlight clearly shows to be a promising technology to develop RIAs. No extra development environments are needed, since developers can use the existing Visual Studio or Expression Studio which are familiar. This lowers the learning curve to implement Silverlight RIAs and creates a worldwide pool of experts who can share their knowledge.
Despite the fact that the first Flash version dates back to the nineties, development in Adobe labs is constantly evolving resulting in new releases that keep Adobe at the first position in RIA development. Adobe Flash still remains the most powerful solution to create rich 3D if not 2D applications. With the biggest market share it is hard to believe that Silverlight will outrun Flash.
It is almost impossible to give an answer if this will happen. Silverlight shows that it can be a serious competitor, but Microsoft has still a lot of work to do for that to happen. In the end, the users will determine which technology will prevail. On the other hand, Flash and Silverlight can perfectly coexist next to each other, since a user can have both runtimes installed. The question is if this is the intention of Microsoft.

All in all it can be concluded that Silverlight is still too young and inadequate to use for developers who are bound to delivering rich multimedia content within mission critical applications. Developers new to RIAs however should definitely try to use Silverlight, since the cost of these developers to learn Silverlight is much less than learning Flash and in particular Actionscript. Bearing in mind however, that for complex 2D and 3D applications Silverlight will not be appropriate.

References

BVSD. Browser vs. desktop. 2008. http://www.adobe.com/products/air/comparison/. [accessed September 2008].

Noda, T. and S. Helwig. “Rich Internet Applications.” [2005].

Schonfeld, E. “Once Nearly Invisible To Search Engines, Flash Files Can Now Be Found And Indexed”. *TechCrunch weblog*, 30 June 2008. http://www.techcrunch.com/2008/06/30/once-nearly-invisible-to-search-engines-flash-files-can-now-be-found-and-indexed/ [accessed September 2008]

Appendix A

Adobe Flash	Microsoft Silverlight
Microsoft® Windows Vista® with Microsoft Internet Explorer 7, Firefox 2.0, AOL 9 and Safari 3.x or later	Microsoft® Windows Vista with Microsoft® Internet Explorer 7, Firefox 1.5 and Firefox 2
Microsoft Windows XP with Microsoft Internet Explorer 6.0 or later, Firefox 1.x, Firefox 2.x, Mozilla 1.x or later, Netscape 7.x or later, AOL 9, Opera 7.11 or later and Safari 3.x or later	Microsoft® Windows XP SP2 with Microsoft® Internet Explorer 6 or later, Firefox 1.5 and Firefox 2
Microsoft Windows Server® 2003 with Microsoft Internet Explorer 6.0 or later, Firefox 1.x and Firefox 2.x	Microsoft® Windows 2000 with Microsoft® Internet Explorer 7
Microsoft Windows 2000 with Microsoft Internet Explorer 5.x, Firefox 1.x, Firefox 2.x, Mozilla 1.x, Netscape 7.x or later, AOL 9 and Opera 7.11 or later	Microsoft® Windows Server 2003 (excluding IA-64) with with Microsoft® Internet Explorer 6 or later, Firefox 1.5 and Firefox 2
Microsoft Windows Millennium Edition with Microsoft Internet Explorer 5.5, Firefox 1.x, Mozilla 1.x, Netscape 7.x or later, AOL 9 and Opera 7.11 or later	
Microsoft Windows 98 with Microsoft Internet Explorer 6.0 or later, Firefox 1.x, Mozilla 1.x, Netscape 7.x or later and Opera 7.11 or later	

Table 2: Windows compatibility for Flash and Silverlight

Adobe Flash	Microsoft Silverlight
Mac OS X v10.1 or later (PowerPC) with Firefox 1.x, Mozilla 1.x, Netscape 7.x or later, AOL for Mac OS X, Opera 6 and Safari 1.x or later	Mac OS 10.4.8+ (PowerPC) with Firefox 1.5, Firefox 2 and Safari
Mac OS X v10.4.x or later (Intel-based) with Firefox 1.5.0.3 or later, Opera 6 and Safari 2.x or later	Mac OS 10.4.8+ (Intel-based) with Firefox 1.5, Firefox 2 and Safari

Table 3: MAC compatibility for Flash and Silverlight

Adobe Flash	Microsoft Silverlight
Red Hat® Enterprise Linux (RHEL) 3 update 8, RHEL 4 update 4 (AS/ES/WS) with Firefox 1.5.0.7 or later, Mozilla 1.7.x or later and SeaMonkey 1.0.5 or later	No support for Linux yet
Novell SUSE™ 9.x or 10.1 with Firefox 1.5.0.7 or later, Mozilla 1.7.x or later and SeaMonkey 1.0.5 or later	

Table 4: Linux compatibility for Flash and Silverlight

Adobe Flash	Microsoft Silverlight
Solaris 10 with Firefox 1.5.x or later and Mozilla 1.7.x or later	No known future support for the Solaris platform

Table 5: Solaris compatibility for Flash and Silverlight

Appendix B

Adobe Flex Builder	Adobe Flex SDK	Adobe AIR
Microsoft® Windows® XP with Service Pack 2 or Windows Vista® Home Premium	Windows 2000, Windows XP, or Windows Server® 2003, Java 1.4 (Sun, IBM, or BEA) or 1.5 (Sun)	Windows XP with Service Pack 2; Windows XP Tablet PC Edition; or Windows Vista® Home Premium, Business, Ultimate, or Enterprise including 64 bit editions; Microsoft® Windows® 2000 with Service Pack 4; Windows 2003 Server
Mac OS X v10.4.7–10.4.10 or 10.5	Mac OS X v10.4.x, Java 1.5 (as shipped from Apple) on PowerPC or Intel processor	Mac OS X 10.4.11 or Mac OS X 10.5.2
	Red Hat® Enterprise Linux® 3 or 4, SUSE® 10, Java 1.4 (Sun, IBM, or BEA) or 1.5 (Sun)	
	Solaris™ 9, 10, Java 1.4 or 1.5 (Sun) Compilers only	

Table 6: Supported operating systems for Adobe Flash development environments

Microsoft Visual Studio 2008 standard edition	Microsoft Expression Studio 2
Windows Vista® (x86 & x64) - all editions except Starter Edition, Windows® XP (x86 & x64) with Service Pack 2 or later - all editions except Starter Edition, Windows Server® 2003 (x86 & x64) with Service Pack 1 or later (all editions) or Windows Server 2003 R2 (x86 and x64) or later (all editions)	Microsoft® Windows® XP with Service Pack 2 or Windows Vista® operating system
	Mac OS X version 10.4 or later

Table 7: Supported operating systems for Microsoft Silverlight development environments