

Eric: Good morning, good afternoon, or good evening depending on where you are in the world and welcome to today's webcast, exploring choices for desktop and mobile RIA applications, HTML5 Dojo, Silverlight, and Flex, brought to you by Dr. Dobbs Journal, IBM, and broadcast by United Business Media LLC. I'm Eric Sherman, today's moderator. We want to make sure this event is as interactive as possible so I'd like to make a few announcements before we begin.

You can join our interactive Q&A session at the end of the event by submitting your question at any time during the webcast. Just select the question tab underneath the speaker pictures on the far left, type your question into the text area below, and then click the "submit" button. The slides will advance automatically throughout the event.

If you'd like you may download a copy of the slides by clicking on the "handouts" tab located above the presentation window. If you are experiencing any technical problems please click on the "help" tab directly to the right of the "handouts" tab. To go back to viewing the presentation simply click the "slides" tab on that same row. If you are having trouble seeing the presentation at any time please press the F5 key to refresh the page. Now on to the presentation: exploring choices for desktop and mobile RIA applications, HTML 5 Dojo, Silverlight, and Flex. Joining me today is Emmanuel Tesandia, senior technical staff member IBM France, and Christoph Jolive, advisory software engineer IBM.

Christoph: Hello everyone. This is Christoph speaking and I have Emmanuel with me and as it was just said we will present you today a session about the choices that you have when developing area applications for the web and for Mobile. So the agenda is the following: First we will have an introduction where we will define what we call advanced area applications so that we are all on the same line and that we share the same understanding of the concept we will talk about during the presentation. Secondly I will share some very quick screenshots of such applications so that we're still sharing the same view of what we will be talking about.

Then our third point we will introduce these three technologies we will discuss today, the first one being HTML 5 and Dojo, the second one being Adobe Flex, and the third one Microsoft Silverlight. Then as you can see on the agenda we will dive into the details of some of the specific parts of each of these technologies, seeing what are the pros and cons of the various technologies, and finally we will conclude with some live demos

and some insights about which technology you can choose for your RIA or project. So let's start with the definition.

Basically what we call an advanced area application is an area application where you basically want to have a desktop experience in the browser. That means that you don't want area applications with forms and grids and what you usually encounter on the web but you want something more. Typical requirements for such applications are heavy data management. That means that you want to display thousands of data. You don't just want to display a few data items but a lot of them.

A second important point is of course the responsiveness of the UI. You don't want to be blocked when you go back to the server to get some answers and things like that. You really want to deal with your application just like with a desktop application one might find. You also want advanced graphics, which means that you don't just want some raster images and colors and things like that. You want to go further. You want to have a two-toned effect. You want to have _____ and things that are quite complex to draw on the screen, and a very important point is that you want complex user interactions.

For example you don't want just a single selection but you want multiple selections. You want to be able to do complex gestures with the mouse or with the touch even if you are on mobile devices, so that's something to take into consideration. Of course you want your application to be nice looking but I would say it's not specific to advanced ARIA that it's important. You want to have animations in your application. Animations are not useful in _____ but in some cases they bring additional information. When you are, for example, moving from one data effect to another one, adding an animation that goes between the two data sets might help you to better understand the changes that are appearing in the underlying data.

Another important point is the ability to have some offline management of the application. It's more and more common to use an ARIA application over a network that can disconnect like a WiFi or a 3G network, and so you won't want to be stopped by the fact that you don't have the network available anymore, so that's something you must take into account when developing that kind of application. So let's see some screen shots now.

So here I have three examples of ARIA applications that we are building. So as you can see this is not just simple applications.

You have map components, you have diagram components, and you have gauges. You have, really, things that you usually see more on the desktop applications than on a web application, but that is what we are targeting and that's what we will be talking about today. So now that we are seeing what we will be talking about, let's describe three technologies that can answer these three requirements. What are these technologies?

The first one that we will be talking about is HTML5 and Dojo. So as you probably know, HTML 5 is an ongoing W3C specification that describes the future of HTML, and it contains very advanced features that were not in HTML before and that are really answering some of the requirements I just talked about. For example the Canvas API that allows you to draw advanced graphics in an HTML application. There is a drag/drop API, very important for desktop integration, there is the ability to do some offline storage and much more, and actually today the HTML 5 term is really encompassing more than these specifications. There are a lot of things going around HTML 5, like for example CFX 3 specifications, or the progresses of the modern browsers in terms of JavaScript engines, and all these things are sometimes encompassing the HTML 5 term and all these technologies are very important when we are building an advanced area application.

But what you are _____ that there are not enough because there are relatively low-level specifications, and when you are building application you usually need a framework. An example of such a framework that is built on top of HTML is the Dojo tool kit. That's what we will concentrate on today. There are other tool kits available but the one we are experienced with Emmanuel is the Dojo tool kit, so that's what we will be talking about.

So the Dojo tool kit, at its core, is really an abstraction of all the browsers. Even if we had HTML 5 today we still have some pretty important discrepancies between what the various browsers are supporting, and so we need an API that abstracts all of these so that we don't have to care about the low-level details. That's what Dojo is doing. It's an open source project. It is hosted by the Dojo Foundation and it's available under a dual license including a BSD license which is very good for reuse of this code. .

So Dojo contains these abstractions that I talked about, but it contains also a set of widgets that you can use for your ARIA application. I've listed some of them on the slide here like button, list, grid but you also have some more advanced widgets that are

not available in the core Dojo package but available as extension like charting. So that's the first technology we will see today.

The second one is Adobe Flex, an open source project that is developed under the control of Adobe. It's available under the MPL license and it's based on the proprietary Adobe Flash player format and API, so it is also providing some predefined components. The current version is reflected in SDK4.1 and the next version 4.5 will soon be released and it's targeting mobile devices, so it still works for web applications of course but they are focused on mobile devices for this next release.

The third technology we will talk about today is Microsoft Silverlight. In a lot of aspects it's very similar to Flex. I would say it's even more proprietary. It also provides a lot of predefined components as you can see on the slide. The latest version is Silverlight 4 and we have a release, a new release of Silverlight 5 that will be coming a little bit later this year.

So now that we have seen which technology we will talk about today, let's go into more details and take some aspects of this technology and see how they behave respectively. So the first important point when you are developing an application is of course the language that you will use to develop this application. Basically each of the solutions we've seen today have their own programming language. We will have JavaScript for HTML 5 and Dojo. You will have Action Script for Flex and Visual Basic or C# for Silverlight.

The last two languages are strongly typed languages, a loosely typed language for JavaScript. If you need some strong type language you need to take the last two ones. The first one won't do that. The last two also share the object oriented paradigm. They are both based on object paradigm concepts, while JavaScript is more based on a prototype paradigm, however if you use the framework like the Dojo framework there's a good chance that the framework will fake the object-oriented paradigm on top of JavaScript, so you might find it more useful to use such a framework if you are used to using an object-oriented language.

Another important point is that the two last languages, Action Script and Microsoft Silverlight languages would be compiled to bytecode that would be interpreted in individual machine while the JavaScript language is directly incorporated by the browser and its engine. So that means that ultimately the Silverlight and Flex languages will be a bit faster than JavaScript because you have this

bytecode, however the JavaScript engine browsers are really improving this year and so the difference isn't as big as it used to be. There are some other tricks about language execution.

For example in Flex and JavaScript you have some time limits about the number of milliseconds an action can take. If you format the thread for more than this time you will have an exception. So you are asked to be sure if you have a lengthy task to cut them in little pieces so that they can be below time limits. As I said you are usually C Sharp that is faster than Action Script that is faster than Java Script.

There are also a lot of tricks that you can apply to improve the execution speed of your web applications. For example in Action Script there are two types of array, the ones of the array class and the ones of the vector class and you have a compact array you should really use a vector array because it's far faster than a regular array. So that's something you should keep in mind. There are really facilities for you to get faster application.

Another important point after languages is the availability of the solution. As I said Dojo is really based on open standard and it's really available everywhere including on mobile devices now with all the browsers and mobile devices that are based on web kits. Adobe Flex technology is based on the Flash player so it's available most everywhere on desktop because the Flash player is almost everywhere there. There's less deployment in mobile, but still you can on mobile deploy Flex applications.

For Silverlight the picture is a bit different. You really have good deployment on Windows and Mac OS platforms but it's a bit low on the other platforms. Of course another important point is the development environment, so what you will use on a daily basis to develop your application. For HTML 5 Dojo you have several IDEs that are available. Examples are rational application developer, but you also have Aptana studio and others.

Usually the debugger in the case of HTML 5 resides in the browser. This can be Firebug for Firefox. It can be a Chrome development tool. This can be Internet Explorer 8 integrated debugger, and sometimes when you add a connection between this debugger and the IDE, but that's not the case for all IDEs so you have to be aware of that and check if you really have a good integrated debugger.

For the Adobe Flex platform there is an IDE that is called Adobe Flash Builder. It comes in two versions, the Flash Builder Standard that is described here on the slide with code view, code hinting, refactoring, and design view. It's good but it's probably better than what you have with Java Script, but it's still a bit limited by some aspects. You have a premium version of this, with unit testing and profiling. You also have some alternatives not from Adobe like the FDT IDE

The best IDEs I think with Microsoft Silverlight where you have really a very good solution with Visual Studio, and here it comes in tons of versions, from a free version Visual Studio Express to a relatively expensive version and here you really have a full solution with everything you would ever dream of for the developing application. Another point that we would like to discuss today is the third party company, because when you will be developing an application you will have the components that are available on the platform, but if you are doing a very thorough application you will soon have to rely on more advanced components that are not available on the platform.

So when you are picking a platform this is something you should look in advance. Will you be able to afford and use new components? For the Dojo HTML 5 world there are a lot of components out there but there are more real system components across systems. Basically each component is using its own way of doing things and you might have compatibility issues when trying to mix them, but you have some components. You have to be aware you might have some difficulty integrating them.

On the Adobe Flex 5 it's a big difference. Adobe is making sure there are common platforms so all the components, and there are some components provider including IBM, with IBM ILOG Elixir that gives you some more advanced components that are not available on the platform. For example in IBM ILOG Elixir we have diagram components, we have gantt components, we have maps components in calendars and things like that, that you can leverage in your application, and it would be quite easy to integrate because the framework is such that it is really built to allow third party components to come in.

It's a little bit the same in the Silverlight world with even more providers I think. Basically all of the .NET component vendors are also Silverlight referring as you can see here. IBM also has again a diagram for Silverlight that you might find useful and there's some open source components by Microsoft called the Silverlight tool

kit. So as I said I was talking about UI framework and there are very important _____ components. They are also very important for the components you might want to create yourself. To Emmanuel now to take the presentation.

Emmanuel:

So about the UI frameworks, this is something important because this is how our components are created. So in the Adobe Flex platform there is new UI framework since Flex 4, which is called Spark and Spark is a UI framework that is making a strong difference between the core logic of the component and the rendering, so the nice thing about this is that you can place any type of rendering, any type of skin on top of the components so that will allow you to create your own personalized look and feel for your company or for your application.

In the Microsoft Silverlight technology the system is very similar with the same separation between the control itself, the components itself, and what Microsoft calls the control translate, which is very similar to Flex skin and the controller template is the actual look of the component and also the behavior. In the Flex platform you would write the skin in an XML language, which is called MXML, which is the declarative way to describe the skin of the component. On the Microsoft Silverlight it's very similar. You will use an XML format called XAML, which again is a language that will allow you to describe the look of your components.

Both systems have what we call a system of binding, which is a way to connect the data to the components. For example you might want to for example connect two components together or you might also want to indicate that in some portion of a list for example you want to display some portion of your data source and this you do in a declarative way, which is something very new in those types of versions. In Silverlight the difference is maybe the binding is going one step further than what Flex is doing and you can for example do many things with binding like for example doing data conversion as you bind the data, maybe converted to get a new format that is more suitable for your components.

So in HTML 5 in the Dojo framework that we're talking today, the UI framework is called Dijit. So Dijit allows you to also make a declarative, declare your application in a declarative way to HTML as is described in the slide here. You see there is a declaration of HTML <div> which is describing a chart widget, so we use here the Dojo type parameter give the type of the dojo component that is going to be displayed in this HTML portion.

In Dojo there is no separation as it exists in Silverlight or Flex between the logic and the rendering, but there is a little bit of this through what we call HTML templates, which is an HTML portion of your components that will be rendered when your component is on the screen. So you can change with templates and make a template that is more suitable for your personal application, but it's still limited to some HTML. So as we see there is a separation. When you can do a separation between the logic and the graphics, the main goal is to have the ability to change the graphic and this requires some designers coming into play to create the graphics of your application or to create the look of your buttons, all your forms elements.

Both Flex and Silverlight provide the tools for designing this, so in your Adobe world, Adobe is providing the well known tools called Adobe Photoshop or Adobe Illustrator that will help as designer tools and very well-known designers to create any kind of sophisticated graphics. Results of these graphics in the Adobe platform can be used by a tool called Flash Catalyst. Flash Catalyst will allow you to import these graphics and create a Flex application from it, a Flex skeleton based on this graphic where you would for example indicate that this portion of the graphic is going to be a button or this portion is going to be a slider. The result of using Flash Catalyst will be finally importing into the Flash Builder which is the Flex IDE so that the developer would add the application logic on top of this graphic. So this is the tool chain for the Adobe Flex solution.

There are still some problems on this tool chain on the Flex platform because there is a round trip from Flex back to the Flash Catalyst not as good as we would expect, and the biggest problem is mainly that this Flash Catalyst tool is today only supporting the Adobe components. So if you are using any third party components you will not be able to use Flash Catalyst to skin them. That's something Adobe is working on.

On the Silverlight side and on the Microsoft side, Microsoft is providing a complete set of tools, which is called the Expression tools, so there is a specific tool for designer called Expression Designer, which is what you can compare to Adobe Illustrator even though it's less popular. This tool Expression Designer is using XAML. This XAML which represents the design of the graphics of the application can be imported into a tool called expression blend, which is a tool which will mix your design and your logic.

Finally the code done inside the IDE inside Visual Studio and there is a round trip that you can do with expression tools and Visual Studio because expression blend tools will allow you to import Visual Studio projects. Inside Visual Studio you can do a little bit of graphics, very minor graphics that you can do. It's a quite good solution. On the Dojo side there's no such tool that you would do the design of your application or your components mainly based on CSS and HTML, but no specific tool chain. That's for the components that I'll build on top of Dojo. They provide you some CSS classes that you can tailor to your particular need, so this is one way to customize the Dojo components.

One important thing that we need when we are building advanced visualization and advanced rich internet applications is the support for advanced graphics on the platform. So first talking about Dojo, Dojo has an API which is called GFX for vector graphics. It's an object oriented Java Script API that is cross-browser like all the Dojo things are. In fact what GFX is providing is a way to hide the complexity of the browsers and the different technologies that are underneath and provide you this API that you can use without really knowing what technology is used underneath. In fact underneath, SVG, VML, HTML 5 Canvas are the technologies that are really used depending on the platform.

There are still difference that exist when you are using GFX like for example when you are on Internet Explorer, you might have some restriction when using gradients because it relies on VML which has in itself some restriction on gradients, but overall it's a very convenient platform for doing graphics for the web. Some other problem that you may – well not may appear, when you are rendering a lot of data on GFX is that the rendering speed and your estimated time of your application depends on the underlying technologies. So you would get very good rendering usually when using SVG and really slower rendering speed when you are using VML on Internet Explorer, but all these things are really improving and soon Internet Explorer 9 will come with SVG support and all the main browsers will be relying on the SVG technology and it will simplify graphics for the web.

Inside Dojo itself there are many components that are relying on this API, this Dojo API, so for example the charts and the gauging of Dojo. For Flex there is also an API, which is called FXG, so it's Adobe proprietary format for 2D graphics and it provides basic 2D graphics in MS Light, rectangle, polygons, plus everything you can dream up for advanced graphics, and those elements can be

manipulated by the language action script actually. Silverlight also provides the same type of thing as part of the XAML format. Main difference is that the graphics in most of the platforms is hardware accelerated, which makes a big difference in execution speed.

For both Flex and Silverlight they have all the features of advanced graphics platforms, transformation, gradient, opacity, filter effects, everything you can dream up for advanced graphics. Of course in Flex and Silverlight these advanced graphics are the thing you are using for finding the graphics of your components, so of schemes or what might call the control paneling.

So one important thing when you're building an application is how you are going to deploy it. Flex and Silverlight have a nicer option, which is a way to create a desktop application from their technology in Flex. Not only you can create a web application but you can also create a desktop application through the technology, which is called Adobe Air. So Adobe Air is something you have to download, actually and install on your computer, and once you install you can run Adobe Air application, which is an application that you install on your local computer, but they are built on top of the same API and the same language as a normal Flex web application.

Silverlight has the same thing so you can in Silverlight 4 create a desktop application, what they call "out of browser" application, and you only have to change one option in your XML file that describes your application to get the desktop application. Some of those applications may have restrictions. In the Silverlight case there is some option to bypass some of the security restriction that you have in such a sandbox environment. For example one important thing is that by signing in the Silverlight application you will have a way to relax the cross-domain access restriction, which is something which is very important in many web applications.

The mobile deployment is always something that may be very interesting. In Dojo and HTML 5 now in most of the most important browsers on mobile, from IOS or iPhone or Android phones or Blackberry's, all those phones are running a web browser that's based on the web kit project and they can run HTML 5 and Dojo nicely and natively. The Dojo community itself is working today to improve even more the Dojo tool kit for the support of mobile by providing some specific components dedicated for mobile, and also are working on mobile enabling some of the existing components like charts or gauges and all the graphics parts.

For the Flex platform there is a Flash player that works magically on Android phones, so you will be able to also deploy your application in Android phones. It's doing an announcement for Blackberry phones and to run on the iPhone there is a solution that requires you to cross-compile your application to create a real iPhone application, but in this case it is not an application anymore. For Silverlight, Silverlight is in fact becoming the main platform for developing applications for the Windows 7 platform, which is a new platform by Microsoft, and you would do that by using Visual Studio 2010 that has an emulator and all you need to build a Windows 7 application as well as additional _____ for the support of the phone itself. So, many options here for mobile.

I'm going to go a little bit faster and a few words about concerns that are very important for internationalized _____ applications. When you are doing an application for a broader audience, a world audience, you would want to internationalize your application. You would do that in Flex by providing file bundles which is a file that contains all your local specific information, which is very similar to what Java is providing. Interesting thing is that Flex 4.1 is introducing the layoutDirection property components, which allows you to create applications with bi-directional mirroring for languages such as Arabic or Hebrew.

On the Silverlight you have the same type of internationalization feature, which would rely on XML format to create resource files. It depends on the language. You would compile them for creating DLL and providing this language-specific information as a DLL. Silverlight 4 also has this native support of bidi directional languages. For HTML 5 Dojo itself there is also strong support for internationalization with formatting what's called the common local data repository of formats for dates and numbers, currency, for any place in the world. The components provided in Dojo also support by the bidi directional language and as described in the slide, I'm not going into details, but you would create a Json in Dojo, a json file for storing your language specific information.

Accessibility also is another concern. There is a strong work on accessibility in the Dojo tool kit around the WAI-ARIA support and also keyboard accessibility. Flex and Silverlight platform also support keyboard accessibility and the ability to write support for screen readers and also values different technology depending on the platform. Going a little bit faster to give space for questions later.

So let's talk a little bit about the connection to server data. So basically all the solution they can work with all existing web technology whether it's web services, REST API, but each of the platforms they had their own specific technology. In Flex you would for example connect using the HTTP service to connect to data and parse results within XML or JSON to treat your components. There's also some specific way to connect to Java enterprise edition data sources through what's called granite data sources, Blase data source, which is an open source project, or the Adobe Life Cycle data sources which are commercial products.

For Microsoft and Silverlight there's what's called the Windows communication foundation, which is a data framework which is trying to unify various technology, not only the web technology, web services, REST technology, but also Microsoft MTS or dot net remoting technology. You have also the way to connect to data using the Microsoft LINQ technology. LINQ is a way to do native data queries inside the language, whether it's C Sharp or VB, and Link has several flavors.

One is this XML so that you can do some queries, SQL-like queries on HTML objects or link to SQL where you're writing queries in the C Sharp language and they will be executed on your SQL databases, and you get resulting objects, either C Sharp or VB objects. It's a pretty nice technology. WCF Data Services is another service provided by Microsoft, which gives access to databases through a REST-like API. A pretty useful tool. For Dojo, Dojo is providing the data API.

This is a service that is an abstraction layer that gives access to the vast majority of data. It can be JSON or it can be XML, CVS, or you can also extend the Dojo data API to connect to any kind of thing where it's your own specific data across the web. So we have now been through a lot of elements and those are technology I wanted to show you some of the components that we are building here in IBM and some example of components that you may find on the various platforms. So I hope you can see my screen. Close to the slide tab there is a little button that you can click to go in the full screen if you want to have a better view of what I'm going to show on the screen.

So first of all we have a set of Flex components here starting with some charts components. Here it's 3D, 2D _____ the type of charts for which I can play with the elevation. You will not be able to see the animation here, but you're going to see some of the

different chart types which are provided by the IBM ILOG Elixir products. These products are also providing gauges, pre-defined gauges, and also a framework for you to build your own look, engaging with your own look again with this styling and templating system that we have been talking about.

This product also is providing advent calendar with day, week, month view. Here is the week view, the month view. _____ value components which is showing you that you can through advanced Flex technology create application that are really providing the same type of user interaction as a desktop application. Other type of example here, the hitmap, which is a map with value data layered on top of it. Another hitmap here is showing some various connections of people all around the world inside a map of like data. This is an example of mixing several components from the IBM ILOG Elixir product.

Other type of components, this time this is the Silverlight platform, so this is again the chart display on the Silverlight platform available from IBM so I can actually interact with the chart data, scheduled data on my project and zoom and pan just like in a desktop application. For the Dojo example I have selected here an example of Dojo charting module, which as you can see is providing a large variety of chart types, which can be skin with different themes, and I think that completes the very fast demo I wanted to show you just so that you get the hint of the type of application that you are able to build with those platforms. Now I think we can move to questions.

Eric: All right, well thanks for that presentation and now on to the question and answer portion of our event. As a reminder, to participate in the Q&A just type your question into the text box located below the media player and then click the “submit question” button. Our first question is, “Where are some of the places to look for third party Dojo components?”

Christoph: Well there is no clear ecosystem of Dojo components out there, so there’s a lot of Dojo components inside the Dojo tool kit. You can look also in the Dojo X package of the Dojo tool kit, which is providing additional components, and then if you have questions about Dojo there’s the mailing list where you can ask question and ask if someone knows a component that would solve your issues and I’m pretty sure there will be.

Eric: All right. Another question is, “Could you go over again what browser environments the various technologies are available in?”

For example, can you run Silverlight on Mozilla or other Non-m Browsers?"

Emmanuel: Yes. Silverlight is supported on all the major web browsers on the Windows platform and also supported on the Mac platform on Safari. There is a project for supporting Silverlight on the – I couldn't find it. Sorry. On the Linux there is a project to support Silverlight on Linux but this is something, which is not done by Microsoft, so the support is not as good as it should be.

Christoph: So basically if you are talking about desktop browsers you have Dojo everywhere. You have Flash nearly everywhere because it's working on many browsers, and you have Silverlight mostly on Windows and Mac OS.

Eric: All right. Another question: "Which of these choices work well for mobile?"

Christoph: I would tend to say that at least for now the present choice for mobile would be HTML 5 and Dojo because there's a lot of deployment of IOS phones, Android phones that support that well. You will be behind with Flash and Flex because you don't have native support on IOS. I don't know if Emmanuel has another view on that.

Emmanuel: Well of course Silverlight would be used mainly for Windows phones. The nice thing with Dojo is that there is some amount of work that you can do for web application that you might be able to reuse for some of your mobile applications, I mean because it's HTML on both platforms.

Eric: Another question, "Can you comment on development time for building standard enterprise applications for new developers for the three technologies?"

Christoph: Well as I said during the presentation for example with Silverlight and Action Script you have a _____ type language and you have a compiled language, so basically you are saving some development time compared to Dojo or HTML 5, but there are also other _____. So it depends on the background of your team. If your team is coming from traditional web development with a lot of Java Script and HTML then there will be probably more productive in Dojo than in Silverlight. I would say there's no definite answer on that. There are some things that help in the Silverlight and Flex applications with doing more rapid development.

- Eric:* Someone asked if the sample applications you showed are they available on the 'net and if so could you give the URL's?
- Christoph:* The missing slide in our presentation is the slide where we reveal the information, where to get more information. There will be a follow up where we will provide additional information where to get these demos, where to get more information, what IBM is doing on those platforms.
- Eric:* So you'll be sending that to the attendees?
- Christoph:* Yes.
- Eric:* Okay. Then are all technologies suitable for real time data visualizations? For example data grids, charts with sub-second data changes? If not, do you have recommendations?
- Christoph:* I would tend to say that what would concentrate more in this area is the exchange with the server. I think on the client side at least on desktop that might be the front on mobile web because the Mobile devices are of course not as full as desktop, but on the desktop I think on the client side choice you're safe with sub-second data changes on the components on all of these three platforms, but then you are asked to be sure that your connection to the server is wide enough to support that I would say, and you would have to be careful with buying your third party components depending on the components you are buying. You have to check that. It would support that, but the technology does support that.
- Eric:* Another question then. Could you comment on the total cost of ownership of the different technologies considering things like licensing, infrastructure, availability training, learning curve for developers and the like?
- Emmanuel:* Well obviously when you're using HTML 5, Dojo, all this is open source software. You might find also pre-IDE, so you can get that ready for not a lot of money on the Dojo platform. For the – well you can also start doing Microsoft as if you were reading the presentation. You were doing Silverlight with a Visual Studios Express, which is a free IDE, but soon you will – if you are a professional developer you will have to buy a professional version of Microsoft tools and while I tend to say that it will be much more costly in terms of buying the tools and also the components, it would be much more costly on the Microsoft platform than on the Adobe Flex platform, but I mean somehow similar. It's not such a big difference anymore.

Eric: Another question. What are the advantages of Dojo compared to other widely used frameworks, for example J Query?

Emmanuel: Good question.

Christoph: I think that we have more experience on Dojo than any other framework, so it's a bit difficult for us to answer as we don't have a deep knowledge of J Query, but I'm pretty sure that Dojo has professional support around internationalization in particular with the support of the _____ database that allows you to query formatting for basically a new language in a word. Also in terms of accessibility support I think that Dojo is very powerful with respect to that.

Emmanuel: There's actually a very strong requirement of our application build out at IBM and Dojo itself is being used very widely on many IBM products. So I mean this is for sure what we can say is that Dojo has proven to be using very large deployments on IBM products. But as Christoph said we are very limited knowledge of J Query today.

Eric: All right. We're about out of time, so thank you for the presentation and the answers and thank you everyone for attending today's webcast, "Exploring choices for desktop and mobile RIA applications, HTML 5, Dojo, Silverlight, and Flex", brought to you by Dr. Dobbs Journal and IBM. Please fill out the feedback form that will open on your computer. To complete the form press the "submit answer" button at the bottom of the page. Thanks for filling out the form. Your participation in this survey helps us improve future webcasts. For more information about today's webcast please visit any of the resource links open before you.

Within the next 24 hours you will receive a personalized follow up email with details and a link to today's presentation on demand. Additionally you can view today's event on demand by visiting www.netseminar.com. This webcast is copyright 2011 by United Business Media LLC. The presentation materials are owned by or copyrighted if that is the case by Dr. Dobbs Journal and IBM who are solely responsible for their content, and the individual speakers are solely responsible for their content and their opinions. On behalf of our guests Christoph Jolive and Emmanuel Tesandia thank you for your time and have a great day.

[End of Audio]