



Choosing a technology for the SOA presentation layer

Huibert Aalbers

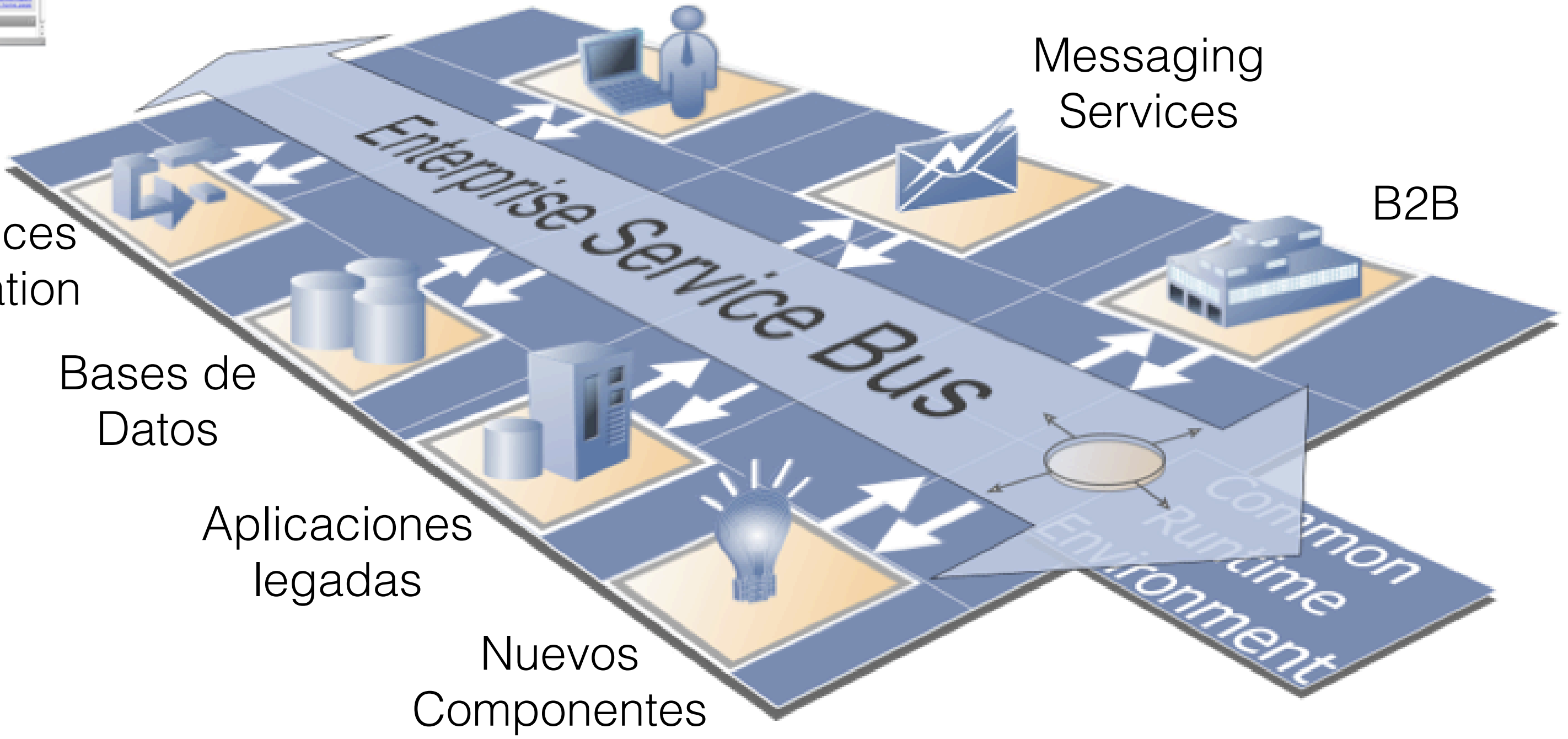
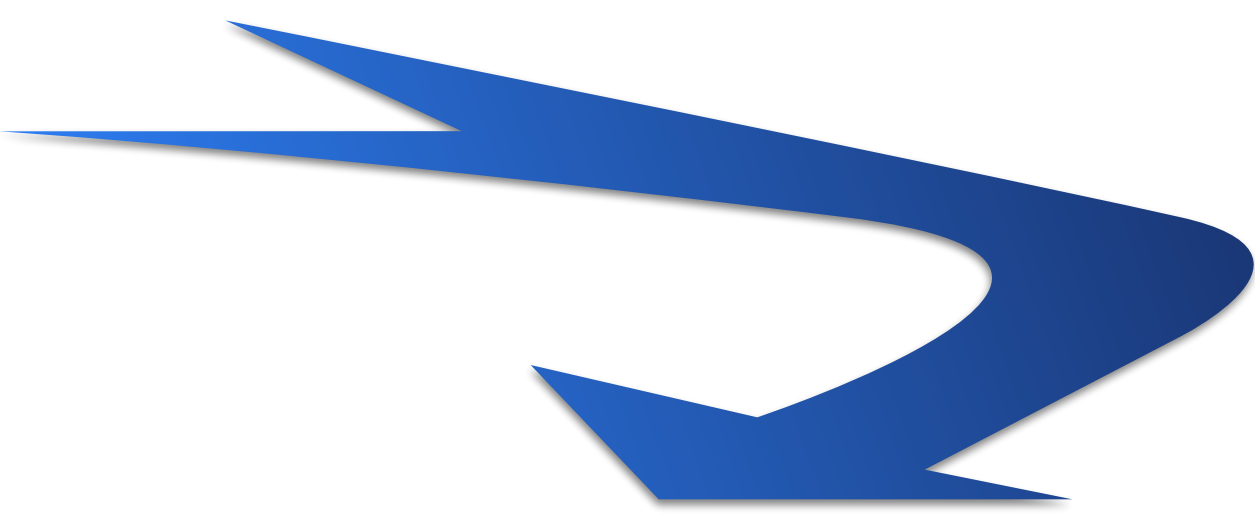
Senior Certified Software IT Architect



IT Insight podcast

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

The SOA presentation layer



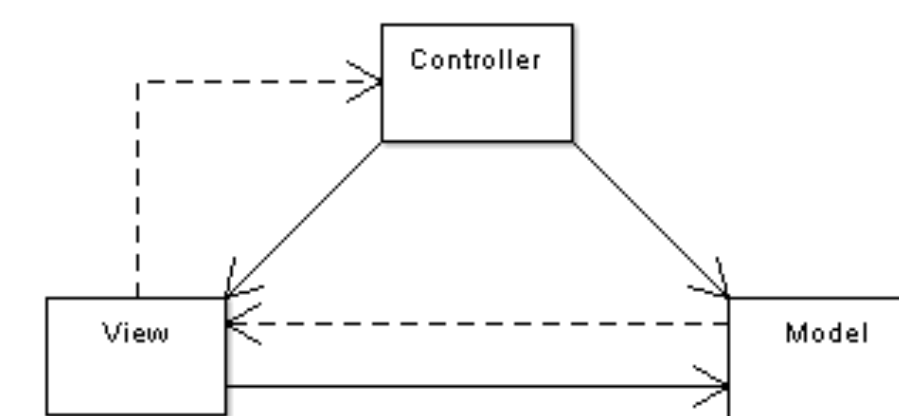


The problem

- HTML was designed to present information in an easy and efficient way
- HTML was never designed to handle complex forms or manipulate data locally in a flexible manner
 - HTML does not solve the paperless office problem, that so many organizations try to solve
 - A lot of applications, specially those which require complex user interfaces, are very difficult to move to the web
- Everyone recognizes that the SOA presentation layer should be created by graphic designers, however, because it is often so difficult to develop complex graphic interfaces, this task has to be handled too often by programmers

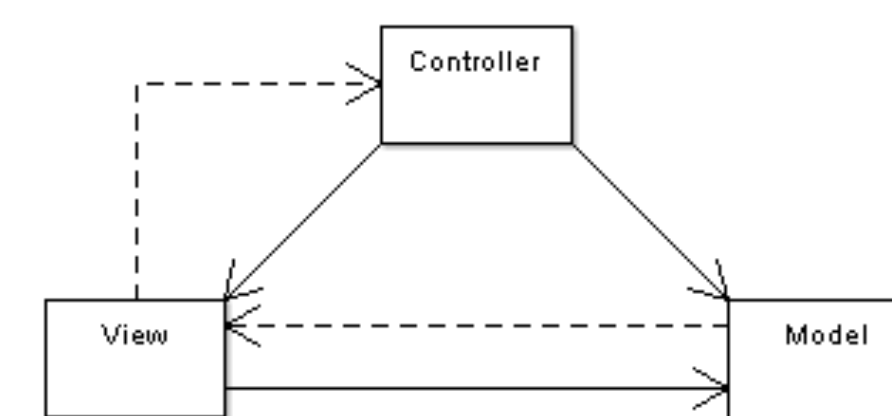
The Model-View-Controller (MVC) pattern

- In order to separate the presentation layer from the business logic layer, the Model-View-Controller pattern was developed
- It is a design pattern (some call it an architecture) that slices the applications in three types of modules
 - View - The presentation layer, usually created by graphic designers, the end-users interact with
 - Controller - The business logic layer which receives data from the View, processes it and returns the results to that layer to be presented nicely to the users
 - Model - The data representation of the information handled by the application



The Model-View-Controller (MVC) pattern

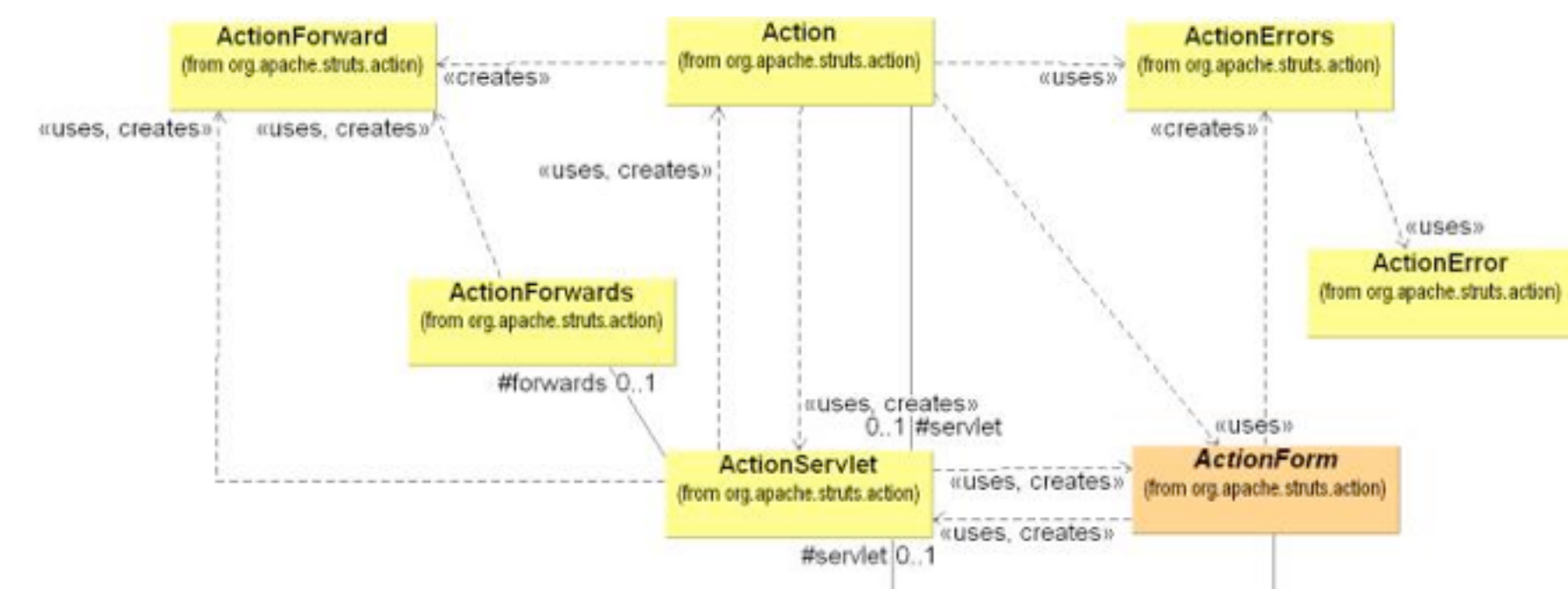
- The MVC pattern was originally developed for Smalltalk client/server and is still used in most GUI frameworks such as Cocoa (MacOS X), Swing or MFC
- The Java world has adopted the MVC pattern for Web applications
- The presentation layer (View) is represented by JSPs
- The Controller is usually implemented as a single or multiple servlets
- The data model can be implemented as components that represent objects stored inside the database (EJBs or POJOs using Hibernate, etc.)



Struts

- Struts, developed and maintained by the Apache Foundation is the most popular open-source MVC (Model 2) pattern implementation for Java
- Most Java development tools allow to create Struts based Java web applications
- However, despite its virtues, Struts is clearly a framework for developers and as such does not really simplify the job of the graphic designers who have to create complex web pages

Struts





Java Server Faces (JSF)

- JSF is a Java standard (JSR-127)
- It is a technology that simplifies the development of web pages by using reusable visual components such as tables, calendars, trees, etc. which replaces Struts
- It offers many advantages for J2EE developers
 - It was designed to integrate seamlessly with Java development tools
 - It can be used with JSPs as well as Portlets (JSR-168)
 - It spurred a dynamic market for JSF components
- JSF components are based on standard technologies which can be displayed on any modern browser and leverage technologies such as AJAX transparently



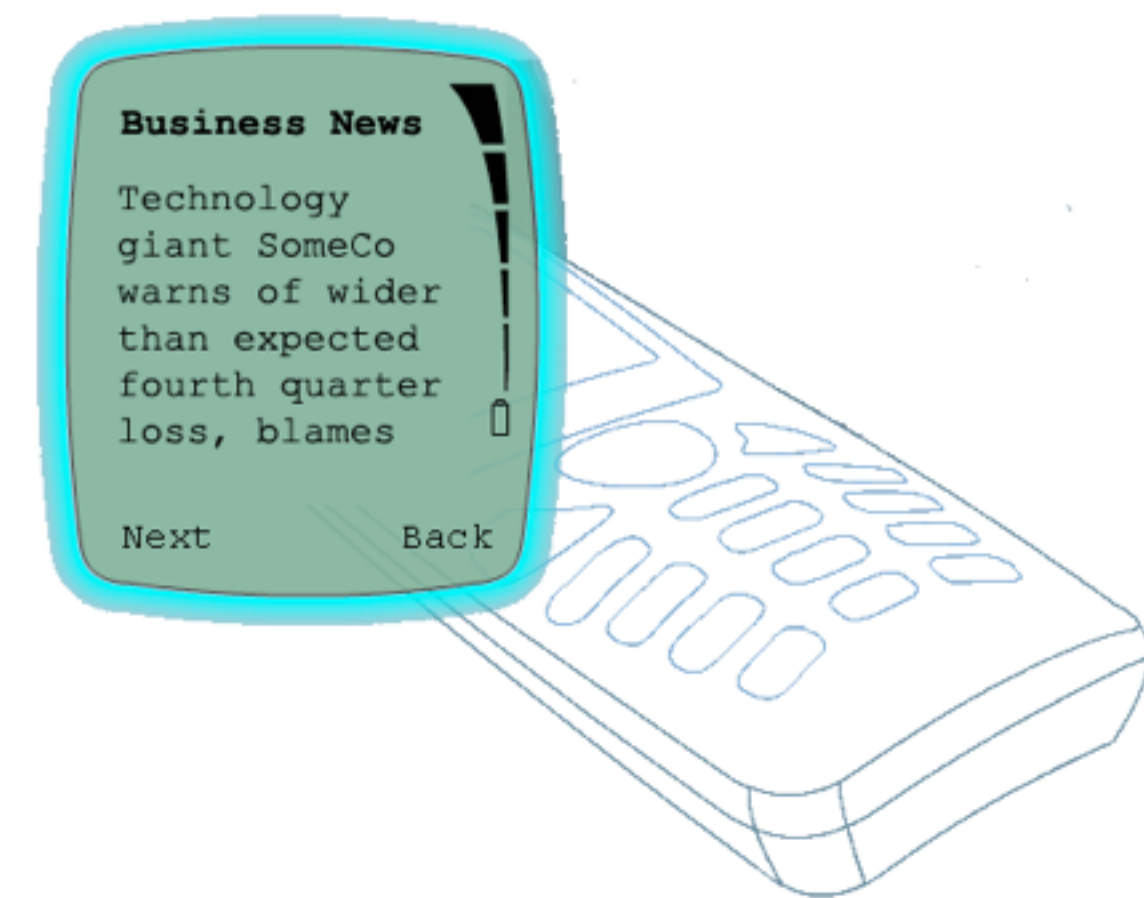
Portlets (JSR-168)

- Conceptually a portlet is very similar to a servlet. However, unlike servlets, which build a full web page, portlets only generate a page fragment
- Portlets are usually linked to a security profile, which allows administrators to easily control which users can use each portlet, based on their role
 - Simplifies the adoption of SSO (Single Sign-On) as well as customizing web pages based on user preferences
 - Supported by some application servers as well as most portal servers
 - Example: <http://www.usda.org>



Portlets (JSR-168)

- Each portlet is an independent application
 - It is developed independently and runs in a container (portal)
 - It can be linked to a security profile
 - Portlets may have different modes and states
- Portlets can be accessed by different kinds of devices
 - Cell phones, PDAs, voice
 - Each device can have its own presentation layer while sharing the business layer





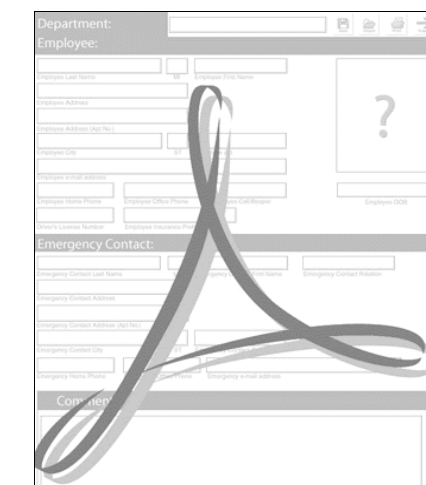
XForms

- XForms is a W3C standard designed to represent complex forms in XML
- It is a very rich standard which difficults the development of visual XForms editors and plug-ins
 - Web-services validated fields
 - Digital signatures on form segments
- Currently all browsers requiere some plug-in to display XForms documents
 - Firefox will probably become the first browser to natively support XForms
 - There are still no free XForms plug-ins for all browsers and all platforms



PDF Forms

- PDF files are a de-facto standard for electronic documents which can be used on any platform
- The main advantage of PDF documents is that they can be viewed in any browser exactly as they will appear as printed
- It is natural that Adobe has extended the format to support forms since most of these documents will eventually be printed and need to look exactly the same as forms filled manually
- Adobe offers a solution that automatically maps the XML data of PDF forms to relational databases fields with no programming





AJAX

- Using HTML, JavaScript and CSS it is possible to create good looking web clients of enterprise applications built in other languages (such as J2EE)
- The main advantage of AJAX is that it is based on open standards which work on most modern browsers without requiring additional plug-ins
- However, AJAX has some drawbacks
 - AJAX applications can only connect to the back-end using HTTP
 - HTML pages built with AJAX can become very complex and there still are no good development tools for this technology

Flash (Flex)

- Adobe Flex 2 is a web application development tool which can be viewed using the popular Flash plug-in
 - Good development tools
 - Flex based applications interact natively with Web Services, XML and (some) databases
- It offers dome interesting features
 - Very visually attractive application user interface, without requiring large amounts of bandwidth
 - Since the development tools are based on Eclipse, they can coexist peacefully with other languages
 - Can be used in JSPs and portlets



Flash (Flex)

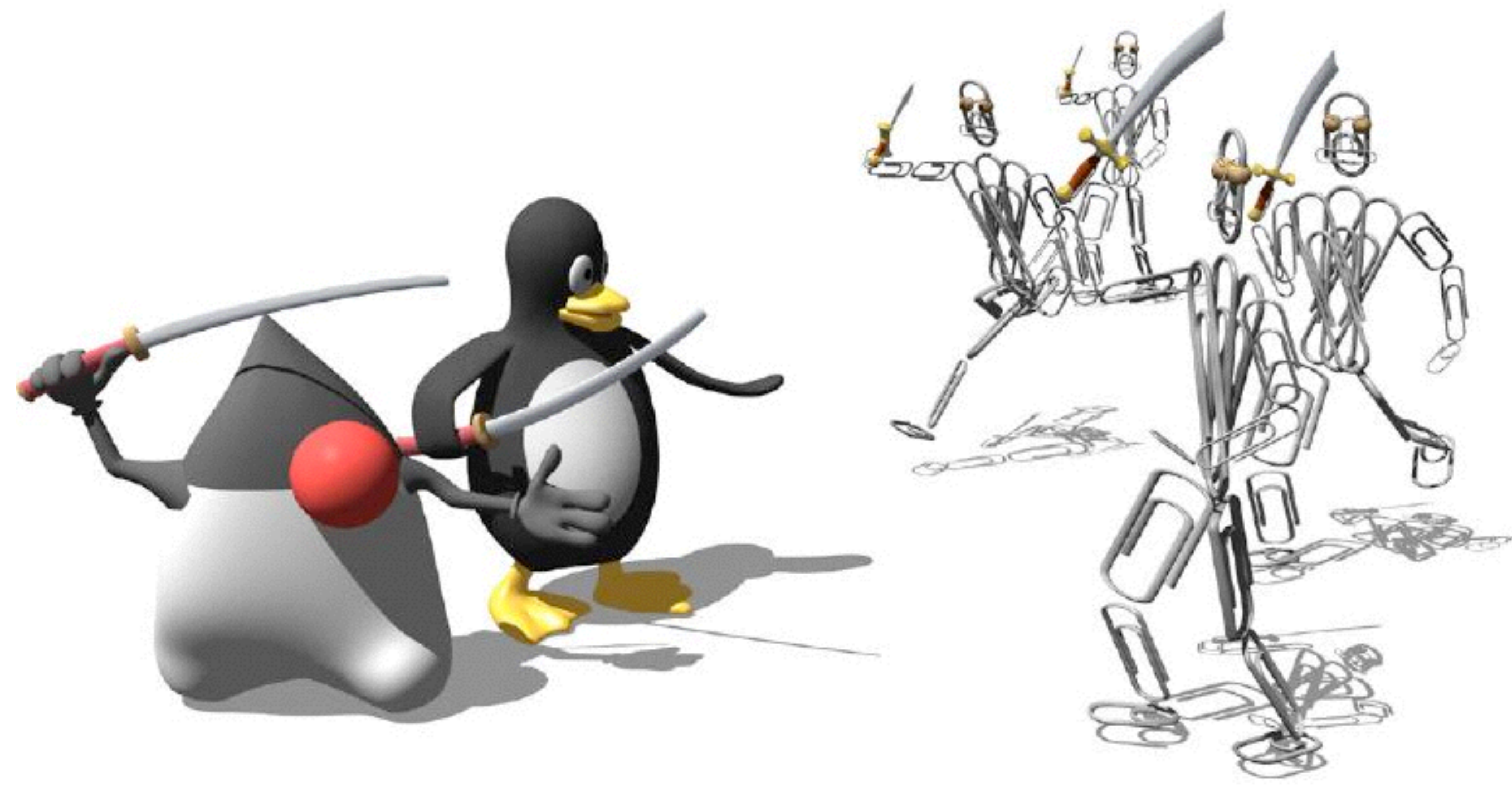
- However, Flex is neither a complete or perfect solution
 - It is not easy to integrate Flex applications with message queues, legacy databases as well as many legacy applications
 - It does not include a broad and robust security model
 - Requires learning new technologies
 - MXML
 - ActionScript
 - Proprietary technology
 - Requires a plug-in (even if very popular)
 - This can be problematic for mobile devices





Conclusions

- Besides all the options discussed in this presentation, there are many others that try to simplify the development of the presentation layer of an SOA based enterprise application
 - That is because none of the existing solutions completely solves the current needs of most developers
 - Some options try to solve the problem of designing complex forms
 - Other products focus on allowing the browser to perform more complex tasks than those currently supported by XHTML, JavaScript and CSS
 - Finally, other solutions try to achieve some of the same goals but maintaining an architectural coherence that is required in large and complex environments
- My personal recommendation is to look for standard based technologies and avoid proprietary products



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