

Aplicaciones de Internet Enriquecidas

Jornada divulgativa



Empresa
Digitala

Nombre Curso	Aplicaciones de Internet Enriquecidas
Fecha	29 de abril de 2009
Horario	de 9:30 a 13:00
Aula	Edificio Central Parque Tecnológico de Miramón
Ponente	<p>Dr. Jon Kepa Gerrikagoitia Informatika Saila www.eps.mondragon.edu ikgerrikagoitia@eps.mondragon.edu Tlf +34 943 739636 (Ext.297)</p> <p>Aitor Ariño Comercio Electrónico B2B 2000 www.b2b2000.com a.arino@ceb2b2000.com</p>

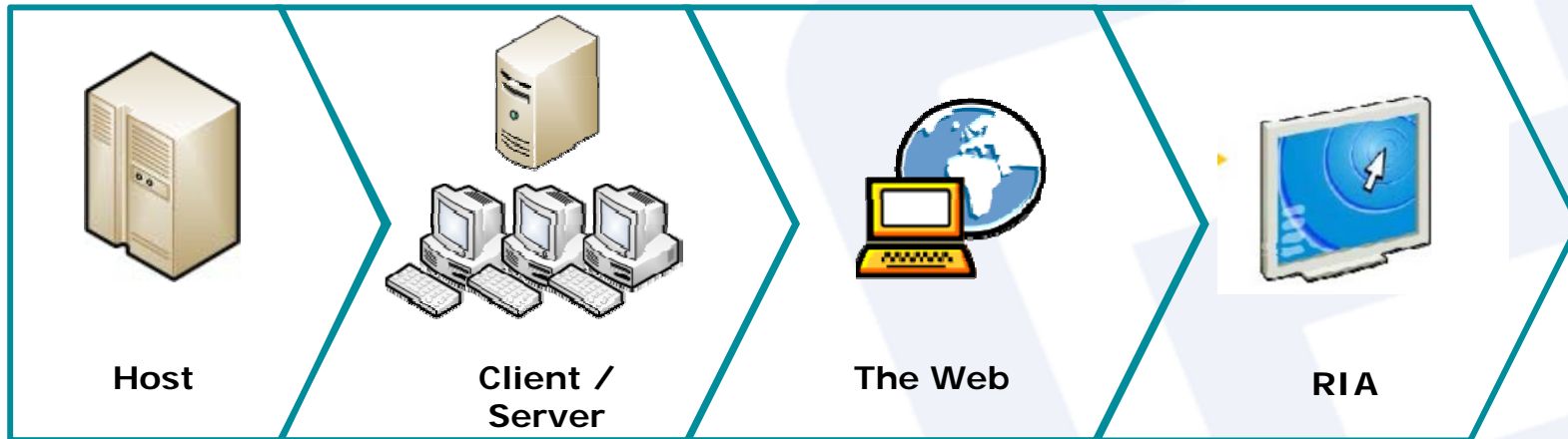
- 09:30 - 10:15 Introducción a las aplicaciones RIA
- 10:15 - 11:00 Tecnologías RIA
- 11:00 - 11:15 Descanso – Café
- 11:15 - 12:00 ¿Por qué Adobe Flex?
- 12:00 - 13:00 Casos prácticos:
 - Caso práctico 1:
Aplicación Cuadro de mando Vaillant Group
 - Caso práctico 2:
Casos de éxito RIA en la web

Introducción a RIA

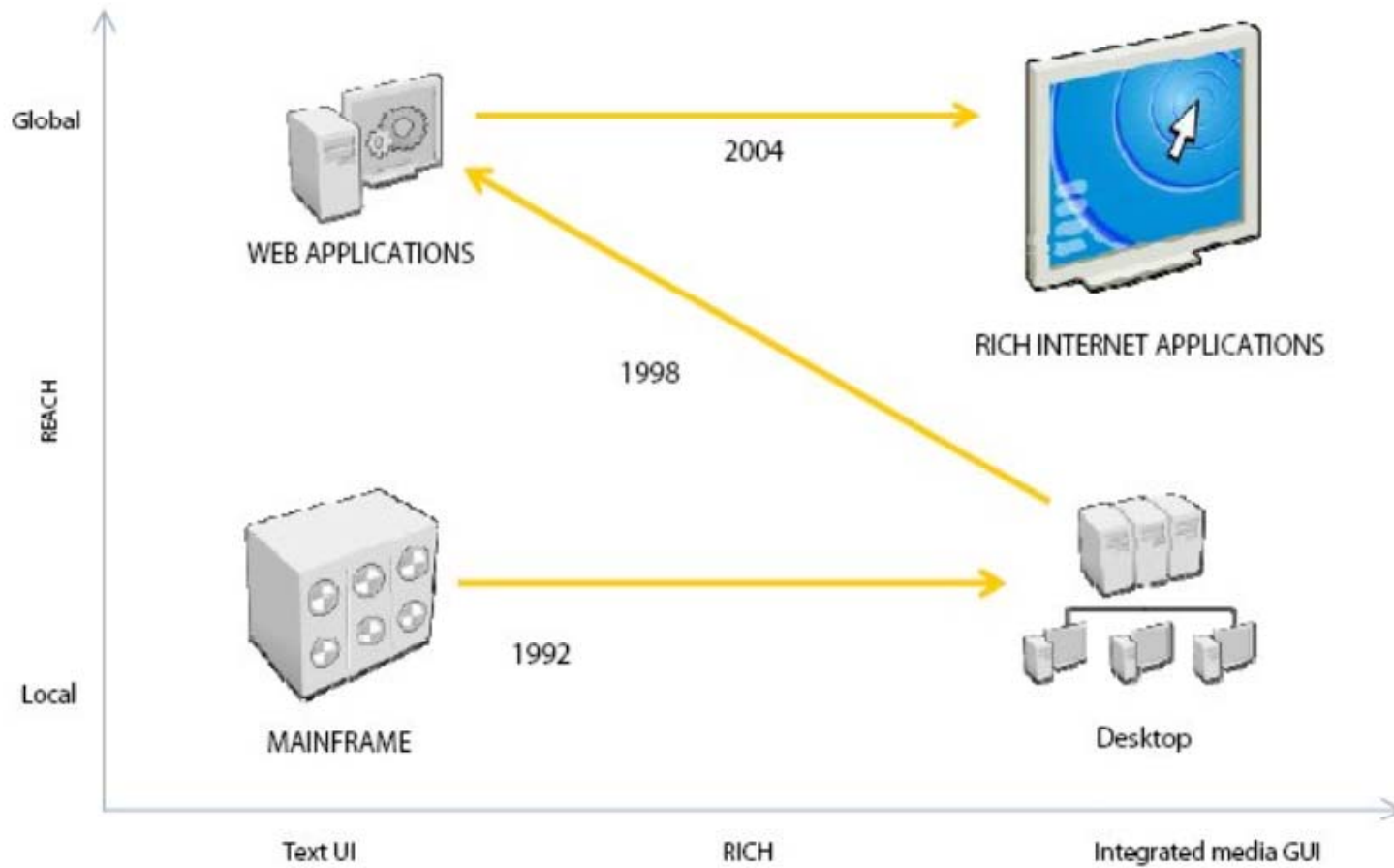


Enpresa
Digitala

Evolución de las aplicaciones



Evolución de las aplicaciones



Marco Casario - CTO Comtaste - www.comtaste.com

¿Por qué existe RIA?

- ¿Cuántas veces nos hemos quejado de lo **poco interactivas** que son las aplicaciones web que utilizamos habitualmente? Necesitan **recargar la página** cuando queremos realizar tareas que son simples o que requieren un cambio mínimo. Esta lentitud suele atormentarnos y hace que echemos de menos la agilidad que nos ofrecen otro tipo de aplicaciones que ya conocemos, como las instaladas en el propio PC o las aplicaciones cliente-servidor.
- A diferencia de las aplicaciones web, las RIA (Rich Internet Applications) **enriquecen la experiencia del usuario** a través de interfaces propias de aplicaciones de escritorio, que suelen ser más interactivas y con mayores capacidades gráficas y multimedia. Éste es el caso de **Gmail**, que tiene un interfaz web que permite al usuario efectuar acciones sobre su correo igual que si estuviera utilizando un programa cliente instalado en su propio equipo.

¿Por qué existe RIA?

- En los **años 90** y con el desarrollo de Internet, las **aplicaciones web** fueron tomando el espacio que antes ocupaban los mainframe y las aplicaciones cliente-servidor. La razón de este cambio estuvo en la **facilidad** que ofrecían estas nuevas aplicaciones para su **distribución y mantenimiento**, que conseguían llegar a más público utilizando un único cliente (navegador web) y en que hacían uso de los protocolos de comunicación de Internet. Pero como contraprestación, la **experiencia de uso de estas aplicaciones no era tan satisfactoria**. El navegador web obligaba a las aplicaciones a tener un interfaz estático tipo request-response, lo que redundaba en una recarga de página para obtener datos del servidor.
- **RIA** introduce un **nuevo modelo de programación** de aplicaciones que combina las ventajas de los dos modelos predominantes hasta el momento: el de las aplicaciones cliente-servidor y el del modelo multi-capa utilizado por las aplicaciones web, con un claro objetivo: **mejorar la experiencia del usuario**.

¿Qué se espera de RIA?



- Pero también existen ciertos retos con los que las tecnologías RIA deberán lidiar en el futuro:
 - Las RIA introducen cambios en los **hábitos de navegación** y en el uso de las aplicaciones web, y el usuario tardará un tiempo en digerirlos. Además, se dan ciertas complicaciones para el cumplimiento de los niveles de **accesibilidad**.
 - Algunas de las tecnologías RIA que hacen uso del navegador web deberán **superar algunos aspectos** no resueltos aún, como la posibilidad de introducir 'Favoritos' o la de utilizar el botón 'Atrás' del navegador web.
 - Las RIA deberán considerar la **optimización de los motores de búsqueda** o la capacidad de los sistemas de análisis para monitorizar sitios web contruidos con esta tecnología.
 - Aspectos relacionados con la **seguridad**.

¿Qué se espera de RIA?

- Con las RIA, los usuarios reciben **respuestas instantáneas** sin esperar a las conexiones de ida y vuelta contra el servidor que requerían las aplicaciones web tradicionales. Pero además, en muchos de los casos, las RIA pueden funcionar en cualquiera de los sistemas operativos que tenga instalado el usuario en su equipo (son multiplataforma) y utilizan el protocolo de comunicación de Internet, TCP/IP.
- Se **espera un gran desarrollo de este tipo de aplicaciones** en un futuro próximo de cara al gran público y al ámbito interno de las organizaciones. Las principales ventajas que introducen estas aplicaciones son las siguientes:
 - Agilidad en la respuesta.
 - Cálculos rápidos, controles prediseñados y funciones gráficas, interactivas y multimedia avanzadas.
 - En muchos casos no requieren de instalación en el equipo del usuario (es suficiente con disponer de un navegador web)
 - Uso desde cualquier ordenador con acceso a Internet.

- La adopción de nuevos enfoques relacionados con **SOA** (Arquitectura Orientada a Servicios) por parte de las organizaciones y por las suites de productos de software más extendidas en el mercado de las TI, constituye un caldo de cultivo interesante para la introducción de las tecnologías RIA.
- SOA ofrece una capa de abstracción que facilita la interrelación entre los servicios ofrecidos por las diferentes aplicaciones de una organización. Pero las ventajas de este nuevo enfoque habitualmente se encuentran ocultas para el usuario de negocio. Una manera de aflorar el valor que SOA reportará al usuario final será mostrando la facilidad de las aplicaciones RIA para **integrar los mash-ups y web-services** desarrollados siguiendo este enfoque.

- “‘rich client’ environment for Internet content and applications that will radically improve the quality of end-user applications, making the Internet more relevant and useful to businesses and consumers”. 2002. (flash player)
- “Rich Internet applications **combina** la capacidad de respuesta y las interfaces de usuario enriquecidas de las aplicaciones de escritorio con el amplio alcance de aplicaciones Web para entregar una más efectiva experiencia al usuario”. 2005.(flex)

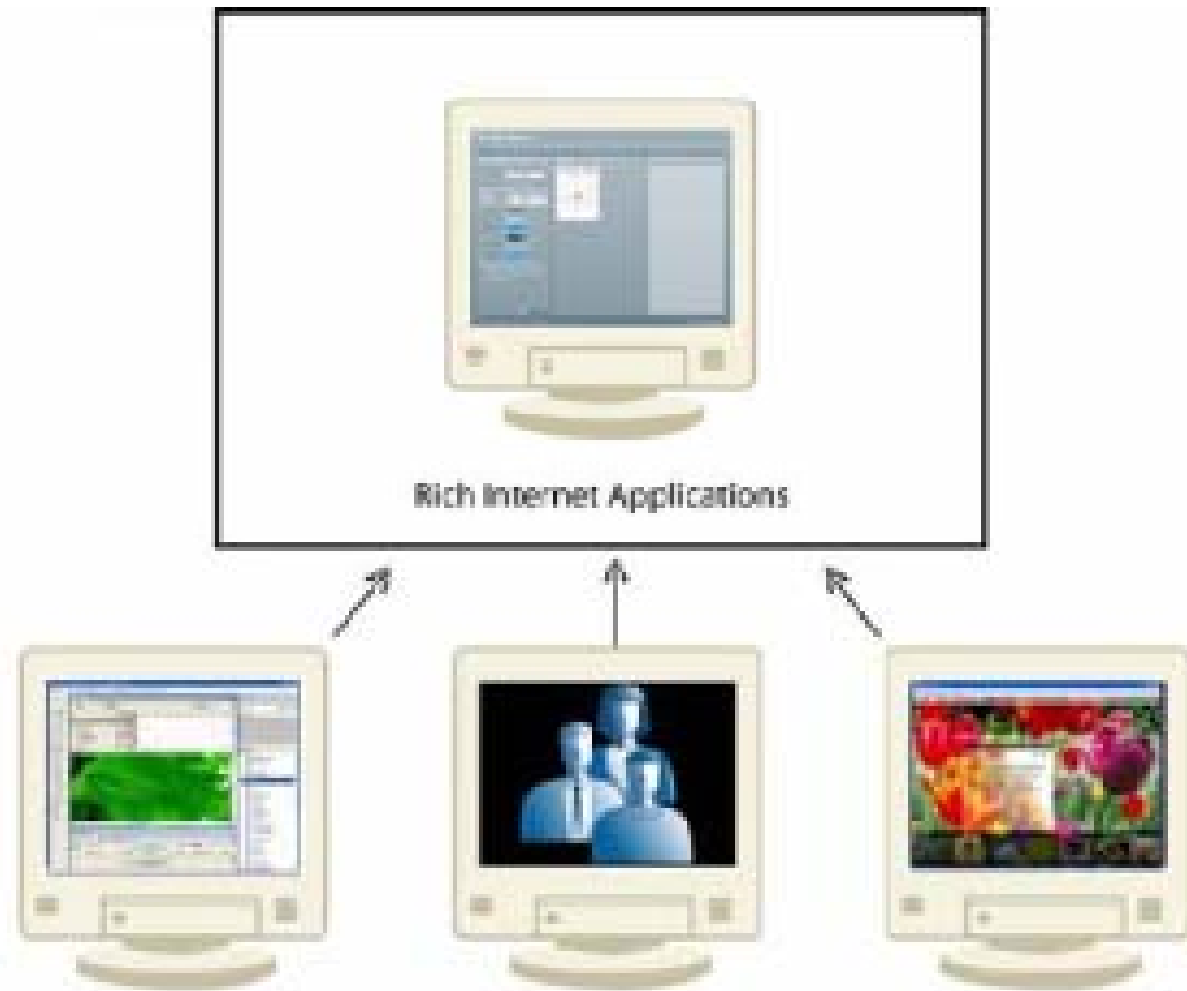
Rich Internet Applications (RIA): es una aplicación que se ejecuta en el navegador, pero utiliza una capa intermedia que puede pasar por alto el tradicional refresco de la página, que ha sido una norma en la mayoría de las actuales aplicaciones Web.

Rich Internet Application, es un término acuñado por la desaparecida (Absorbida por Adobe), pero siempre recordada Macromedia. Ahora cobra más vigencia que nunca su visión, ahora la de todos los competidores, de crear aplicación cada vez más enriquecidas “poderosas, robustas” en el lado del cliente. El panorama es cada vez más interesante, aquí las apuestas de los grandes jugadores en esta arena.



- La tecnología RIA es una aplicación Web distribuida que rompe con el **modelo de petición/respuesta HTTP**
- Las aplicaciones RIA manejan contenido con interacción visual para proveer **datos dinámicos** al cliente más **rápido** a la vez que proveen una **experiencia de usuario más rica**.

Definiciones

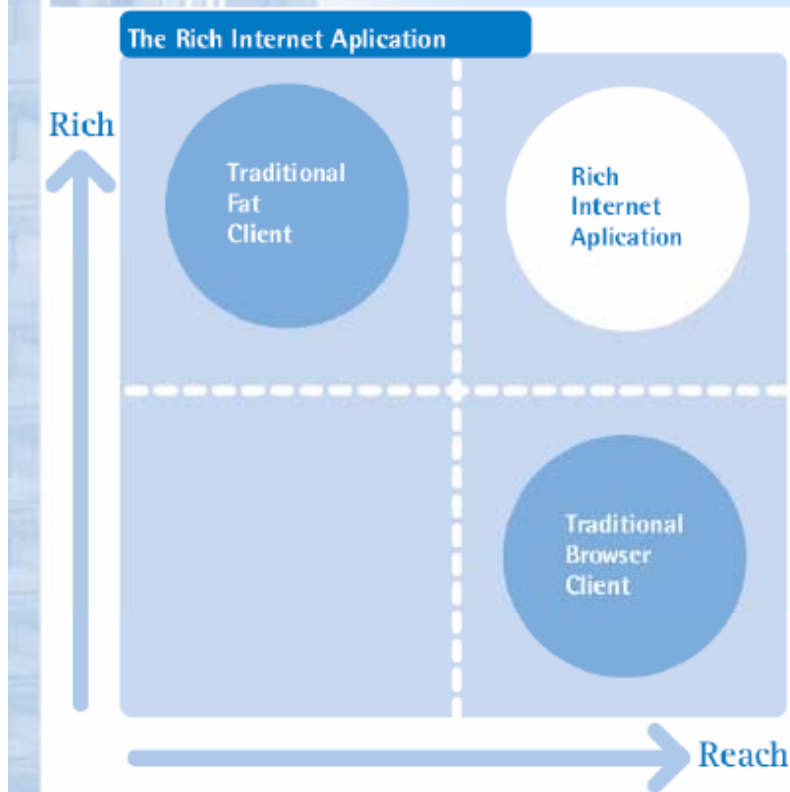


Lo mejor del **software**
de escritorio

Lo mejor de las
comunicaciones

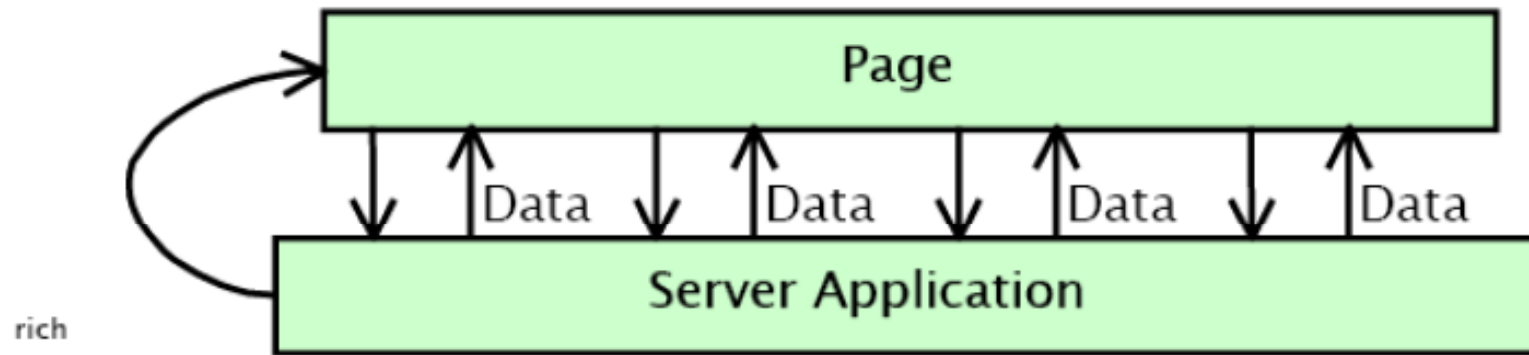
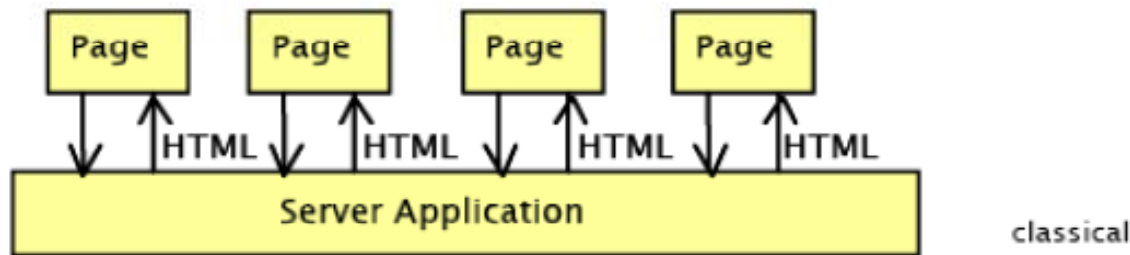
Lo mejor de la **web**

Rich Internet Application RIA (I)



- Interfaces de usuario enriquecidas (Rich)
- Habilidad para alcanzar la interfaz de usuario (Reach)

Rich Internet Application RIA (II)



¿Qué tecnología elegir?

Existen diferentes tecnologías de programación de aplicaciones RIA. Se consideran principalmente dos categorías de aplicaciones:

Desktop

- Aplicaciones que no utilizan navegador web y que se instalan en cada equipo personal. Suelen ser adecuadas para aplicaciones que requerirán un gran nivel de interacción con el usuario, intensivas en el uso de gráficos y con necesidad de utilizar funciones de otras aplicaciones.

RWA (Rich Web Applications)

- Aplicaciones que se ejecutan utilizando el navegador web. Adecuadas cuando se requiere una interacción con otras aplicaciones o servicios web y cuando el procesamiento se realiza en el servidor. Ejemplo: Google Docs and Spreadsheets.
- Pero existen más variables: Uso de **estándares abiertos** o no, así como las que resultan ser **multiplataforma** o solamente funcionan en unos determinados entornos.

- El objetivo de estas tecnologías es superar las **limitaciones de HTTP** y construir interfaces “fluidas” y más usables que imiten la inmediatez del escritorio.
- HTTP, a diferencia del escritorio, **no mantiene estado**. Esto quiere decir que, entre una página y otra, no hay “memoria” de las acciones efectuadas anteriormente por el usuario.
- Cada vez que presionamos un link, nuestro navegador envía una petición (“request”) al servidor, quien a su vez procesa los datos enviados y responde adecuadamente. El navegador recibe y despliega esta respuesta y la comunicación termina hasta la siguiente interacción del usuario. Esta es la arquitectura que nos ha acostumbrado a entender la internet como una serie de “páginas” o documentos individuales, relacionados entre si por hipervínculos.

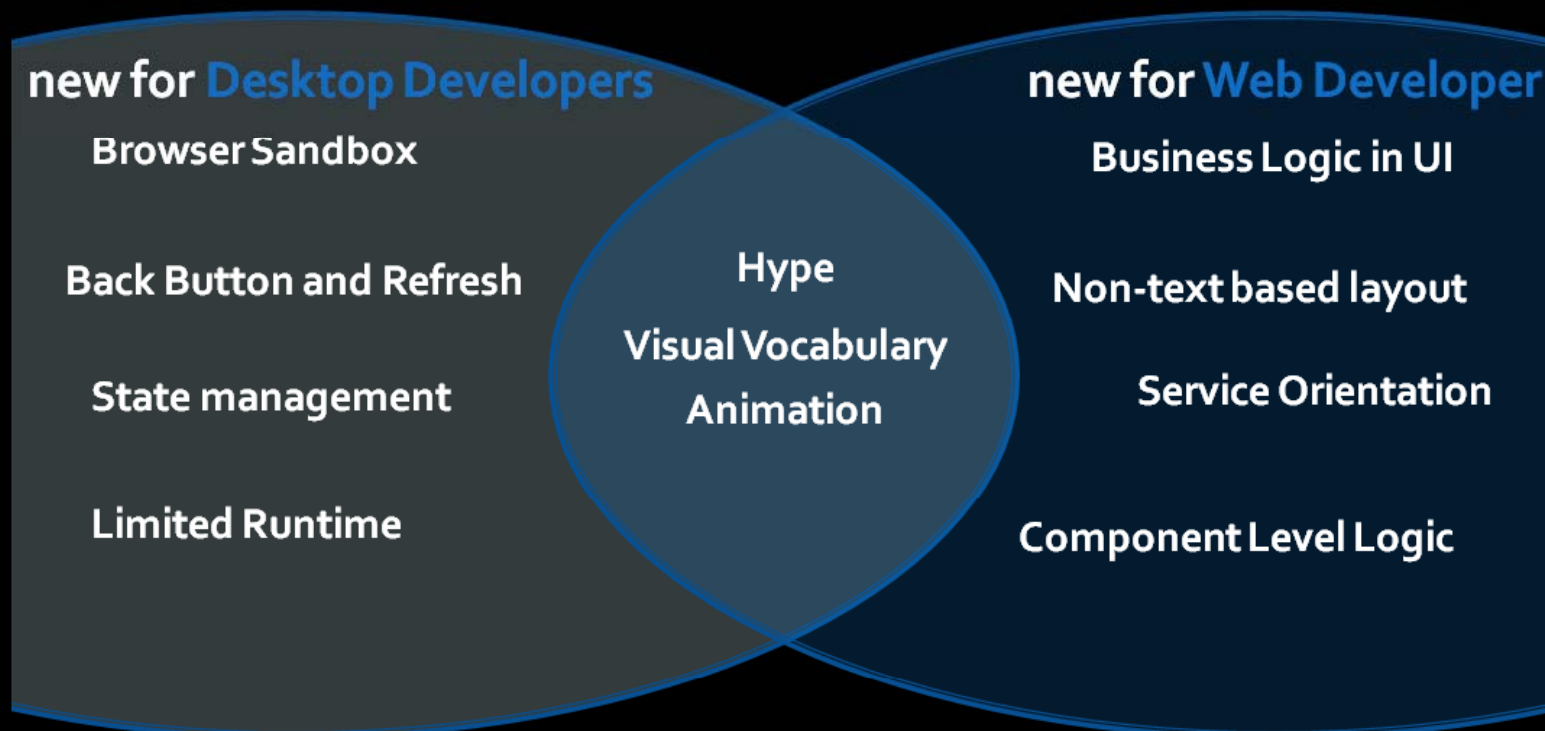
- Las aplicaciones de escritorio, por el contrario, mantienen un contacto permanente entre los procesos internos del programa y lo que sucede en la interfaz de usuario. Ofrecen una experiencia de usuario más fluida entre una acción y otra.
- Flash, JavaScript y las tecnologías citadas al comienzo tienen la virtud de mantener estado, al menos en la interfaz. Si bien la comunicación con el servidor se sigue haciendo a intervalos discretos mediante HTTP, estas aplicaciones evitan la parcelación de la interacción en “páginas” y logran una experiencia mucho más cercana al escritorio.
- Pero hay ocasiones donde las virtudes de HTTP y HTML, el modelo tradicional, son esenciales. En el modelo de petición y respuesta de HTTP, cada documento HTML - de hecho cada imagen y cada elemento que incluyamos en la página - es un recurso. Cada recurso tiene su propia URI en la Red y en virtud de eso puede ser visitado individualmente y utilizado en múltiples contextos.

- Los recursos son reutilizables ,y se pueden agregar a los favoritos del navegador. Pero lo más importante es que los recursos son indexables (los motores de búsqueda pueden llegar a ellos, analizarlos y categorizarlos, facilitando la búsqueda para nosotros los usuarios)
- Incluso las RIA's que respetan los estándares tecnológicos y aquellas accesibles suelen generar el contenido dinámicamente. En las aplicaciones Flash los datos son **totalmente invisibles para los motores de búsqueda**, que sólo entienden información estructurada en formatos conocidos.
- En muchos casos las RIA's se han hecho imprescindibles. Aplicaciones innovadoras como **gMail** facilitan el manejo de grandes volúmenes de información junto con la ubicuidad de la Web. En otros, los comportamientos “de escritorio” en la web **perjudican la usabilidad**.

- No es necesario un proceso de instalación para acceder a aplicaciones o sitios web
- Ofrece un interface de usuario rico que se asemeja al “look and feel” de una interface gráfica de usuario nativa (GUI) sobre un navegador
- Las aplicaciones en sí son pequeñas y proveen un rendimiento razonable tanto a usuarios con conectividad limitada como con ancho de banda importante.

- Aprovecha la CPU del cliente
- Interfaz de Usuario en tiempo real
- Componentes para aplicaciones Web basadas en browsers
- Balance en las cargas del cliente y el servidor
- Reducción del tráfico de red
- No requiere instalación
- Aumenta la productividad del personal de la organización
- Aumenta la capacidad de respuesta del sitio web.

What's different with RIA?





Maximizar la experiencia de usuario

La **experiencia de usuario** es el conjunto de factores relativos a la interacción del usuario, con un entorno o dispositivo concretos, cuyo resultado es la generación de una percepción positiva de dicho servicio, producto o dispositivo.

La **experiencia de usuario** depende no sólo de los factores relativos al diseño (usabilidad, diseño de interacción, accesibilidad, diseño gráfico y visual, calidad de los contenidos, etc) sino además de aspectos relativos a las emociones, sentimientos, transmisión de la marca, confiabilidad del producto, etc.



Experience Engineering

- User Needs +
- Interaction Design +
- Information Design +
- Visual Design

Experiences that create value!

Proporcionar un **aspecto atractivo** a las aplicaciones empresariales mediante la utilización inteligente de aplicaciones enriquecidas de internet y contenidos multimedia.

Porque si se mejora el acceso a la información:

- Los clientes son más felices
- Se mejora la **calidad del trabajo** de los empleados

El equipo



J2EE Developers



Flex/Flash Developers



Creative UI Designers

- Drag & Drop
- Menu & Tool bars
- Windows & Wizards
- Panels
- Trees
- Form validation
- Non-HTML controls
 - Accordion
 - Combobox
 - Spinner box
 - Sliders
- Keyboard Actions
- Context Menus

■ Enlaces:

<http://www.adobe.com/es/products/flex>

<http://www.openlaszlo.org> (Laszlo es otra implementación Open de Flex)

<http://www.microsoft.com/silverlight>

<http://java.sun.com/javafx>

<https://openjfx.dev.java.net>

■ Ejemplos

<http://www.adobe.com/devnet/flex/?tab:samples=1>

<http://www.openlaszlo.org/demos>

<https://openjfx.dev.java.net>



GOI ESKOLA
POLITEKNIKOA
ESCUELA
POLITÉCNICA
SUPERIOR

Tecnologías RIA



Enpresa
Digitala

- Es un **hecho indiscutible** que las **aplicaciones de internet enriquecidas** ofrecen a los desarrolladores y usuario web importantes beneficios respecto a las aplicaciones HTML
- El problema es **decidir** que solución elegir

- Comenzamos echando un vistazo a las **características** y **beneficios** de las tecnologías RIA con los **pros** y **cons** de cada una de ellas.

- Vamos a examinar las siguientes tecnologías:
 - Ajax
 - Adobe Flex
 - Laszlo
 - Microsoft Silverlight
 - JavaFX
 - XUL

A la hora de elegir una tecnología RIA habría que considerar los siguiente **factores**:

1. El alcance (browsers, plugins,...)
2. Productos open source o productos comerciales
3. Conocimiento interno
4. Tiempo de desarrollo (time to market)
5. Comunidad e infraestructura de soporte
6. Gestión de:
 - » Contenidos ricos
 - » Interacción de usuario

- AJAX (Asynchronous JavaScript and XML) es una técnica de desarrollo web utilizada para crear aplicaciones web interactivas
- El objetivo es hacer que las páginas web intercambien pequeñas cantidades de datos con el servidor detrás de la escena para que la página web al completo no tenga que recargarse cada vez que el usuario requiera una modificación en la misma

AJAX utiliza una combinación de:

- XHTML y CSS para los estilos.
- Un lenguaje de script del lado del cliente(normalmente JavaScript) para acceder a objetos DOM
- El objeto XMLHttpRequest para intercambio de datos asíncronos con el servidor
- XML se utiliza como formato para transferir datos entre el servidor y el cliente (cualquier formato funcionaría, incluyendo HTML preformateado, texto plano o JSON)

- No requiere una gran curva de aprendizaje para profesionales experimentados en JavaScript + HTML + CSS
- Comunidad numerosa de desarrolladores
- Gran cantidad de frameworks libres que aumentan la productividad del desarrollo
- Muchos recursos en la web
- La adopción de AJAX para empezar es rápida

- AJAX requiere esfuerzos en pruebas y depuración
- Elegir una solución comercial AJAX requiere adoptar un framework y herramienta propietario
- El mercado AJAX está fragmentado
- La comunidad es grande pero existe bastante confusión porque no hay ninguna voz oficial que seguir

- Adobe Flex es una solución de desarrollo de aplicaciones para crear y proporcionar aplicaciones de internet enriquecidas a través de la web utilizando **Flash Player**



Los siguientes componentes son parte de la línea de productos Flex:

- **Flash Player 9:** El último runtime de alto rendimiento de la parte cliente para potenciar experiencias web
- **Flex SDK:** El núcleo del modelo de programación y la librería de componentes para Flex
- **Flex Builder:** Un IDE basado en eclipse para desarrollar aplicaciones de internet enriquecidas con el Flex SDK
- **LiveCycle Data Services:** Servicios de acceso a datos y arquitectura abierta que provee aplicaciones de internet enriquecidas con acceso a datos intensivo que se encuentra muy integrada con las infraestructuras empresariales orientadas a servicios
- **Flex Charting componentes:** Componentes extensibles para visualización de datos avanzadas

- **MXML** significa **M**acromedia **eX**tensible **M**arkup **L**anguage
- Es un lenguaje de programación declarativo basado en XML

```
Flex Start Page  LangileenGestioa.mxml X
Source Design
1 <?xml version="1.0" encoding="utf-8"?>
2 <mx:Application xmlns:mx="http://www.adobe.com/2006/mxml" layout="absolute" backgroundGradientAlpha:
3   <mx:Label x="194" y="77" text="LANGILEEN" fontFamily="Verdana" fontSize="20" color="#74121B" fo
4   <mx:Label x="338" y="77" text="GESTIOA" fontFamily="Verdana" fontSize="20" color="#74121B" fontF
5   <mx:TextInput x="276" y="135"/>
6   <mx:Label x="194" y="139" text="Izena" color="#DCF5FB" fontFamily="Verdana" fontSize="12" fontW
7   <mx:TextInput x="276" y="165"/>
8   <mx:TextInput x="276" y="195"/>
9   <mx:Label x="196" y="169" text="Abizena" fontWeight="bold" fontSize="12" color="#DCF5FB"/>
10  <mx:Label x="196" y="199" text="Herria" color="#DCF5FB" fontSize="12" fontWeight="bold"/>
11  <mx:ComboBox x="276" y="225">
12    <mx:String>Bizkaia</mx:String>
13    <mx:String>Gipuzkoa</mx:String>
14    <mx:String>Araba</mx:String>
15  </mx:ComboBox>
16  <mx:Label x="196" y="227" text="Probintzia" color="#DCF5FB" fontSize="12" fontWeight="bold"/>
17  <mx:Button x="276" y="264" label="Bidali" fontSize="14"/>
18 </mx:Application>
19
```

- Cumple al 100% la especificación ECMA-262
- Fácil de aprender (parecido a Java), muy potente y orientado a objetos

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml" layout="absolute" creationComplete="carga();">

  <mx:Script>

    <![CDATA[

      import mx.rpc.events.FaultEvent;
      import mx.rpc.events.ResultEvent;
      import mx.rpc.*;
      import mx.collections.ArrayCollection;
      import mx.controls.Alert;
      public var cadena:Array;

      public function carga():void{
        remoteObject.bezeroak();
      }
      public function bezeroak_Result(event:ResultEvent):void{
        cadena= event.result as Array;
        mydg.dataProvider=cadena;
      }
      public function onFault(event:FaultEvent):void{
        Alert.show(event.fault.faultString,"error");
      }

    ]]>
  </mx:Script>
```



Ventajas de Adobe Flex

- Flash Player se encuentra implantado en más del 98% de los navegadores
- Flex Builder es un entorno de desarrollo de alta productividad
- Soporta ActionScript 3.0, XML y CSS
- La comunidad Flex es muy grande y existen gran cantidad de recursos
- Flex 3 SDK va a ser Open Source
- Soporta los protocolos SOAP/AMF/HTTP facilitando las entradas y salidas de datos
- Integración con aplicaciones JavaScript y AJAX
- Integración con el mundo empresarial (LiveCycle DS)

- La curva de aprendizaje de Flex es mayor que HTML y JavaScript
- Flex está atado a Flash Player (no es DHTML)
- Flash Player no es abierto sino propietario de Adobe

- **Silverlight** es un plug-in de .NET que permite a los desarrolladores construir experiencias multimedia ricas y RIAs para navegadores. La versión actual esta soportada para Firefox, Safari e Internet Explorer tanto para Windows como Mac.



Microsoft®
Silverlight™



Cómo funciona Silverlight

- El framework Silverlight es un subconjunto del framework completo .NET (utilizado para WPF) y posibilita a los desarrolladores .NET reutilizar de forma sencilla las habilidades y conocimientos adquiridos
- Codec built-in para reproducir video VC-1 y WMV y audio MP3 y WMA en un navegador
- Soporta la reproducción de video de alta definición (720p)
- Soporta la descarga progresiva y reproducción de contenidos multimedia desde cualquier servidor web sin requerir ningún software especial

- Utiliza el lenguaje XAML (*eXtensible Application Markup Language*) para definir interfaces de usuario y JavaScript para programar

Este ejemplo en XAML muestra un texto "Hola Mundo!" dentro de un contenedor del tipo Canvas.

```
<Canvas xmlns="http://schemas.microsoft.com/client/2007"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml" >
```

```
<TextBlock>Hola Mundo! </TextBlock>
```

```
</Canvas>
```



Ventajas de Silverlight

- Microsoft proporciona dos herramientas diferentes: **Expression Studio** para diseñadores y **Visual Studio** para desarrolladores
- Gran comunidad de desarrolladores .NET
- Soporta el formato de video VC-1 que proporciona una robusta protección de contenidos a través de una solución nativa DRM

Inconvenientes de Silverlight

- Las características de Silverlight son muy pobres por el momento. Básicamente se orientan a la reproducción de video
- El plugin no está muy extendido
- Muy lejos de lo que se puede desarrollar con Flash/Flex u otras tecnologías basadas en Flash
- No soporta Linux por el momento (hay algunos resultados positivos a través del proyecto Mono)

- Laszlo es un proyecto open source de una plataforma (compilador, herramientas y librerías) para desarrollar aplicaciones de internet enriquecidas desarrollada por Laszlo Systems.
- Laszlo es una tecnología cliente/servidor. Al igual que Flex puede ser utilizado por cualquier lenguaje del lado de servidor.



Cómo funciona Laszlo

- Laszlo utiliza un lenguaje XML declarativo para definir interfaces de usuario (ficheros con extensión .lzx)
- Las funcionalidades avanzadas como el databinding etc se implementan utilizando JavaScript/ActionScript
- Los ficheros LZX son compilados por Laszlo en ficheros SWF que se ejecutan por Flash Player
- Laszlo es capaz de exportar a AJAX (utilizando el framework DOJO)

Cómo funciona Laszlo

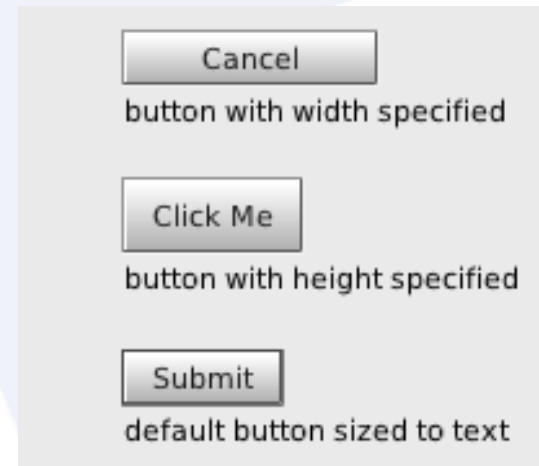


- El lenguaje de script es JavaScript
- Fácil de aprender y muy potente

```
<canvas height="550" bgcolor="0xaeaea" >  
<font name="smallfont" src="verity/verity9.ttf"/>  
<class name="box" extends="basecomponent"  
focusable="false" height="10" width="10"/>  
<view x="40" y="40">
```

```
<view >  
<simplelayout axis="y" spacing="2"/>  
<button width="100">Cancel</button>  
<text>button with width specified</text>  
</view>
```

```
<view >  
<simplelayout axis="y" spacing="2"/>  
<button height="30">Click Me</button>  
<text y="10">button with height specified</text>  
</view>
```



Para ver más componentes:

<http://www.laszlo systems.com/lps/examples/components/>

Ventajas de Laszlo

- Alta utilización de Flash Player (más del 98%)
- Laszlo es open source
- No es necesario instalar nada
- Se utiliza JavaScript para programar componentes
- El IDE OpenLaszlo está basado en Eclipse
- Soporta J2ME para dispositivos móviles

Desventajas de Open Laszlo

- La curva de aprendizaje de Laszlo es mayor que la de HTML+JavaScript
- Laszlo no está basado en la última versión de Flash Player
- El rendimiento es inferior a las aplicaciones Flex
- La comunidad de desarrolladores no es tan grande por lo que no hay tantos recursos

- JavaFX en la nueva familia de productos Sun basada en tecnología Java con el objetivo de llegar al mercado de las aplicaciones enriquecidas para internet.
- Java FX está basado en la máquina virtual Java pudiendo ejecutarse desde el Java Runtime



- JavaFX Script es un lenguaje de script de alta productividad para desarrolladores de contenidos para crear contenidos ricos multimedia para desplegar en tecnología Java

```
package foobar.samples;

import java.util.Date;
import java.text.SimpleDateFormat;
import javafx.ui.*;

var formatter = new java.text.SimpleDateFormat("EEE MMM dd, hh:mm aa");
var now = new java.util.Date();
var timestamp = formatter.<<format>>(now);

Frame {
  visible: true
  title: "Text Clock"
  width: 200
  content: SimpleLabel { text: bind timestamp }
```

Ventajas de JavaFX



- Lenguaje potente y robusto
- Tipado estático: tiene las mismas características que Java en cuanto a estructuración de código, reutilización y encapsulación
- Soporte para móviles: JavaFX Mobile es un entorno operativo y de aplicación móvil construido sobre las tecnologías Java y Linux
- Amplia comunidad de desarrolladores Java que pueden comenzar a desarrollar aplicaciones JavaFX de forma rápida

Desventajas de JavaFX

- Ausencia de entornos RAD (Rapid Application Development) para desarrollar aplicaciones más rápido
- No es un proyecto maduro, se encuentra es sus etapas iniciales.
- Las aplicaciones son interpretadas (pronto habrá un compilador)
- La curva de aprendizaje es muy alta para desarrolladores no Java
- La productividad del desarrollo no es alta

- XUL se refiere al lenguaje XML para construir interfaces de usuario.
- XUL es una especificación de presentación para crear interfaces de usuario crossplataform, crossdevice ligeras



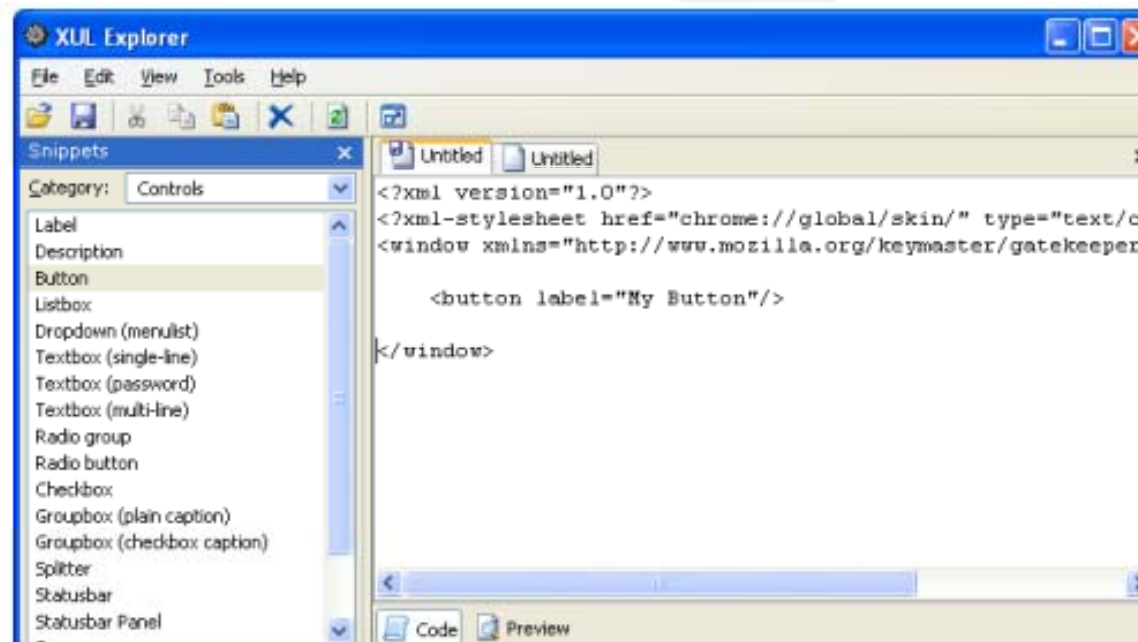
Cómo funciona XUL

- XUL está soportado y utilizado por Mozilla
- Por ejemplo la ventana de preferencias de Firefox está desarrollada utilizando XUL



Cómo funciona XUL

- Al igual que XHTML, XUL utiliza un lenguaje de marcado para definir interfaces de usuario
- El lenguaje de marcado proporciona acceso a numerosos componentes a la par de frameworks de programáticos



IDE XUL explorer

Ventajas de XUL

- Es una tecnología producida por el proyecto open source Mozilla y soportada por el navegador Mozilla
- Existe una variedad de herramientas open source para utilizar
- Esta integrado con el motor Gecko que permite la utilización de muchos estándares Web
- Amplia comunidad de desarrolladores

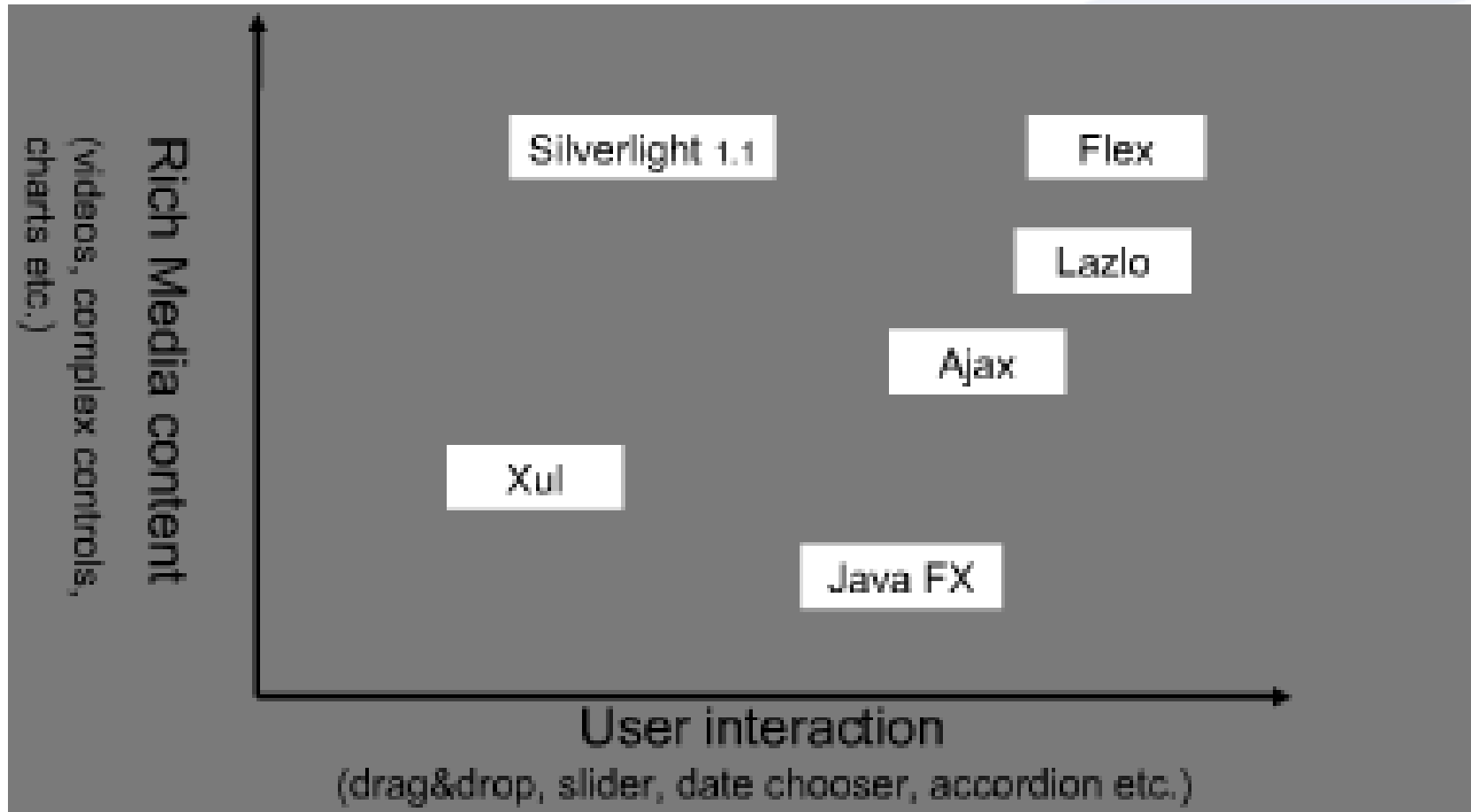
Desventajas de XUL

- No es un estándar oficial y las especificaciones no son completas. No tiene un gran soporte de Mozilla
- No está apoyado por un fabricante de referencia
- No existen demasiados recursos en la web
- La curva de aprendizaje es alta
- La productividad del desarrollo no es rápida

- **Adobe Air:** Es un sistema runtime que permite a los desarrolladores web contruir y desplegar en el escritorio RIAs basadas en web(Fash, Flex, HTML, Ajax). Integra SQLite.
- **Google Gears:** Google Gears es una extensión de navegador open source que proporciona funcionalidad offline utilizando APIs JavaScript
- **Prism:** Ha sido anunciado por Mozilla Labs como una iniciativa para crear aplicaciones de escritorio utilizando tecnologías web. Permite crear accesos directos desde el escritorio o menu de inicio a las aplicaciones web favoritas
- **Dojo offline:** Es un toolkit open source basado en Google Gears que permite crear una web offline. Extiende Google Gears con una funcionalidad importante y crea una API de mas alto nivel.
- **XULRunner:** runtime de aplicaciones de escritorio con un motor de navegador embebido que permite a los desarrolladores acceder al sistema de ficheros

El sexto factor

Dominio RIA



[Marco Casario at the Web 2.0 Expo of Berlin]

■ Each RIA technology is GOOD ... Not GOD!

- Desterrar la idea de que existe la tecnología perfecta que encaja con las necesidades de todos los proyectos
- Usabilidad: Debido a su naturaleza RIA puede derivar fácilmente en una mala usabilidad. A la hora de diseñar y desarrollar una RIA poner atención a aspectos de usabilidad (pistas visuales, iconos, gráficos, gestión de errores, mensajes de usuario...).
- Una RIA puede embeber muchas flujos de tareas en una única página. Aunque sea una ventaja puede acarrear problemas.

- Consideramos que **Adobe lleva la delantera** y esta aprovechando muy bien sus tecnologías consolidadas (las heredadas de Macromedia), pero el reto es complejo
- **Microsoft** aprovecha su **capacidad publicitaria** y el recorrido de los pioneros para atacar los puntos críticos.
- **Java** por su parte tiene un gran nicho capturado al cual le será más sencillo seguir en su línea de desarrollo y aprender las nuevas clases que se agregan con FX.
- **FireFox** por su parte sigue concentrado en lo suyo, fortalecer su navegador y depurar sus herramientas de desarrollo, este si que viene pisando duro, por lo que ya no es tan fácil saber si es una buena estrategia o no amarrar sus tecnologías al motor de Mozilla.

¿Por qué Adobe Flex?



Enpresa
Digitala



RIAs HOLD GREAT APPEAL FOR SITE OWNERS

The increasing popularity of Ajax and Flex applications is changing the way companies do business online. Why are RIAs so attractive to site owners?

- **People know them and like them.** Fifty-two percent of online consumers have used highly interactive applications like Google Maps and Zillow.com. More importantly, the overwhelming majority of those who tried RIAs say that rich applications greatly enhance their Web experience.¹
- **They enable tasks that HTML can't.** RIAs give companies a superior alternative to HTML for crafting customer interactions.² That's because RIAs provide a range of capabilities — like improved data visualization and streamlined processes — that go beyond the limitations imposed by HTML (see Figure 1).
- **They get results.** Firms that measure the business impact of their RIAs say that rich applications meet or exceed their goals (see Figure 2). As the RIA success stories of these early adopters become more widely known, site owners at mainstream companies also look to RIAs to boost online effectiveness.



Figure 1 Firms Use RIAs For A Wide Variety Of Purposes

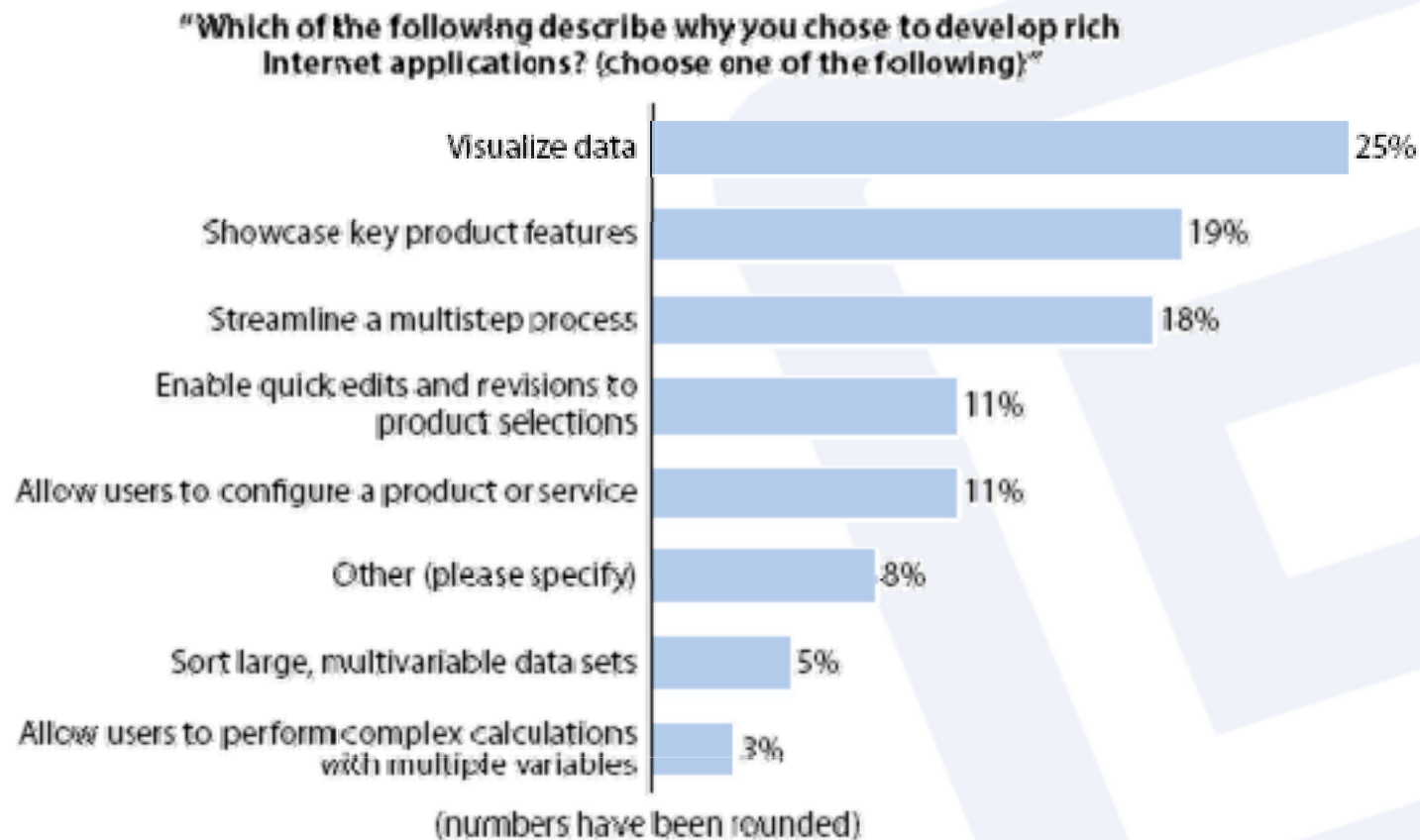


Figure 2 RIAs Usually Meet Or Exceed Business Expectations

"Which of the following statements best reflects the business impact of your largest RIA implementation? (choose one of the following)"

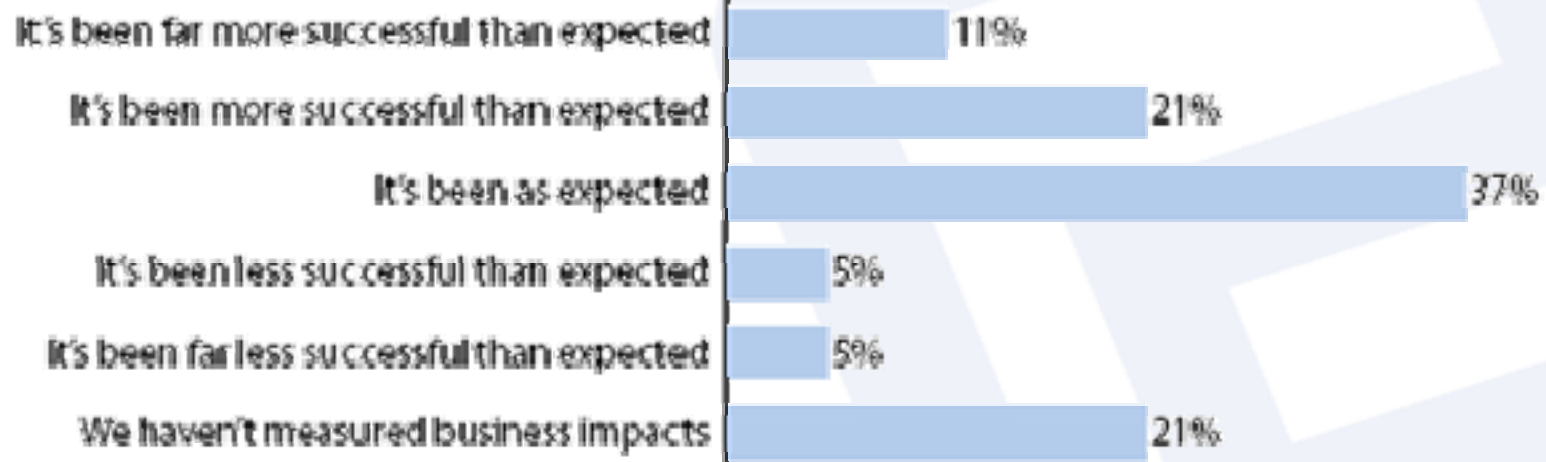


Figure 3 Many Firms Haven't Begun Developing RIAs

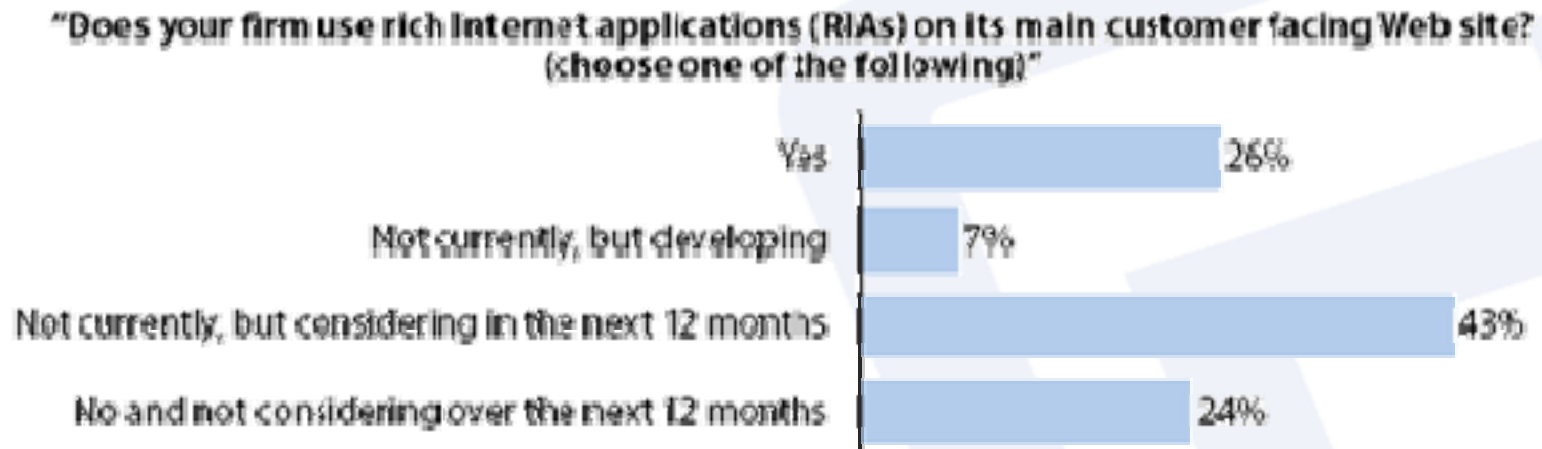




Figure 4 Sample RIA Cost Model

Tool costs	Flex builder		Number of developers		
	\$800	×	5	=	\$4,000
	Flex data services		Number of CPU		
	\$6,000	×	2	=	\$12,000
Total cost of tools					\$16,000
Ramp costs	Intall and setup		Number of days		
	\$1,000	×	2	=	\$2,000
	Training costs		Number of days		
	\$2,000	×	5	=	\$10,000
	Short-term productivity cost		Number of developer months		
	\$10,000	×	10	=	\$100,000
Total cost of ramp-up					\$112,000
Total cost of RIA					\$128,000

Manufacturers Will Boost Sales And Margins Of Configurable Products

Selling complex, customized products with a static, HTML-based configurator is difficult. Our model shows that streamlining the sales process by upgrading to a powerful, RIA-based configurator brings big economic benefits (see Figure 5). That's because:

- **Improved ease of use drives higher conversion rates and order size.** Our research found that more shoppers convert to buyers when they can easily trade off product options and costs in real time. Interestingly, we also found that average order size increases due to RIA configurators' superior ability to present options and add-ons. And because of increased ease of completing complex orders online, fewer customers give up and switch channels to higher-cost phone agents, resulting in significant savings.
- **Rich media helps boost margins.** RIAs not only enable better configurations, they also allow firms to embed video and other contextual help content into applications. The second part of

Hotel Sites Can Expect Increased Conversion Rates And Higher Reservation Values

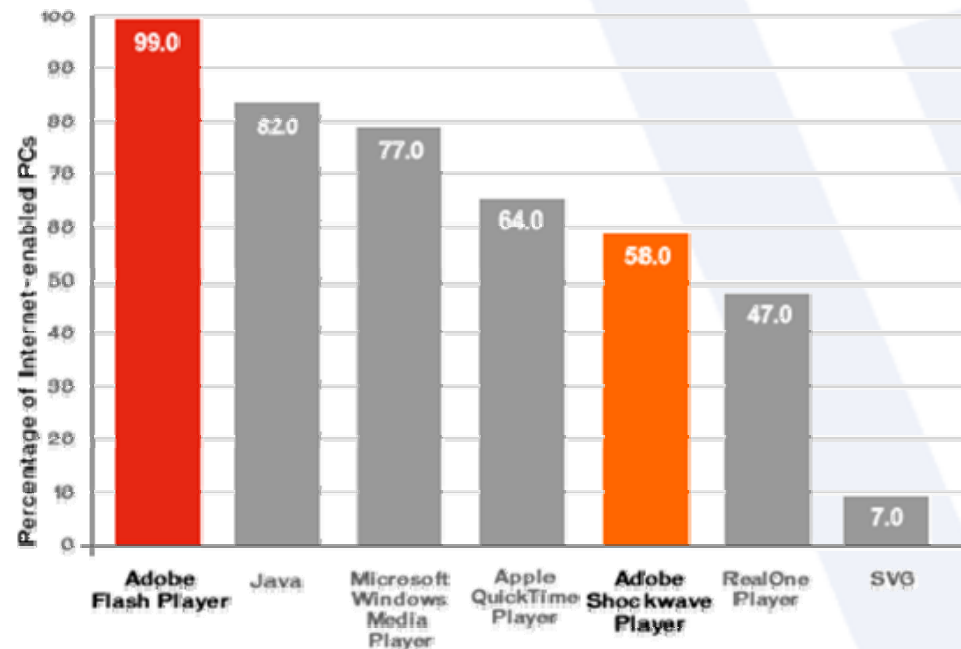
Similar to buying a complex manufactured product, making a travel reservation requires shoppers to choose between complex variables — many with critical interdependencies like available dates, class of service, and price. Our model shows that hotel sites benefit by adopting RIAs to facilitate these multifaceted decisions because (see Figure 6):

- **Single-screen checkout reduces drop-off rates.** Companies can make the reservation process quicker and less prone to error with RIAs that expose available options and validate fields in real time. The result: fewer abandoned orders and lower customer service costs. According to our model, enhancements like these that keep users engaged and reduce system hiccups will increase online conversion rates and take some burden off of phone reservation agents.
- **Improved merchandising increases reservation size.** Hotel sites struggle to sell upgrades and bundles because multistep reservation processes can give users tunnel vision that makes them unaware of these possibilities. But RIAs like TravelCLICK's iHotelier's iStay put upgrades and add-ons in front of bookers, where they can be easily added on the fly.⁴ As a result, our model shows that RIAs boost revenues by increasing average booking values — while decreasing phone reservation costs.⁵

Figure 6 ROI Model For Hotel Reservations Site

	Starting assumptions	% Improvement, low	Low benefit after 1 year	% Improvement, high	High benefit after 1 year
Increased sales					
Visitors to booking engine	1,500,000		1,500,000		1,500,000
Conversion rate (completed bookings)	10%	10%	11%	30%	13%
Avg. reservation value	\$97	5%	\$102	15%	\$112
Total reservation value	\$14,550,000		\$16,305,205		\$21,752,250
Total increased reservation value			\$2,255,250		\$7,202,250
Lower cost of sales					
Call center reservations per year	50,000	-5%	47,500	-20%	40,000
Cost per call center reservation	\$6		\$6		\$6
Cost per online reservation	\$1		\$1.55		\$1.39
Total cost of reservations	\$250,000		\$211,375		\$184,400
Savings from online bookings			\$38,625		\$65,600
Gross benefit of RIA			\$2,293,875		\$7,267,850
Cost of RIA			\$300,000		\$500,000
Net benefit of RIA			\$1,993,875		\$6,767,850
Total benefit: ROI			665%		1,354%

- El entorno de ejecución (Flash Player 9 o superior) está instalado en más del 95 % de los navegadores Web de todo el mundo



Millward Brown Septiembre 2008

Flex vs. Flash

- Flex evoluciona de Flash
- Flash era una herramienta completa de animación
- Flex es una plataforma de desarrollo
- La salida de Flex es un fichero swf compilado



- No es necesaria la detección del navegador o la detección de objetos típica de JavaScript o Ajax
- “Cross browser” compatible
- Flash player se ejecuta en MAC, Windows, Linux, Solaris
- Código compilado
- Más maduro, estructurado
- Mejor documentación, comunidades online
- Mejores capacidades para desarrollar gráficas
- Mejores capacidades de depurado
- Un “look-and-feel” mucho mas atrayente

- Aplicación-cliente, desplegado como Flash .swf
- Adobe Flex Framework
 - MXML (Macromedia XML)
 - ActionScript 3.0
 - Componentes de interfaz de usuario
 - Librerías de clases

¿Como encaja FLEX?

- Como capa de presentación
- Con cualquier lenguaje de “back-end”.
- SWF puede ser servido por cualquier servidor Web
- Puede usarse como IU o para diferentes aplicaciones
- Puede ser **la aplicación** tambien.
- Con AIR, se puede trabajar como aplicaciones de escritorio
- Se desarrolla una vez. Es portable a Web/escritorio.

FLEX para diseñadores

- MXML ~ HTML, Actionscript ~ Javascript
- Flex Builder incluye un diseñador WYSIWYG
- Soporte de CSS
- Soporte para cambiar máscaras en tiempo de ejecución
- Mantenimiento de estados
- Capas/Navegación
- Controles
- Gráficas

FLEX para desarrolladores

- Actionscript ~ Javascript + Java.
- MXML ~ Actionscript
- Sintáxis familiar
- Lenguaje dinámico
- Tipado de variables fuerte o débil
- Modificadores de ámbito
 - public, protected, private, internal

- Conectividad de datos
- Validación
- Efectos
- Data Binding
- XML Parser
- Colección de clases del SDK
- Amplia Comunidad online

- **ActionScript 3 incluye un conjunto de clases para trabajar con XML denominado E4X (ECMAScript for XML)**
- **E4X se implementó con los siguientes objetivos**
 - Simplicidad
 - Consistencia
 - Familiaridad
- **E4X incluye las siguientes clases para trabajar con XML**
 - XML, XMLList, QName y Namespace

```
<employeeList>
  <employee id="347">
    <lastName>Zmed</lastName>
    <firstName>Sue</firstName>
    <position>Data analyst</position>
  </employee>
  <employee id="348">
    <lastName>McGee</lastName>
    <firstName>Chuck</firstName>
    <position>Jr. data analyst</position>
  </employee>
</employeeList>
```

- **x.employee.(lastName == "McGee")** — Segundo nodo de empleado.
- **x.employee.(lastName == "McGee").firstName** — Propiedad firstName del segundo nodo de empleado.
- **x.employee.(lastName == "McGee").@id** — El valor del atributo id del segundo empleado.
- **x.employee.@id == 347** — Primer nodo de empleado.
- **x.employee.@id == 347.lastName** — Propiedad lastName del primer nodo de empleado.
- **x.employee.@id > 300** — Un XMLList con las propiedades de los empleados.
- **x.employee.(position.toString().search("analyst") > -1)** — Un XMLList con las posiciones.

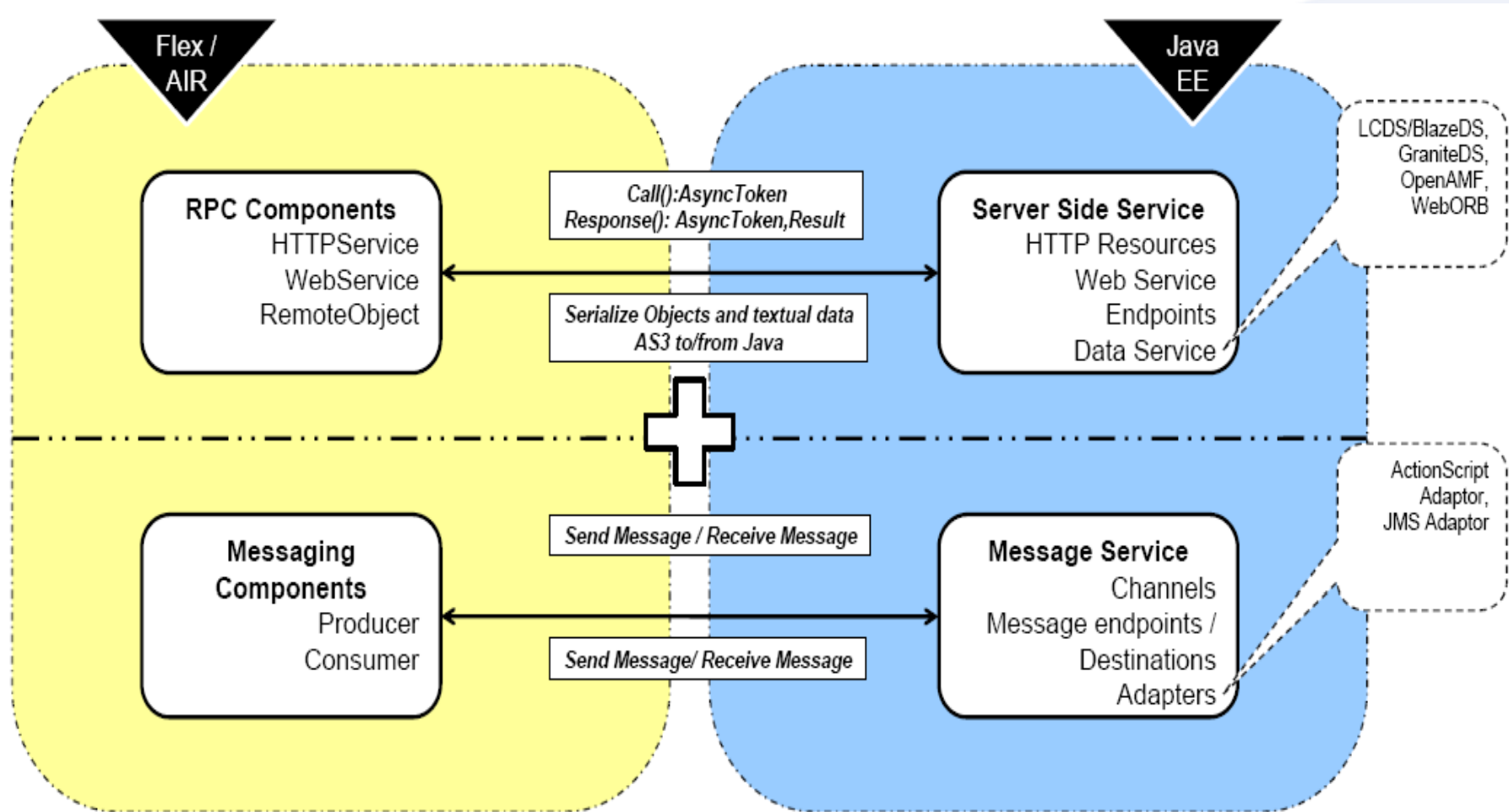
Frameworks más comunes

	Conciso	Familiar	Flexible	Herramientas	Entretenido	Total
Cairngorm	2	4	4	3	3	16
PureMVC	1	2	5	3	1	12
Swiz	5	5	3	2	4	18
Mate	4	4	4	5	5	22

Existen otros frameworks para usar con Flex, e incluso algunos específicos para usar con algunos frameworks de Java como Struts (FXStruts)

- Es la evolución OpenSource de “LiveCycle Data Services” de Adobe. (licencia LGPL)
- Implementa varios servicios de forma implícita
 - Messages
 - Remoting
 - Servicio Proxy
- Comunicación “casi nativa” entre Java y AS3

BlazeDS+Java+Flex





Se va a utilizar el **agente de transporte** inteligente denominado WebORB.

Un agente de transporte permite la comunicación entre una aplicación Flex y Java u otro lenguaje de programación.

WebORB, es una alternativa a los "LiveCycle Data Services" de Adobe, además es libre y gratuito. Además LiveCycle sólo puede utilizarse en un entorno Java y WebORB además de Java se puede utilizar con .NET y PHP ofreciendo un **rendimiento alto** y estable.

MVC DIAGRAMA

