

# **Desarrollo Rápido de Aplicaciones (RAD) con XUL y PHP5**

**Expositor: Clever Flores  
cleverflores@gmail.com**

# Panorama Actual del RAD

- ◆ Las primeras aplicaciones de empresa normalmente se hacían a través de un servidor central, con terminales tontos que se conectaban al servidor y ejecutaban desde allí las aplicaciones.
- ◆ Actualmente, las aplicaciones de gestión del tipo cliente-servidor; son las que predominan en el mercado (VB6, VFP, Power Builder). Estas herramientas se caracterizan por su facilidad para crear aplicaciones con base de datos con un par de clicks de ratón. Todas estas herramientas se ejecutan en un entorno win; y es lo más popular en RAD en el medio.

# Desventajas del Cliente Servidor

- ◆ Se ejecuta en un esquema de LAN; para WAN necesita VPNs. No se ven desde Internet.
- ◆ Necesita instalarse por cada actualización en cada uno de los usuarios del sistema.
- ◆ La mayoría solo funcionan en win.
- ◆ Implica altos costos de implementación; que al no poder pagarse, se opta por software pirata. Resultado: La mayoría de empresas usa software ilegal, en el medio.
- ◆ Si una empresa desea legalizar su software usando Linux; no puede hacerlo, porque la aplicación solo funciona en win. wine no es una solución eficiente.
- ◆ M\$ dejará de dar soporte a VB6 (obsoleto).

# Evolución y Futuro de RAD

- ◆ El desarrollo cliente-servidor se está dejando de lado para dar paso al desarrollo basado en Internet. Las nuevas herramientas de desarrollo se basan principalmente en Java y .Net
- ◆ La mayoría de estas herramientas genera código de cliente basado en DHTML; que basa el flujo de la aplicación en enlaces y botones. No es eficiente para ingreso intensivo de datos.
- ◆ En 1998 Netscape creó el proyecto Mozilla; donde posteriormente se creó XUL (Xml-based User-Interface Language) pionero en el desarrollo RIA (Rich Internet Application).
- ◆ M\$ vuelve a copiar ideas y promete el futuro de RAD en Avalon y XAML basado en RIA para Loghorn. Macromedia lanzó Flex (RIA para Flash)

# RAD con PHP5 y XUL

- ◆ PHP5 es un lenguaje que soporta la P.O.O; tiene numerosas extensiones y librerías para el desarrollo de aplicaciones web con base de datos.
- ◆ XUL es un lenguaje basado en XML para el desarrollo de Interfaces Gráficas que tiene la potestad de ejecutarse desde el escritorio o desde internet (similar al DHTML).
- ◆ La lógica de interfaz del cliente (eventos ,teclado, menús, etc) y flujo de aplicación se maneja en XUL.
- ◆ La lógica de negocio reside en el servidor web (Linux, Apache, PHP5) y de Base de Datos (MySQL, PostGreSQL). PHP5 controla las peticiones y seguridad; accede a los datos y genera respuesta basada en texto o XML (Soap o XML-RPC).

# Cliente

Interfaz Gráfica de Cliente (XUL)  
generada desde el Servidor.

Se controla con CSS, JavaScript, Overlays, DOM, RDFs, y Templates.

Para crear componentes Gráficos se usa XBL

Para hacer las llamadas al servidor se usa el objeto XmlHttpRequest que recogerá la respuesta de los Servicios Webs (XML) y/o cadenas generadas por otros procesos

# Servidor

Capa de Modelo (Lógica de Negocio)  
Base de Datos  
Lógica Nativa: Funciones, Triggers.  
Lógica de Lenguaje:  
DataObjects en PHP5 (Propel, PDO)  
Metadata (Descripción de Datos)

Capa de Control, validación de la petición; llamada al modelo, despliegue de servicios web.

Capa de Autorización y Seguridad.  
Usuarios, Perfiles, Control de Sesiones

Capa de Servicios Web PHP5 (SOAP - WSDL) Invocada por el sistema o para integrar sistemas heterogéneos, B2B

**Servidor Web - Apache (Http + Https)**

# XPFE *(Mozilla Cross-Platform Front End)*

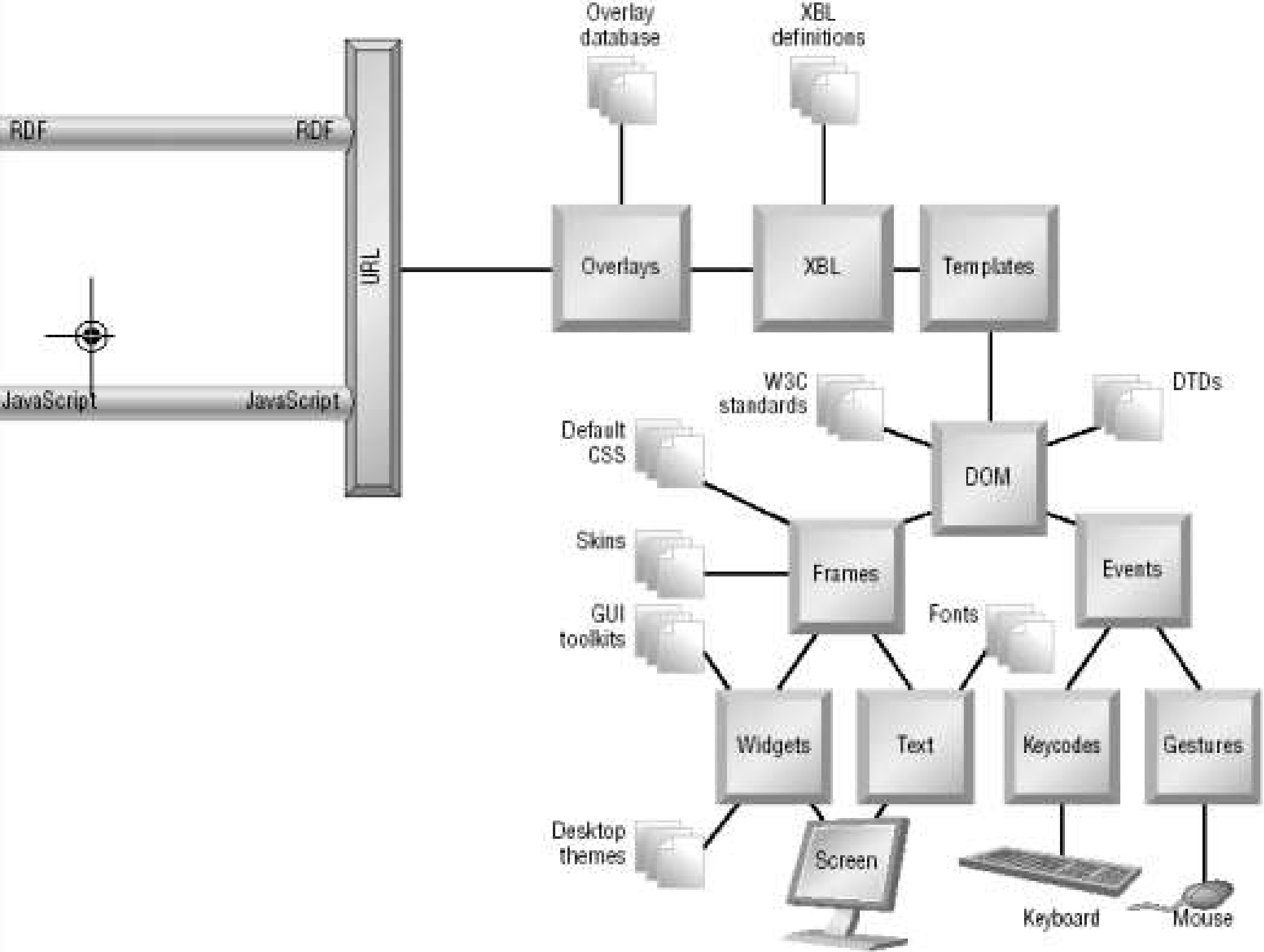
- ◆ Cuando nació el Proyecto Mozilla; se vió la necesidad de crear un framework para el desarrollo del browser y para aplicaciones independientes de él que pudieran ejecutarse en diferentes sistemas operativos.
- ◆ XPFE (XUL+CSS+JavaScript) fué diseñado para resolver este problema y permitir a los ingenieros crear una interfase que funcionara igual en cualquier sistema operativo.
- ◆ Para comprender como funciona XPFE, debemos mirar como esos diferentes componentes actúan juntos. JavaScript crea la funcionalidad para una aplicación Mozilla, CSS formatea el Look&Feel y XUL crea la estructura de la aplicación. Todas estas tecnologías son interpretadas por el núcleo de la plataforma (Gecko)

# Componentes de XPFE

- ◆ **XML-bases User-interface Language (XUL):** Usado para crear la estructura y el contenido de una aplicación.
- ◆ **Cascading Style Sheets (CSS):** Usado para crear el look and feel de una aplicación.
- ◆ **JavaScript y DOM:** Usado para crear la funcionalidad de una aplicación, pudiendo ser en el futuro, otros lenguajes de scripting como Python, Perl o Ruby.
- ◆ **Cross-Platform Install (XPInstall):** Usado para empaquetar e instalar aplicaciones en cualquier plataforma.

# Componentes de XPFE

- ◆ **eXtensible Binding Language (XBL):** Usado para crear widgets reusables en combinación con XUL y JavaScript.
- ◆ **XPCOM/XPCConnect:** Usado para permitir potencias a JavaScript o acceder y reutilizar librerías C y C++.
- ◆ **XUL Templates:** Usado para crear un framework que importe datos dentro de una aplicación en combinación con RDF y XUL.
- ◆ **Resource Description Framework (RDF):** Usado para almacenar y transmitir información. Generalmente mirado como uno de los aspectos más complicados de XPFE.



# El Proyecto MAB

- ◆ **MAB (Mozilla Amazon Browser):** Es un proyecto hecho en XUL; que permite hacer consultas a la base de datos de Amazon; accediendo a ellos a través de los servicios webs que ofrece en forma pública.
- ◆ MAB fue creado para poder ejecutarse en forma local mediante un paquete de instalación (xpi) o en forma remota desde un servidor web; similar a una aplicación web común.
- ◆ Usa librerías de javascript para manejar las peticiones y respuestas en forma directa(local) o por un script proxy (remoto) a los servicios web de Amazon. La respuesta (XML) e evaluada y volcada a por al datagrid de resultados.

# Enlaces de Interés

- ◆ [www.xulplanet.com](http://www.xulplanet.com) (Tutorial y Manual de Referencia en Línea)
- ◆ [www.mozilla.org/projects/xul/](http://www.mozilla.org/projects/xul/) (Referencia Oficial)
- ◆ [www.mozdev.org](http://www.mozdev.org) Repositorio de Proyectos en XUL y extensiones de Mozilla  
[books.mozdev.org](http://books.mozdev.org) (Libro de XUL)
- ◆ [xul.sf.net](http://xul.sf.net) Recursos de XUL (librerías y libro)
- ◆ [www.zend.com/php5/](http://www.zend.com/php5/) Tutoriales de PHP5 y librerías
- ◆ [php.net](http://php.net) Sitio oficial de PHP y el mejor manual de referencia
- ◆ [pear.php.net](http://pear.php.net) Librerías Oficiales del Proyecto PHP (incluidos algunos de PHP5)