Ajax

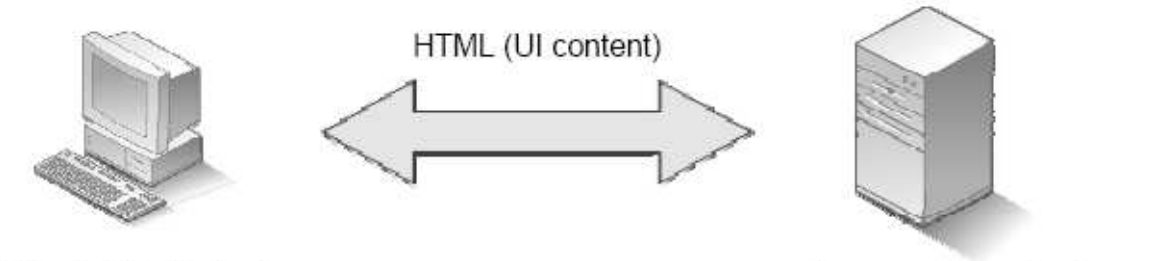
Technology review

AJAX (*Asynchronous JavaScript And XML)*

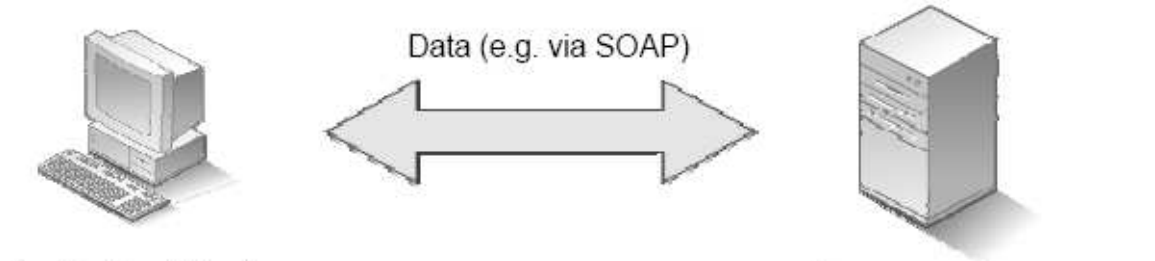
- XHTML (o HTML) y hojas de estilos en cascada (CSS) para el diseño que acompaña los datos
- Document Object Model (DOM) accedido con un lenguaje de scripting por parte del usuario
  - Especialmente implementaciones ECMAScript como JavaScript y JScript, para mostrar e interactuar dinámicamente con la información presentada.
- El objeto XMLHttpRequest para intercambiar datos asincrónicamente con el servidor web.
  - En algunos frameworks y en algunas situaciones concretas, se usa un objeto iframe en lugar del XMLHttpRequest para realizar dichos intercambios.
- XML para la transferencia de vuelta al servidor
  - Cualquier formato puede funcionar, incluyendo HTML preformateado como texto plano o JSON.

Cambio de Paradigma con AJAX

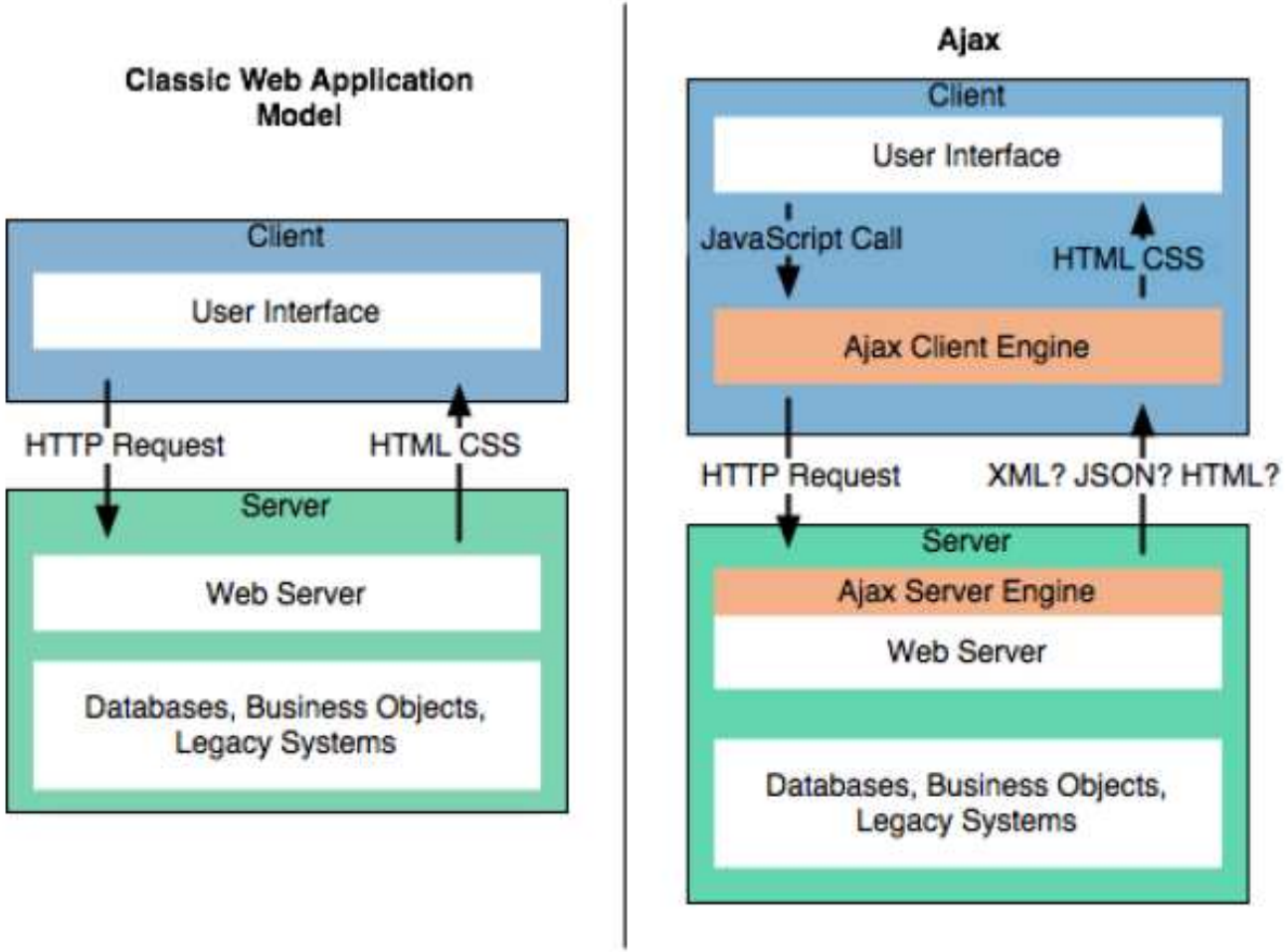
Web-Based Application Model



Browser-Based Application Model



Arquitectura de AJAX vs. Aplicación web tradicional



Funcionamiento de componentes en AJAX

JavaScript define las reglas de negocio y el flujo del programa.

COM y CSS permiten a la aplicación reorganizar la apariencia en respuesta a los datos obtenido en “background” desde el servidor por medio del objeto XMLHttpRequest

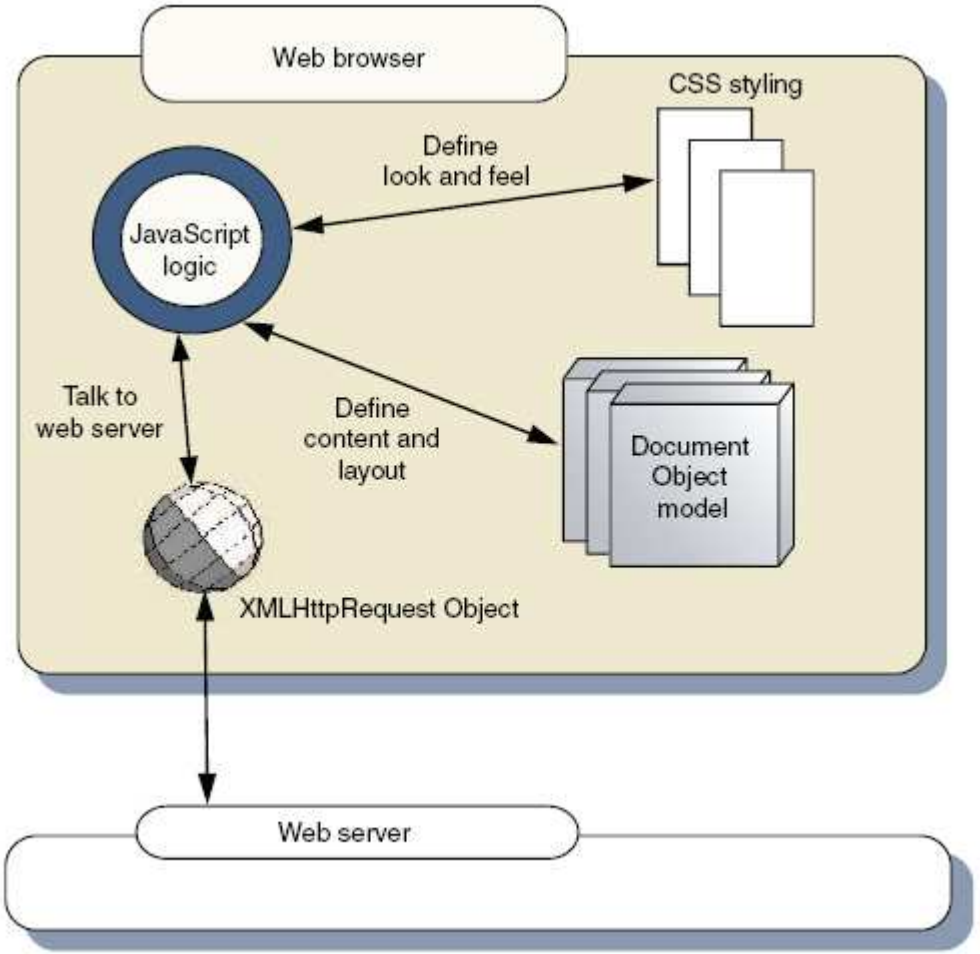
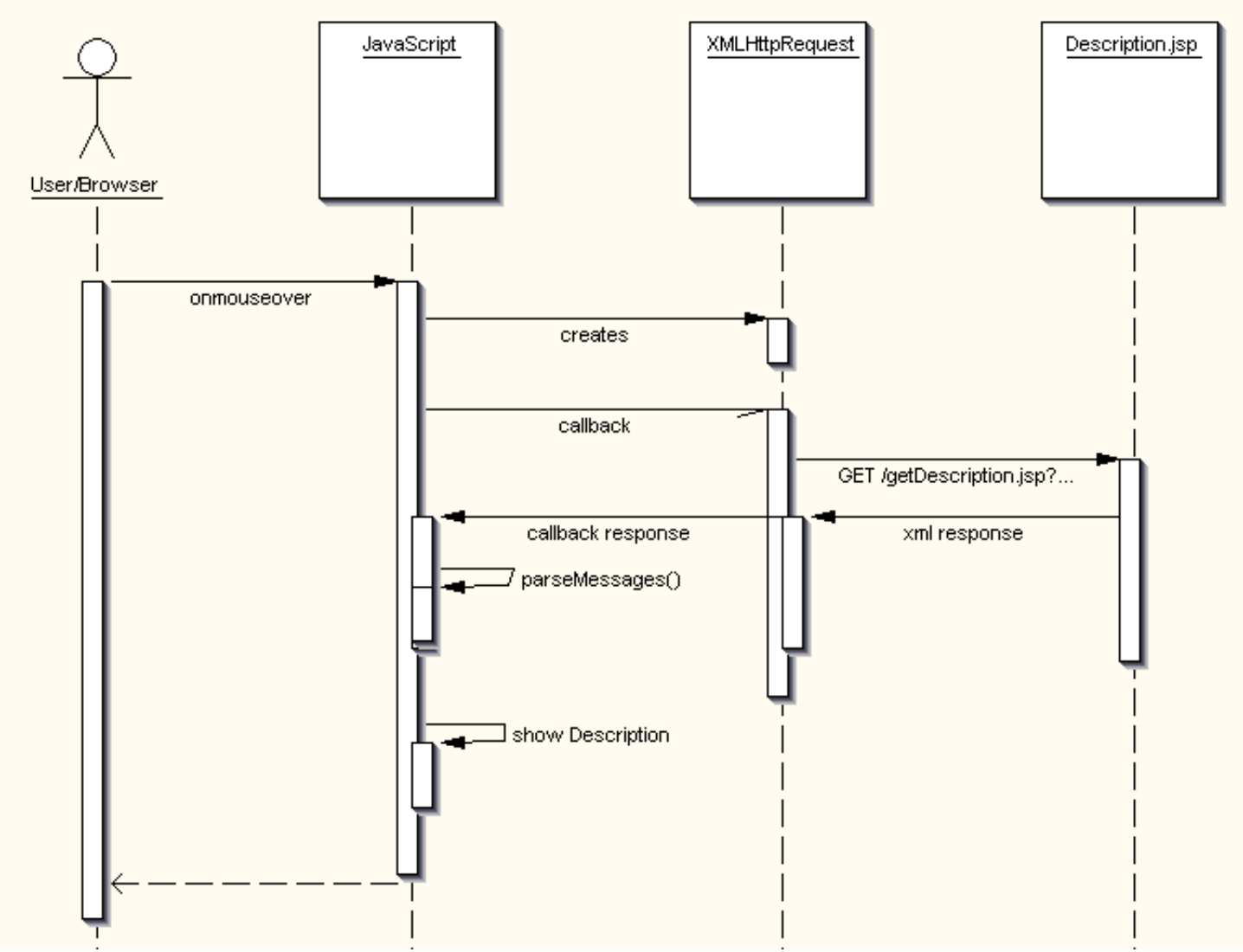


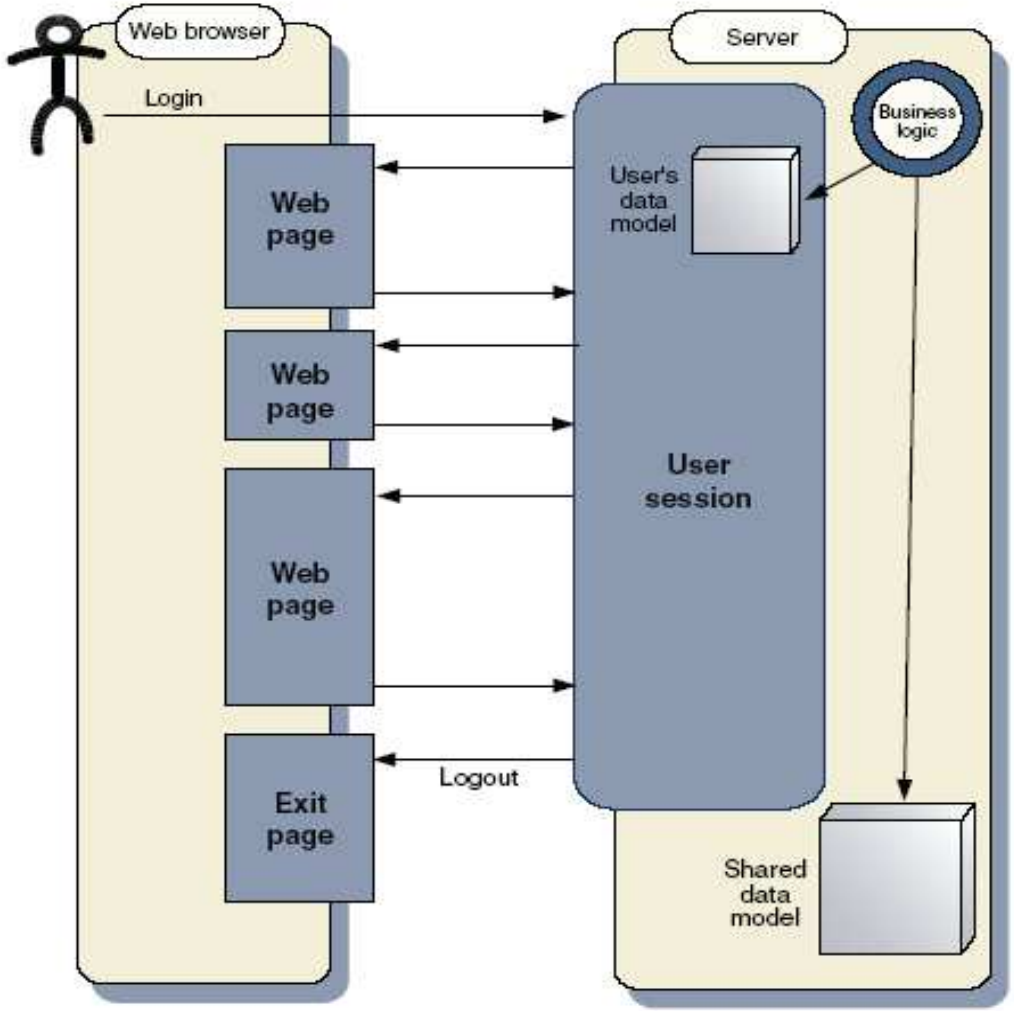
Diagrama de Secuencia de una invocación AJAX



4 principios para AJAX

1. *El browser hospeda parte de la aplicación:*
   - En una página Web tradicional, el browser es una terminal “tonta” porque no conoce nada acerca de cómo interactúa el usuario con la aplicación.
   - Toda la información es mantenida en el servidor Web, en la sesión del usuario

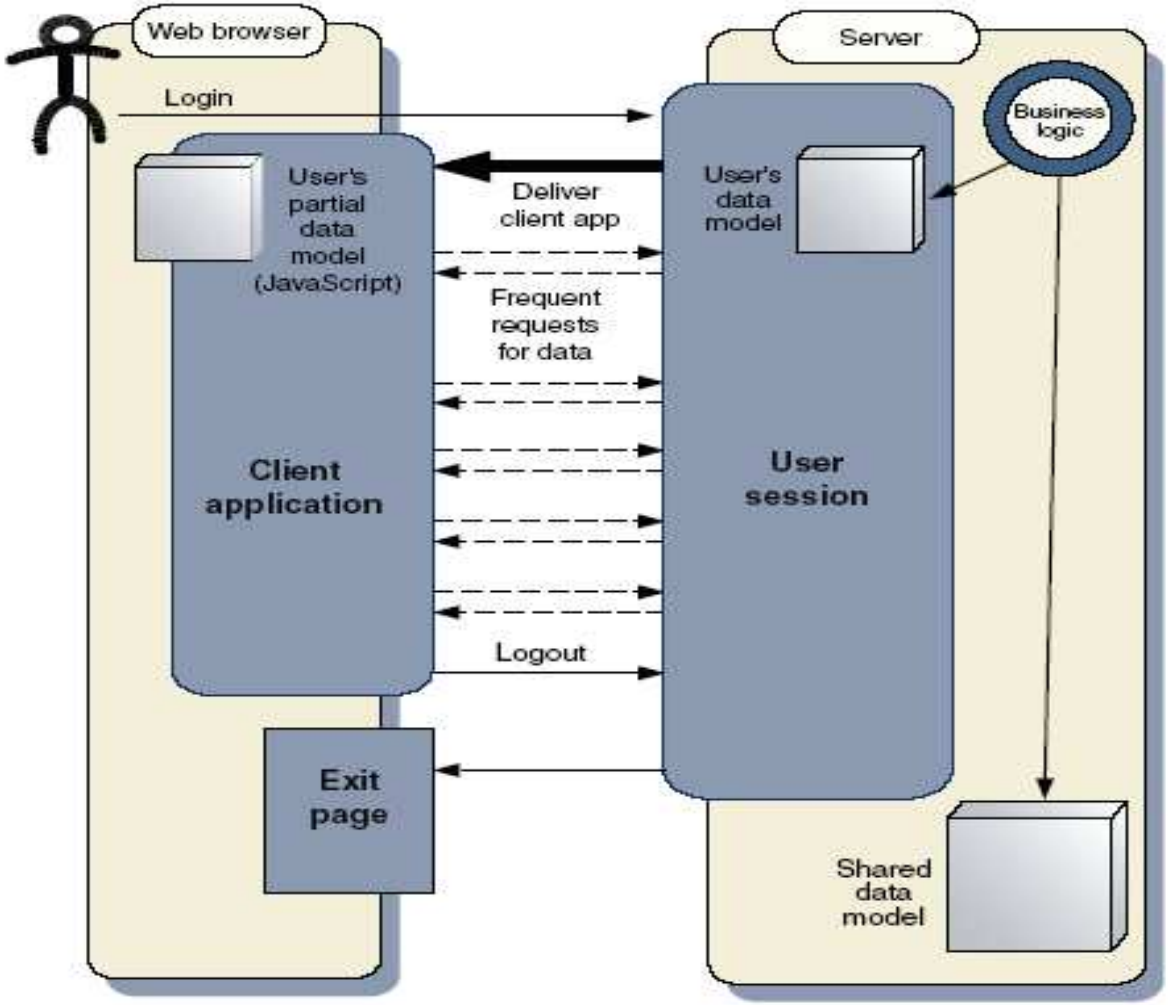
Aplicación Web Tradicional



El browser hospeda la aplicación, no la contiene (cont)

- Una aplicación AJAX mueve algo de la lógica hacia el browser
- Cuando un usuario se logea, un documento más complejo que simple HTML es enviado al navegador, del cual una gran parte es código en JavaScript
- El documento permanece en la sesión del usuario y probablemente cambiará mientras el usuario interactúa con él
- La lógica embebida en el documento sabe como responder a las entradas del usuario y decidir si manejar dichas entradas o pasar el requerimiento al servidor Web (quien tiene acceso a la base de datos por ejemplo) o una combinación de ambos.

Aplicación AJAX



Principios (cont)

2. *El servidor entrega datos, no contenido:*
   - En una aplicación Web tradicional el servidor Web entrega una mezcla de datos y contenido en cada paso
   - Por ejemplo en un carrito de compras, si un usuario adiciona un ítem, lo que se debe hacer es actualizar es la cuenta total. No reenviar el contenido de todos los ítems actuales en el carrito, ni las listas de navegación, estilos de presentación, etc.
   - Una aplicación AJAX en cambio retorna un fragmento de JavaScript, o un pequeño documento XML.

Principios (cont)

3. *La interacción con el usuario puede ser continua y fluida:*
   - En una aplicación Web tradicional mientras la página es enviada, el usuario está en el limbo. Aunque la pagina vieja está aún visible, el browser le permite al usuario hacer click en los enlaces visibles con resultado impredecibles.
   - La aplicación AJAX envía los datos asincrónicamente, lo que permite al usuario hacer varios clicks rápidamente (como por ejemplo cargar varios ítems en un carrito de compras simultáneamente) y saber que está pasando realmente.

Principios (cont)

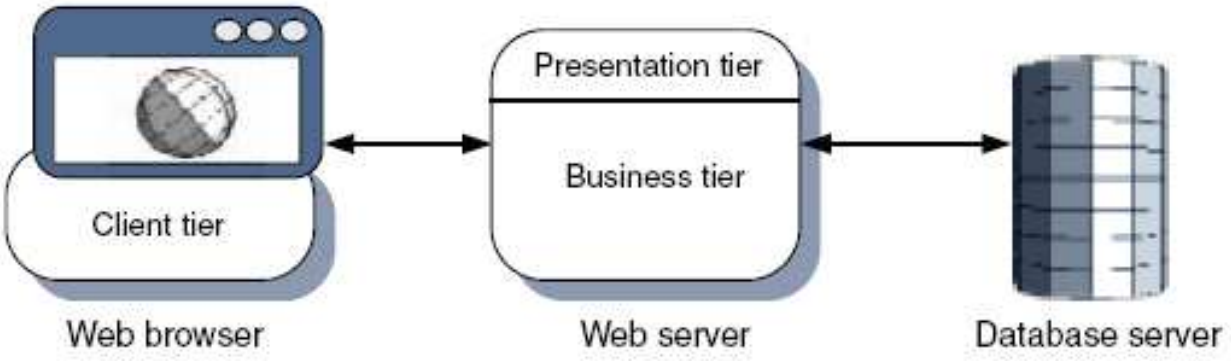
- *Requiere disciplina para codificación (patrones)*
  - Una aplicación AJAX es una pieza de código compleja que se comunica eficientemente con el servidor mientras el usuario trabaja.
  - Aunque tiene claras diferencias con una aplicación tradicional (basada en páginas Web) las diferencias no son tan dramáticas por lo que requiere conocer muy bien donde hacer énfasis en las fortalezas de usar los principios anteriores.

Consideraciones sobre AJAX

- Ventajas
  - Portabilidad, soportada entre capas de la arquitectura de la aplicación
  - RIA (Rich-Internet-Applications), aplicaciones Web más ricas en la interacción con el usuario
  - Manejo de datos estructurado
  - Mejor desempeño (no para todos los casos)
- A tener en cuenta
  - Arquitectura de la aplicación
  - Lógica en Javascript
  - El framework a utilizar

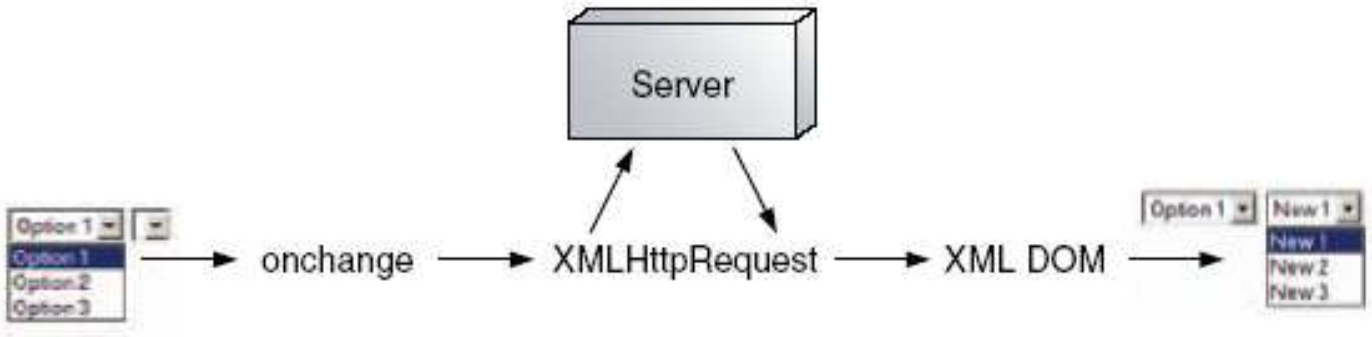
Consideraciones (1)

Portabilidad: Una aplicación AJAX permite reasignar algunas de las responsabilidades de la capa de presentación desde el servidor hacia el navegador, en una nueva entidad llamada ahora la capa cliente de la aplicación



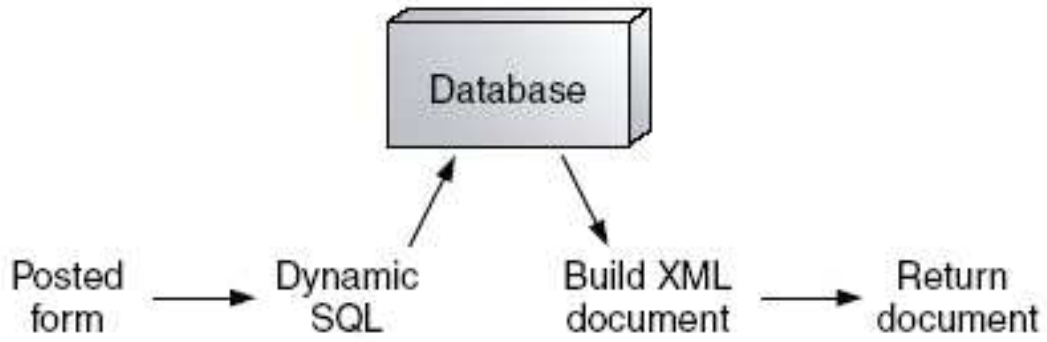
Consideraciones (2)

RIA: En la arquitectura cliente-servidor de una aplicación AJAX, la interacción dinámica con el servidor permite construir la presentación de la aplicación en respuesta a la navegación o entradas del usuario

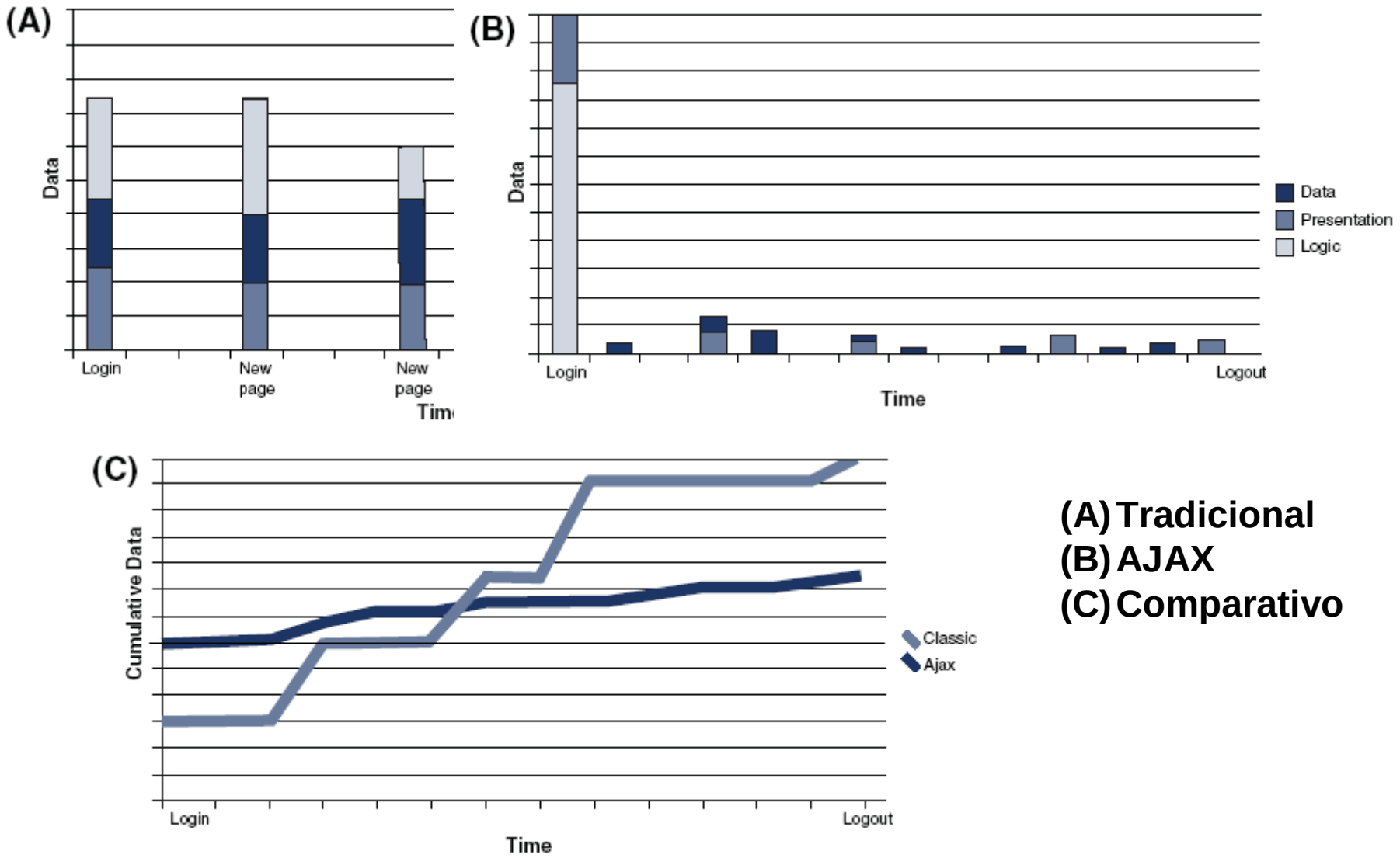


Consideraciones (3)

Datos estructurados: El acceso a los datos mediante SQL dinámicos permite construir documentos XML que mejoran su representación y manipulación por parte de la aplicación en el lado cliente



Consideraciones (4): Desempeño



(A) Tradicional
(B) AJAX
(C) Comparativo