

Web Application Solutions: A Designer's Guide



As the Web continues to extend its reach into our daily lives, an increasing number of our interactions will happen online. The practical implication of this for interface designers is lots of Web application projects that cover everything from filing taxes to sharing photos.

“The fundamental purpose of Web applications is to facilitate the completion of one or more tasks” [Bob Baxley 1]. But depending on the type and complexity of the tasks involved, different technical solutions may be better suited to enable the specific interactions each product requires. Flash, Java applets, DHTML, Active X, Smart Clients, Java Web Start, SVG— what do you choose and why? What types of interactivity and visual presentation does each technology enable? What does each limit?

As Web application interface designers, these are questions we encounter time and time again. As a result, we decided to document what we’ve learned and research what we didn’t know about the opportunities and limitations that characterize some of the most popular Web application presentation layer solutions available today. We evaluated each solution against a consistent set of criteria and described it with a concise definition, set of examples, and references that enable further analysis. We also shared our findings with a team of expert reviewers (page 18) to ensure we were on the right track. The end result is this designer’s guide.

For your convenience, each solution we’ve evaluated is presented on a single-page snapshot that helps designers, product managers, and business owners make an informed decision for their Web application’s front-end technology. Of course, this guide is not meant to be a substitute for consulting with knowledgeable programmers and system architects. We simply want to prepare you for those discussions.

The right Web application presentation layer needs to meet your specific business, user, and technology needs and multiple solutions can exist per business and per product. We hope this guide helps steer you in the right direction.



Luke Wroblewski
Principal, LukeW Interface Designs

luke@lukew.com



Frank Ramirez
Principal, Ramirez Design LLC

frank@ramirezdesign.com

This guide is not intended to define the multitude of presentation layer solutions available for Web applications. We're focused on explaining the trade-offs. However, an overview of the grouping and evaluation criteria we've used to describe each solution is required to ensure we present accurate comparisons.

Web Application (Web-based application)

Web applications are Web-accessible (deployed and/or accessed through a Web browser) Web-connected (utilize a http connection for information retrieval or display), and task-oriented (beyond the simple browsing of information) software.

Due to the variance and subjectivity inherent in defining types of Web applications, we've established a continuum from thin client to rich client on which each Web application solution can be plotted. Many solutions (and their various implementations) have characteristics of more than one group. For example, DHTML with XMLHttpRequest (commonly referred to as AJAX) is part thin client and part Rich Internet Application (RIA). Likewise, some desktop applications really straddle the line between Rich Internet Application and rich client (a.k.a. a thick client).

Lastly, while you could think of any Internet-enabled desktop application (like Apple's iTunes or Yahoo's Instant Messenger) as a Rich Internet Application, we've left this type of software out of the discussion. Applications that require a traditional client-side install process (regardless whether or not the install file can be downloaded from a Web server) are not evaluated within this guide.

Thin Client

Web applications that utilize the Web browser for security, state management, and script execution (run-time). Most data processing and storage occurs on a remote server and not a user's local machine. Server request and response mostly occurs through the http protocol.

The primary benefits of thin clients are wide reach (accessible by anyone with a Web browser), open development platform (built on popular open standards), no footprint (quick download, no artifacts on user machines beyond browser cookies), and deployment/manageability (distributed and maintained from a central source).

Thin Client Solutions:

- HTML, XHTML
- HTML, JavaScript, and CSS (DHTML)
- DHTML with Remote Scripting via iFrame
- DHTML with XMLHttpRequest (AJAX)

Rich Internet Application (RIA)

Web applications that enable richer, locally processed user interactions (fluid animation, multimedia content, real time validation, etc.) and advanced remote messaging (Java Objects, Web Services, etc.). Sun's Java plug-in and Macromedia's Flash are two of the most common RIA run-times. RIAs can only run in a Web browser within embedded plug-ins.

The HTTP communication layer used by Thin Client applications technically does not support guaranteed message delivery, does not guarantee the order of message delivery, and does not support server-initiated communications. Many RIA communication layers provide reliable messaging and enable server "pushes" of content.

Thin Clients that utilize asynchronous JavaScript-driven server requests and responses and advanced DHTML user interactions (a combination of technologies frequently referred to as AJAX) are sometimes considered JavaScript RIAs.

A Thin Client can embed (integrate through a Web browser plug-in) an RIA (i.e. a Flash module in a DHTML application).

Rich Internet Application Solutions:

- Flash 6 (and higher)
- Flash with Flex or Laszlo Presentation Server
- Java Applets
- Active X

Rich Client (smart client, desktop client)

Web-connected applications that do not run within the Web browser and can be delivered as compiled code. Rich Clients leverage local processing to enable rich interactions and can utilize Web Services to connect to distributed data sources and auto-update. Unlike Thin Client applications, Rich Clients can be used offline and more easily integrate with local hardware and software.

A Rich Client can embed a Thin Client (i.e. a desktop application with an integrated Web browser within which a thin client application can run).

Rich Client Solutions:

- Windows Smart Clients
- Java Web Start

Upcoming, Interesting, or Relevant Solutions

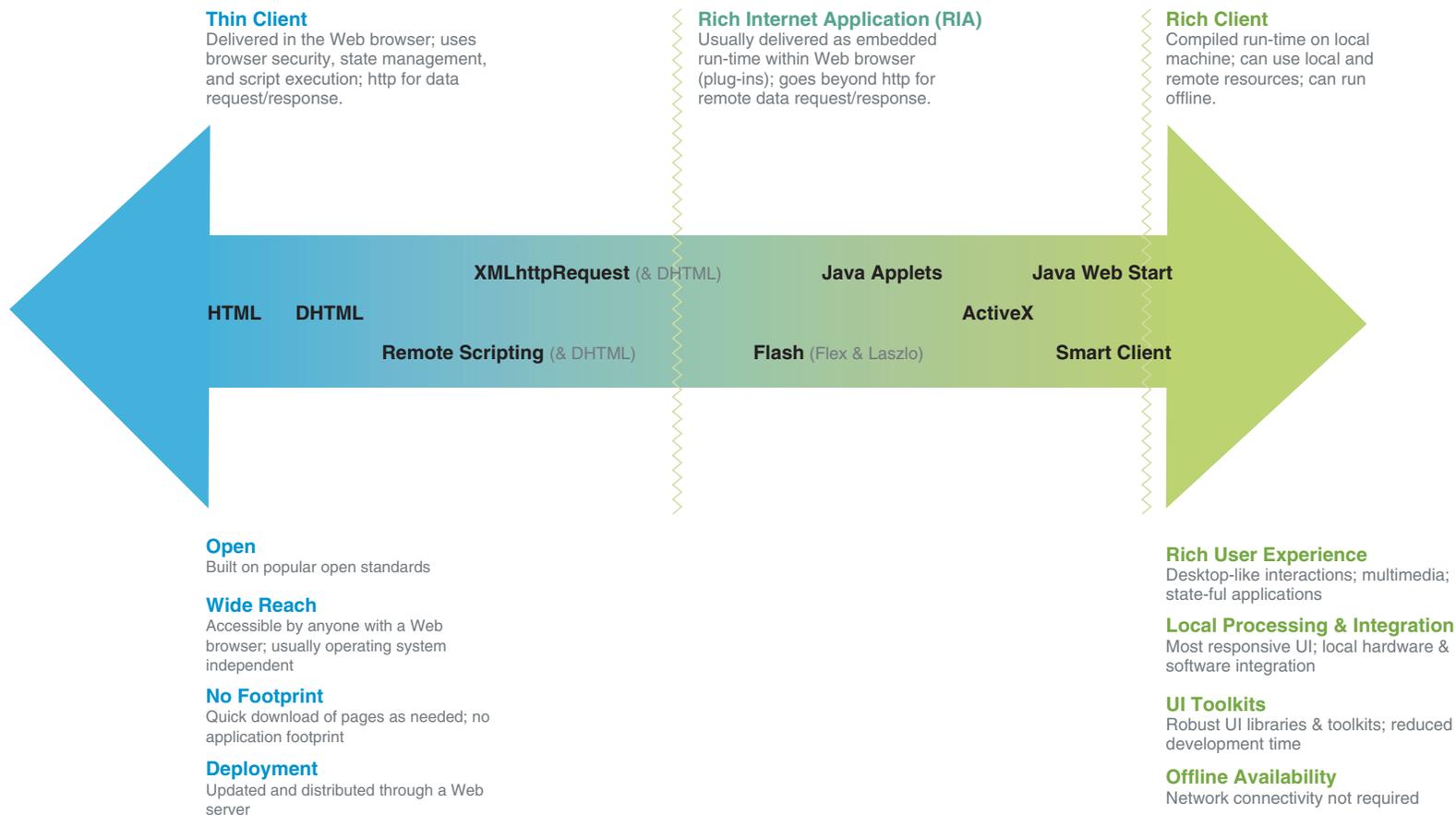
While this guide focuses on some of the most popular Web application solutions available today, there are a number of relevant technologies that may be considered for specific uses (i.e. PDF for data entry) or future projects (i.e. XUL or XAML).

Examples include

- Canoo Ultra Light Client (ULC)
- Java WebCream
- PDF
- XSLT
- SVG
- Macromedia Central
- Nexaweb
- Xforms
- XAML

The primary benefits of a thin client are reach (anyone with a Web browser can use it) and deployment (can be updated and distributed through a Web server). The primary disadvantages are limited interaction options and typically slower response times. Rich Internet Applications and Smart client technologies enable richer (desktop-like) interactions, more sophisticated messaging, and prevent server request/responses from having to rewrite entire pages. They also maintain some of the deployment and updating benefits of thin clients. The chart below overlays popular Web application technology solutions on a continuum from thin to rich client. Most solutions fall within a range on the continuum.

Each Web application solution falls on a continuum from thin client to rich client, though some solutions really "straddle the line". A Smart Client can even embed a Thin Client (within a Web browser inside a rich client) and a thin client can embed an RIA (i.e. Flash module in a DHTML application).



Each Web application solution was evaluated against the following criteria.

User Experience

What user interactions are possible? (drag and drop, fluid animation, auto-saving, real-time validation, non-linear workflows, integrated audio and video, etc.) Does each user action require a full-page rewrite (as in the case of HTML/http solutions)? How well integrated is the user interface with the Web browser environment (Back, History, etc.) and/or local operating system (local hardware and software integration)? Can application states be easily maintained?

Deployment & Reach

How can users access the application? Do they need specific Web browsers, Web browser plug-ins, Web browser settings, operating systems, devices (mobile, etc.), or local applications? How large is the reach of the application? Can search engines index the content within the application? How is the application deployed (install, download, footprint, etc.)?

Processing

How much application logic can reside on the client-side? What type of action requires a trip to the server? What is the impact to the UI for a server request? How fast are the initial load and subsequent response times of the user interface? What kind of remote messaging is possible (http, Web Services, Java objects, etc.)? Can server requests run uninterrupted (in the background) while the user focuses on other tasks?

Interface Components & Customization

Does the solution include an easily accessible library of UI elements? Are the elements easily customizable? Is there a WYSIWYG development environment (IDE) that leverages the UI library? Can the interface be easily transformed for different devices (i.e. mobile)? Can the user interface be built with grid or flexible layouts, or both?

Back-end Integration

What back-end technologies enable this front-end solution? What is required on the back-end? Does the solution increase or decrease server loads?

Unique Features

Are there unique security concerns? Is application logic exposed? Can the application run offline?

Future Proofing

What level of support is available for the solution? If the technology is proprietary, does the vendor have a proven record of product support? Does the solution conform to existing standards?

Staffing & Cost

What are the costs required to build out the solution? Are technology experts easily found? What tools exist to reduce costs? How long is a typical development cycle?

Overview:
 HTML is a document markup language for structure, presentation and monodirectional linking. XHTML is a reformation of HTML that is XML-compliant. Data requests are sent through the Hypertext Transfer Protocol (http) standard.

+ Benefit - Drawback • Neutral

User Experience	Deployment & Reach	Processing	Components & Customization	Back-end Integration	Unique Features	Future Proofing	Staffing & Cost
<ul style="list-style-type: none"> + Elements that can be specified in HTML are familiar to the largest amount of Web users (including back, history, bookmarking, etc.) • Small file sizes: quick download times - Sufficient for relatively static content and trivial tasks - Limited to form entry, data display, and link navigation - Integration with local resources (hardware & software) is nearly impossible 	<ul style="list-style-type: none"> + Has the biggest reach- can be rendered by any browser. is standardized and available to users running all types of software. Effective for users with slow connections. + Search engines can index content 	<ul style="list-style-type: none"> • Application states can be achieved through cookies, server-side apps, and/or URLs - Client requests require full reloads of Web pages to update displays increasing download times and server loads - HTTP requests do not support guaranteed message delivery, guaranteed order of message delivery, nor server-initiated communications (push) 	<ul style="list-style-type: none"> + HTML can be styled with CSS (Cascading Style Sheets) • Flexible layouts are possible, fixed grid are difficult to maintain consistently across browsers. - Components limited to those available through HTML markup 	<ul style="list-style-type: none"> + Can be delivered with JSPs, Servlets, ASP, ASP.NET, JSP, ColdFusion, PHP, and more. 	<ul style="list-style-type: none"> • Code exposed - easy to copy 	<ul style="list-style-type: none"> + Relatively easy to port to other devices such as PDAs, phones, etc. + Code is likely to be forward-compatible due to standardization. 	<ul style="list-style-type: none"> + Relatively easy to find people with HTML, XHTML programming skills. + WYSIWYG development environments and content management systems (CMS) widely available

Examples:

- Craigslist – online local classified ads
- Bloglines

Read More:

- <http://www.w3.org/MarkUp>
- <http://www.oreilly.com/catalog/html4>

Overview:
 DHTML is not a programming language, but the combination of HTML, JavaScript (or VBScript), and Cascading Style Sheets (CSS). JavaScript is a cross-platform, object-oriented scripting language. JavaScript is also small, lightweight, and was designed to be embedded in other products and applications, such as web browsers. Inside a host environment, JavaScript can be connected to the objects of its environment to provide programmatic control over them [2].

+ Benefit - Drawback • Neutral

User Experience	Deployment & Reach	Processing	Components & Customization	Back-end Integration	Unique Features	Future Proofing	Staffing & Cost
<ul style="list-style-type: none"> • See HTML... + Elements that can be specified in HTML are familiar to the largest amount of Web users + Animation (repositioning and motion), drag and drop, layering, and style changes are enabled for interface elements + Inconsistent support for transparency and layering + Usually has faster download times, though large JavaScript apps can be slow to load - No easy way to integrate audio & video content 	<ul style="list-style-type: none"> • See HTML... + Most content can be indexed by search engines. • Works with most modern browsers (4.x browsers and later) - JavaScript may be turned off by users and administrators 	<ul style="list-style-type: none"> • See HTML... + Reduced server loads may be achieved through increased local processing (dynamic hiding, revealing, repositioning, and rewriting of interface elements) - Logic is delivered to browser with HTML page and can be dissected. As a result, secure data cannot be sent this way (passwords, serial numbers etc.) 	<ul style="list-style-type: none"> • See HTML... + Some UI components can be custom made. ie: sliders + Toolkits: dynamic drive, dynarch.com, softplex.com, navsoft.com, and many more 	<ul style="list-style-type: none"> • See HTML... 	<ul style="list-style-type: none"> • See HTML... 	<ul style="list-style-type: none"> • See HTML... + Code is likely to be forward-compatible due to open standards supported by multiple browser vendors. For example: W3C DOM (Document Object Model) activity 	<ul style="list-style-type: none"> • See HTML... + Small size reduces bandwidth costs • Relatively easy to find people with DHTML programming skills. - Higher development costs due to sophisticated, branching code (required to support multiple browsers) - Common UI features can be difficult to implement (tabbing, tabular data, tree views, drag and drop, saving, etc.)

Examples:

- TurboTax –file your tax returns (<http://www.turbotax.com>)
- Yahoo! Mail
- Hotmail

Read More:

- Dynamic Duo –Cross browser Dynamic HTML [http://www.jalix.org/ressources/internet/dhtml/_dynduo/dynduo/]
- What is DHTML? [<http://webmonkey.wired.com/webmonkey/geektalk/97/39/index3a.html>]

DHTML with Remote Scripting via iFrame

Overview:

See DHTML Overview... An iframe is an inline frame (placed inside another HTML page) that contains a separate HTML page with its own URL. DHTML can use hidden iFrames to get information from the server without refreshing the page [Ashley 3]

+ Benefit - Drawback • Neutral

User Experience	Deployment & Reach	Processing	Components & Customization	Back-end Integration	Unique Features	Future Proofing	Staffing & Cost
<ul style="list-style-type: none"> • See DHTML... + Applications are more responsive - Linking to, saving, and book-marking content can be tricky. - Content may be incorrect when users click reload and back buttons 	<ul style="list-style-type: none"> • See DHTML... + Works with most modern browsers (Internet Explorer 5.5+, Netscape 6+, and Mozilla 1+) - Dynamic content won't be indexed by search engines 	<ul style="list-style-type: none"> • See DHTML... + iframes can be dynamically refreshed with server-side content with Javascript and a DOM ID - Only enables content to be loaded in a pre-defined area (frame) on a Web page (though iframe content can be put in a DIV and moved procedurally) 	<ul style="list-style-type: none"> • See DHTML... 	<ul style="list-style-type: none"> • See DHTML... + reduced server loads due to more processing on client-side 	<ul style="list-style-type: none"> • See DHTML... 	<ul style="list-style-type: none"> • See DHTML... 	<ul style="list-style-type: none"> • See DHTML... - Higher development costs due to sophisticated, branching code (required to support multiple browsers) - Extra effort required for good accessibility (ie: screen readers)

Examples:

- <http://farechase.yahoo.com> (Flight Search only. Hotel search uses xmlhttprequest)

Read More:

- <http://www.cs.tut.fi/~jkorpela/html/iframe.html>
- Remote Scripting with iframe [<http://developer.apple.com/internet/webcontent/iframe.htm>]

DHTML with XMLHttpRequest (aka AJAX)

Overview

See DHTML Overview... XMLHttpRequest is an object that allows (compatible) web clients to submit and retrieve XML data from a server without reloading the page. JavaScript processes the data in the background and renders it via the Document Object Model (DOM) [Garret 4]. This can happen asynchronously, allowing the user to continue with his task while the data gets processed and updated.

+ Benefit - Drawback • Neutral

User Experience	Deployment & Reach	Processing	Components & Customization	Back-end Integration	Unique Features	Future Proofing	Staffing & Cost
<ul style="list-style-type: none"> • See DHTML... + Enhance user experience with better, faster forms. Enables real-time updates and validations as user moves from field to field or even after each character entry. - Content may be incorrect when users click reload and back buttons - Linking to, saving, and book-marking content can be tricky - Enhanced functionality within a web browser can be difficult to communicate 	<ul style="list-style-type: none"> • See DHTML... + XMLHttpRequest is available in Netscape/Mozilla, Internet Explorer 5+, and Safari browsers (Safari & Opera can only support a subset of requests) - Content requested through XMLHttpRequest object cannot be indexed by search engines or read by some screen readers. 	<ul style="list-style-type: none"> • See DHTML... + Further reduction in server loads due to processing on the client side and pure data requests (without presentation layer) after the initial load. - No socket connections - server can not initiate communications to client app 	<ul style="list-style-type: none"> • See DHTML... + Due to large interest in the web development community, DHTML toolkits with XMLHttpRequest are sure to come soon. (For now, see SAJAX and JSON-RPC) 	<ul style="list-style-type: none"> • See DHTML... 	<ul style="list-style-type: none"> • See DHTML... 	<ul style="list-style-type: none"> • See DHTML... • XMLHttpRequest is an API that belongs to Microsoft-could be depreciated. However W3C has initiated a similar "XML Load and Save Specification" in DOM Level 3. 	<ul style="list-style-type: none"> • See DHTML... - Higher development costs due to sophisticated, branching code (required to support multiple browsers) - Extra effort required for good accessibility (ie: screen readers)

Examples:

- www.gmail.com (load JavaScript engine at start of session)
- map.search.ch
- maps.google.com
- www.flickr.com (edit the titles of your photos in situ)
- www.fiftyfoureleven.com/resources/programming/xmlhttprequest/examples

Read More:

- <http://developer.apple.com/internet/webcontent/xmlhttpreq.html>
- XMLHttpRequest for the masses: <http://www.allinthehead.com/retro/241/xmlhttprequest-for-the-masses>
- <http://www.standards-schmandards.com/index.php?2005/03/01/16-ajax-and-accessibility>
- Simple Ajax Toolkit: <http://www.modernmethod.com/sajax/>
- JSON-RPC: <http://oss.metaparadigm.com/jsonrpc/>

Flash 6 (and higher)

Overview:

Macromedia's Flash 6+ is a multi-platform plug-in installed in most Web browsers as well as an application for developing content, interfaces, and applications. Flash 6+ supports rich vector-graphics, animation, and the ability to build advanced web applications using scripting, forms, XML, and server socket connections. Flash 5 has limited support for some of these feature sets.

+ Benefit - Drawback • Neutral

User Experience	Deployment & Reach	Processing	Components & Customization	Back-end Integration	Unique Features	Future Proofing	Staffing & Cost
<ul style="list-style-type: none"> + Flash offers a rich set of features for engaging interactions including drag and drop, animation, transparency, layering, audio and video streaming. + Enables real-time updates and validations as user completes form fields - Potential latency due to relevantly large initial download and plug-in launch - Not fully integrated with browser environment (reload, back buttons and bookmarks). Workarounds exist 	<ul style="list-style-type: none"> + According to Macromedia, Flash 6+ is installed on 95.1% of Internet-enabled desktops in the US, over 96% in Europe and Canada and 92% in Asia. + Applications delivered through the Flash player can run identically on all the major operating systems + Near seamless upgrade process for plug-in • Traditional tracking and metrics can be problematic; but has ability to track specific states - Flash content is not indexed by search engines. (workarounds exist) 	<ul style="list-style-type: none"> + Built-in ActionScript enables dynamic processing without page reloads (server hits) + Natively supports vector graphics, streaming audio, and streaming video - Client-side XML/DOM parsing has latency issues 	<ul style="list-style-type: none"> + Flash Pro comes with many pre-built components: data grid, tree view, accordion, menu bar, etc. + Many additional Flash UI toolkits and components available by 3rd parties + Both flexible and grid layouts are possible (using container objects) • Flash's standard components are not easy to customize 	<ul style="list-style-type: none"> + Reduction in server loads due to processing on the client side and pure data requests (without presentation layer) after the initial load. + Socket connections allow server-initiated communications to client app + Flash 6+ can be used as the presentation layer with common server technologies such as Java Servlets, JSP, PHP, ASP, etc. 	<ul style="list-style-type: none"> • Applications can be ported to Flash Lite for use on mobile devices (w/Flash 4 functionality) • Can enable back button by accessing history object • Bad rep from designers reinventing simple UI widgets (in part due to Flash's component set) • In theory code is protected, but in reality it can be cracked with simple tools. • Can be connected to things like Flash Communication Server to enable real time AV Chat, IM, etc. between Flash Clients 	<ul style="list-style-type: none"> • Given proprietary format, APIs may change from version to version. However, Macromedia seems to be committed to standards such as CSS, XML, SOAP, SVG, etc. 	<ul style="list-style-type: none"> + Decrease costs by developing/testing once for all platforms + ActionScript 2 is object-oriented and based on ECMA Script (like JavaScript), so developers familiar w/JavaScript can quickly pick up ActionScript - However, programmers with relevant Flash experience are relatively scarce - Development could take longer because everything must be created from scratch: concept of pages, links, browse history, scale, etc.

Read More:

- Flash Installation Stats: http://www.macromedia.com/software/player_census/flashplayer/version_penetration.html
- Accessibility: http://www.macromedia.com/macromedia/accessibility/features/flex/best_practices.html
- Java and Flash socket connections: <http://www.dagblastit.com/java/sockets.html>

Examples:

- Rich Internet Applications (<http://www.klynch.com/archives/000074.html>)

Flash with Flex or Laszlo Presentation Server

Overview:
See Flash Overview...Flex and Laszlo offer presentation servers that can be installed on the server side. Each has a rich library of user interface components, an XML-based markup language used to declaratively lay out these components, and an object-oriented programming language which handles user interactions with the application [Coenraets 6].

+ Benefit - Drawback • Neutral

User Experience	Deployment & Reach	Processing	Components & Customization	Back-end Integration	Unique Features	Future Proofing	Staffing & Cost
		<ul style="list-style-type: none"> + Further reduction in server loads due to processing on the client side and pure data requests (without presentation layer) after the initial load. - Laszlo presentation server has performance issues 	<ul style="list-style-type: none"> + Flex's MXML toolkit provides components with standard presentation + Flex's MXML code can be styled with CSS or Flash + Laszlo's LZX toolkit provides components with standard presentation • Laszlo only compiles to Flash 5 standard (Laszlo has Flash 6 in beta) 	<ul style="list-style-type: none"> + Presentation tier (ASP, JSP, Servlets, ColdFusion, Flex or Laszlo) can integrate with common back-ends (EJB, Java Beans and Web Services , etc.) 		<ul style="list-style-type: none"> - Macromedia has abandoned server products in the past including: Drumbeat, SiteSpring, LikeMinds, Kawa, etc. (Most of these features are rolled into other products such as Flex or DreamWeaver) 	<ul style="list-style-type: none"> + Laszlo and Flex presentation servers can speed up development + Laszlo presentation server is open source + Flex Builder IDE makes it relatively easy to build applications with Flex - Development in Flex/Laszlo more familiar to traditional developers (relative to Flash programming) - Flex presentation server is expensive - Laszlo does not have a comparable IDE to Flex Builder

Read More:

- Flex Overview: http://www.macromedia.com/software/flex/productinfo/brz_overview/
- Laszlo Overview: <http://www.laszlo.com/products/>
- Comparison of Flex vs. Laszlo: http://www.infoworld.com/article/04/12/03/49TClasflex_1.html
- IDE for Laszlo: <http://alphaworks.ibm.com/tech/ide4laszlo>

Examples:

- Laszlo: Behr Color Smart: <http://www.behr.com/behrx/index.jsp>
- Flex: http://www.macromedia.com/devnet/flex/example_apps.html

Overview:

A Java Applet is a small program written in the Java programming language that can be embedded in an HTML page. Java applets can run in a web browser using a Java virtual machine (JVM)

+ Benefit - Drawback • Neutral

User Experience	Deployment & Reach	Processing	Components & Customization	Back-end Integration	Unique Features	Future Proofing	Staffing & Cost
<ul style="list-style-type: none"> + Java Applets offer a rich set of features for engaging interactions including drag and drop, animation, and other UI elements found in traditional desktop apps. + Enables real-time updates and validations as user moves completes forms - Latency due to relevantly large initial download and plug-in launch - Linking to, saving, and book-marking content can be tricky 	<ul style="list-style-type: none"> + Java applets require the Java plug-in or Java Virtual Machine (JVM) already installed in most browsers. - Java Plug-in supports IE and Netscape 3.0+ on Windows 95, 98, NT 4.0, and Solaris 2.5 and 2.6. JVM download is 5MB - Inconsistent browser implementations. (However, Sun's Java plug-in allows applets to run in Sun's Java Runtime Environment to ensure consistency) - Java Applet content is not indexed by search engines. 	<ul style="list-style-type: none"> + Applets are embedded in web pages and go beyond http for remote data request/response + Plug-in needs to be installed only once and then it caches all applets. + Very good processor-intensive visualization rendering (interactive and dynamically generated graphs, charts, etc.) + Can offload server-side processing to the client to save both CPU overhead and network bandwidth 	<ul style="list-style-type: none"> + Swing and JavaBean controls available - Swing can be slow and cumbersome in a web-based client UI 	<ul style="list-style-type: none"> + HTTP, HTTPS, FTP, and Gopher are supported in the 1.2.2 release of Java Plug-in Software. + Full Java Compatible support; offer all the benefits of the Java language, including portability, faster development, powerful API libraries, etc. [1] + Can create a socket connection to servers and stream data between the applet and the server 	<ul style="list-style-type: none"> + Applets can write to local disk if they are digitally signed (via a dialog window) + Applets run in a "sandbox" so few security concerns exist; strict rules on how an applet can interact with your computer and the network • Network access only to the launching server; only reliable source of data and storage for an applet is on its host server. Applets are not permitted to invoke any local executable program on the client's machine. 	<ul style="list-style-type: none"> • Java is more commonly used as a platform than it is for applets • The use of applets does not seem to be matching the growth of the Web - Microsoft decided to stop shipping a JVM with it's newest versions of Internet Explorer 	<ul style="list-style-type: none"> + Many developers who know Java relatively available

Examples:

- <http://java.sun.com/applets/>
- Catalog View is a useful database applet that can display your data in a grid display. <http://javaboutique.internet.com/CatalogView/>

Read More:

- <http://www.matisse.net/files/glossary.html>
- Sun hasn't set yet on Java applets, Rick Moore <http://www.adtmag.com/article.asp?id=7074>

Overview:
Active X is an application program interface (API) developed by Microsoft that extends and enhances the functionality of web sites viewed in Internet Explorer. When accessed through Internet Explorer, Active X controls can have full access to the Windows operating system.

+ Benefit - Drawback • Neutral

User Experience	Deployment & Reach	Processing	Components & Customization	Back-end Integration	Unique Features	Future Proofing	Staffing & Cost
<ul style="list-style-type: none"> + Adds interactive features and animation found in traditional desktop applications + Integration with local resources (files, hardware, office applications, etc.) possible 	<ul style="list-style-type: none"> + Works on most Windows machines + Active X controls can be installed on the user's computer and run locally during subsequent sessions. - Not cross-platform (Microsoft Windows only), not recommended when user platforms can vary 	<ul style="list-style-type: none"> + Reduced server loads due to processing on the client side and pure data requests (without presentation layer) after the initial load. + Ability to create socket connections and server initiated communications 	<ul style="list-style-type: none"> + There are many ActiveX controls available with functionality ranging from simple to full-featured spreadsheets and word processors. 	<ul style="list-style-type: none"> + Active X can integrate directly with SQL servers and many other server technologies. 	<ul style="list-style-type: none"> + Active X controls can be reused in other Microsoft programming and database languages. - Access to local resources can create security issues (Mandatory registration system for Active X controls helps deal with this issue) 	<ul style="list-style-type: none"> • Microsoft will determine the future of Active X (hasn't received much development attention from Microsoft for number of years) - Increased use of non-windows platforms (ie: Max OS X, mobile technologies, etc) 	<ul style="list-style-type: none"> • Can be developed in Windows IDE tools.

Examples:

- <http://activex.microsoft.com>

Read More:

- <http://msdn.microsoft.com/library/default.asp?url=/workshop/components/activex/intro.asp>
- <http://msdn.microsoft.com/workshop/components/activex/intro.asp>
- Security issues: <http://office.microsoft.com/en-us/assistance/HA011403101033.aspx>

Overview:

Java Web Start is an application that enables Java client applications to be launched, deployed, and updated from a standard Web server.

+ Benefit - Drawback • Neutral

User Experience	Deployment & Reach	Processing	Components & Customization	Back-end Integration	Unique Features	Future Proofing	Staffing & Cost
<ul style="list-style-type: none"> + Applications are cached locally once downloaded + Supports complete desktop interactions • Applications are launched independent of Web browser - Variable application access may be counter-intuitive to users 	<ul style="list-style-type: none"> + Java Web Start version 1.0 or 1.0.1 must be installed on the client machine (10 MB file); then applications can be deployed via URLs + Locally cached version of application is automatically updated when run and remote version has been updated • Available on Windows, Solaris, OS X (pre-installed), and Linux - Requires manual installation of Java Web Start on each client machine (more tedious than browser's automatic installation of Java plug-in) 	<ul style="list-style-type: none"> • Same as any Java desktop application 	<ul style="list-style-type: none"> • Same as any Java desktop application 	<ul style="list-style-type: none"> • Applications must be packaged in accordance to the Java Web Start guidelines • Can only be used for client Java applications written for the Java 2 platform • Requires custom code on server; requires special configuring of the JNLP MIME type 	<ul style="list-style-type: none"> + Applications work offline + Java Web Start allows client-side Java applications access to a variety of local-machine resources, such as file access, and clipboard access (require dialog box acceptance) • Not well-suited for deploying applications to mobile devices 		<ul style="list-style-type: none"> + Free, Redistributable in 12 languages • Applications can be written normally as stand-alone applications for the Java 2 platform

Examples:

- <http://www.up2go.net> (More Than 70 Apps Listed; Apps Are Rated)
- <http://www.connectandwork.com/external>

Read More:

- Java Web Start to the rescue, Raghavan N. Srinivas <http://www.javaworld.com/javaworld/jw-07-2001/jw-0706-webstart.html>
- Unofficial Java Web Start/JNLP FAQ, Gerald Bauer: <http://webstartfaq.com/>
- <http://www-106.ibm.com/developerworks/java/library/j-webstart/>

Overview:

A smart client is a windows application that bridges the gap between thin clients and rich clients. Smart clients are web-maintainable and deployable (can load components it needs to run from a remote location), yet offer rich interactions through compiled, local runtimes

+ Benefit - Drawback • Neutral

User Experience	Deployment & Reach	Processing	Components & Customization	Back-end Integration	Unique Features	Future Proofing	Staffing & Cost
<ul style="list-style-type: none"> + Full desktop interactions are possible (direct manipulation, hotkeys, etc.) + Application components can be downloaded dynamically as needed allowing applications to load quickly + Integration with local resources (files, hardware, office applications, etc.) possible with .NET framework - Variable application access may be counter-intuitive to users 	<ul style="list-style-type: none"> + Works on most Windows machines (Requires Microsoft Windows .NET framework) + Smart clients can be deployed by CD, DVD, an application deployment infrastructure, or a web server using HTTP - Not cross-platform, not recommended when user platforms can vary 	<ul style="list-style-type: none"> + Server and network load is reduced because components are only updated if needed. + Peer to peer connectivity is possible through ability to listen for incoming network requests - Initial download can be larger than comparable thin client solutions 	<ul style="list-style-type: none"> + Windows form libraries available - Currently only supports grid layout (flexible layouts will be supported in Longhorn smart clients) 	<ul style="list-style-type: none"> + Can use SOAP, XML, HTTP, etc. for data 	<ul style="list-style-type: none"> + Applications can run offline • .Net framework provides security through code access security (CAS) infrastructure • Multiple versions of a smart client application can target specific devices 	<ul style="list-style-type: none"> • Microsoft has a development roadmap to continue enhancing smart clients over the next several years. 	<ul style="list-style-type: none"> + Developers can use Microsoft Visual Studio .NET as a drag and drop IDE • Reduce development and maintenance time due to applications not subject to DLL conflicts

Read More:

- <http://msdn.microsoft.com/netframework/programming/winforms/smartclient.aspx>
- http://msdn.microsoft.com/architecture/journal/default.aspx?pull=/library/en-us/dnmaj/html/choosing_presentation_layer.asp
- www.code-magazine.com/Article.aspx?quickid=050023

Examples:

- <http://www.windowsforms.net/Default.aspx?tabindex=6&tabid=48>

Canoo Ultra Light Client (ULC)

Build Rich Internet Applications with Java standards. ULC's pure java library offers a server-side API to Swing (servlet or bean), providing rich GUIs for J2EE applications. Java-based thin-client presentation engine needs to be installed once for any number of applications; may be deployed by Java Web Start, as an applet, or by any other deployment mechanism

- <http://www.canoo.com/ulc>

Java WebCream

Acts as a Java AWT or Swing to HTML bridge. It is a dynamic Java to HTML converter. WebCream turns thick client GUI applications and applets into DHTML websites.

Macromedia Central

Unlike conventional web sites, Central applications can function both online and offline, keeping important information constantly available. Central applications run in a controlled environment to keep sensitive data secure. And because Central automatically notifies the user when new versions are available, applications are always up-to-date.

- <http://www.macromedia.com/software/central>

Macromedia Director

A well-established way to create and deploy applications for CDs, DVDs, kiosks, and the Internet

- <http://www.macromedia.com/software/director>

Microsoft's Avalon & XAML

Avalon is the platform for the next-generation of Microsoft Windows client applications. It will create a new presentation platform by combining the features of the DHTML, a vector-based display engine, and Win32 development platforms. XAML, Avalon's markup language used to declaratively represent the user interface, enables the separation of UI design from the underlying code.

- <http://msdn.microsoft.com/Longhorn/understanding/pillars/avalon/avnov04ctp/default.aspx>

Nexaweb

Nexaweb provides a software platform for developing and deploying Enterprise Internet Applications. The Nexaweb platform includes the following: 1) J2EE server component, 2) 150k java-based client app, 3) real-time messaging layer, 4) WYSIWYG development environment.

- <http://www.nexaweb.com/products.asp>

PDF & XDP

Adobe is trying to evolve PDF into an allpurpose fill-in-the-forms container for Enterprise workflows. The new Adobe tool will have an IDE for creating forms in XDP files that are a superset of PDF and will use JavaScript as the programming API [6].

- <http://www.adobe.com/products/acrobat/readstep2.html>

SMIL

The Synchronized Multimedia Integration Language (SMIL, pronounced "smile") HTML-like authoring language for interactive audiovisual presentations.

- <http://www.w3.org/AudioVideo/>

SVG

Language for describing two-dimensional graphics in XML. Graphics can be vector, bitmap or text. Promises the following features: rich manipulation of graphic objects, leverage of W3C specifications and standards efforts, ability to use light-weight vector graphics for fast downloads, high resolution, high quality printing.

- <http://www.adobe.com/svg/demos/main.html>
- <http://www.w3.org/TR/SVG/intro.html>
- <http://www.w3.org/Graphics/SVG/SVG-Implementations>

Xforms

Traditional HTML Web forms don't separate the *purpose* from the *presentation* of a form. XForms, in contrast, are comprised of separate sections that describe what the form does, and how the form looks. This allows for flexible presentation options, including classic XHTML forms, to be attached to an XML form definition. XForms 1.0 is a W3C recommendation.

- <http://www.w3.org/MarkUp/Forms/>

XSL & XSLT

The eXtensible Stylesheet Language (XSL) is a vocabulary describing how to format eXtensible Markup Language(XML) for different media.

XSL Transformations (XSLT) can transform data using one XML vocabulary into another. XSLT processors can be installed on the server or executed within certain web browsers. XSLT have performance issues in high-volume applications.

- <http://www-106.ibm.com/developerworks/xml/library/x-xslt/?article=xr>
- <http://www.xml.com/pub/a/2003/11/26/learnXSLT.html>

XUI

XUI is a Java and XML framework for building rich client, desktop and mobile applications (including PDAs).

- <http://xui.sourceforge.net/>

XUL

XML User Interface Language developed by Mozilla. Pronounced "zool". It is a markup language used to create dynamic user interfaces. Provides access to programming interfaces such as read/write over the network or communicating with web services. Compatible with many standards such as HTML, XHTML, XSLT, CSS2, DOM2. Will not work with Microsoft Internet Explorer.

- <http://www.xulplanet.com/tutorials/xultu/>
- <http://xulplanet.com/tutorials/whyxul.html>

Luke Wroblewski is an interface designer, strategist, author, and Principal at LukeW Interface Designs, a consulting firm he founded in 1996. Luke has authored a book on Web interface design principles titled *Site-Seeing: A Visual Approach to Web Usability* (Wiley & Sons, 2002) and multiple articles on interface design including those featured in his own Web-zine, *Functioning Form*.

Previously, Luke taught Interface Design courses for the Graduate School of Library and Information Science at the University of Illinois at Urbana-Champaign and worked as a Senior Interface Designer at the National Center for Supercomputing Applications (NCSA), birthplace of the first readily available graphical Web browser, NCSA Mosaic. At NCSA, Luke designed interface solutions for Hewlett-Packard, IBM, and Kellogg's, and co-developed the Open Portal Interface Environment (OPIE). Luke's research on Web-based interface designs has been published and presented at national and international conferences.

Frank Ramirez, Principal at Ramirez Design LLC, is a user experience strategist, researcher, and design consultant. Frank has led the design of many digital products and services, including e-commerce applications, touch-screen kiosks, rich internet applications, intranets and training programs. Clients include Yahoo!, eBay and numerous start-ups.

Prior to starting his consultancy, Frank was Sr. Information Architect at Walmart.com where he led requirements definition and information architecture for multiple in-store, online and cross-channel initiatives. He participated in a wide variety of project activities including: discovery research, concept generation, usability testing, cross-functional collaboration, IA documentation and communication.

Frank earned his B.S. in Graphic Design and has designed enterprise-level web applications since 1996.

Reviewers

Much thanks to the following individuals who contributed to the ideas and accuracy of this paper:

Benjamin Wigton, Lead Interface Engineer, Avenue A / Razorfish
David Heller, Principal Designer, IntraLinks & Steering Committee Member, Interaction Design Group (IxDG)
Dirk Knemeyer, Principal, Involution Studios
Erin Malone, Director, Platform Design, Yahoo! Inc.
James Reffell, Sr. UI Designer, eBay Inc.
Jamie Hoover, Sr. UI Designer, eBay Inc.
Johnathan Boutelle, Principal, Uzanto Consulting
Michael Hoch, Director of Prototype Design, Yahoo! Inc.
Peter Bogaards, Founder, BogieLand Information Design & Information Architecture
Todd Warfel, Partner, Design & Usability Specialist, MessageFirst
Uday Gajendar, User Experience Pathfinder, Adobe Systems Inc.
Ziya Oz, Consultant, Application Strategy/Architecture/Design

*Note: The opinions of these reviewers may not reflect those of their employers.

- [1] Bob Baxley, What is a Web Application?. Boxes And Arrows, 2003
http://www.boxesandarrows.com/archives/what_is_a_web_application.php
- [2] Croczilla.com, Core JavaScript Guide 1.5.
http://www.croczilla.com/~alex/reference/javascript_guide/intro.html
- [3] Brent Ashley, Ashley IT.
<http://www.ashleyit.com/rs/main.htm>
- [4] Jesse James Garret, Ajax: A New Approach to Web Applications.
<http://www.adaptivepath.com/publications/essays/archives/000385.php>
- [5] Christophe Coenraets, An Overview of MXML, the Macromedia Flex Markup Language.
<http://www.macromedia.com/devnet/flex/articles/paradigm.html>
- [6] The Open Sourcing.
<http://www.theopensourcery.com/osrevFlex.htm>
- Jonathan Boutelle. Flash RIAs vs. Javascript RIAs
http://www.jonathanboutelle.com/mt/archives/2005/03/flash_riAs_vs_j.html
- Adam Rifkin
<http://ifindkarma.typepad.com/relax/2004/12/weblications.html>
- Kevin Lynch. Rich Internet Applications: It's Happening!, Macromedia
<http://www.klynch.com/archives/000074.html>
- Alex Kalinovsky. Savor success with Java on the front end. HTML, Swing, or XML: Choose the best front-end technology for your Java development
<http://www.javaworld.com/javaworld/jw-04-2001/jw-0420-swing.html>
- Sun hasn't set yet on Java applets, Rick Moore
<http://www.adtmag.com/article.asp?id=7074>
- Chapter 4: Java Applets. Internet Computing
<http://www.cs.swan.ac.uk/~csneal/InternetComputing/index.html>
- David Hill, What is a Smart Client anyway?
<http://blogs.msdn.com/dphil/articles/66300.aspx>
- David Hill, Choosing the Right Presentation Layer Architecture. Microsoft Corporation
http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnmaj/html/choosing_presentation_layer.asp
- Microsoft Corporation. Smart Client Application Model and the .NET Framework 1.1
<http://msdn.microsoft.com/netframework/programming/winforms/smartclient.aspx>
- Markus Egger. The Revenge of the Thick Client
www.code-magazine.com/Article.aspx?quickid=050023
- James R. Borck. RIA platforms lend apps more Flash: Macromedia Flex and Laszlo Presentation Server bring new tricks to Internet application delivery
http://www.infoworld.com/article/04/12/03/49TClasflex_1.html
- Marc A. Garrett. Review: Macromedia Flex and Flex Builder
http://digital-web.com/articles/macromedia_flex_and_flex_builder/
- Drew Falkman. Review: Macromedia Flex, Java Boutique
http://javaboutique.internet.com/reviews/macro_flex/
- Open Sourcing. Overview - Macromedia Flex
<http://www.theopensourcery.com/osrevFlex.htm>
- Mark Eagle. Integrating Macromedia Flex with Java, O'Reilly
<http://www.onjava.com/pub/a/onjava/2004/12/01/flexjava.html>
- Drew McLellan. Very Dynamic Web Interfaces
<http://www.xml.com/pub/a/2005/02/09/xml-http-request.html>
- Peter-Paul Koch. Ajax, promise or hype? O'Reilly
http://www.quirksmode.org/blog/archives/2005/03/ajax_promise_or.html
- Drew Falkman. Review: Canoo ULC 5.1, Java Boutique
http://javaboutique.internet.com/reviews/canoo_ulc/
- Overview. Web-enabling Java by Java to HTML converter
<http://www.creamtec.com/webcream/overview.html>
- Coach K. Wei . Why Web Applications Can be Problematic and Unreliable
<http://www.sys-con.com/story/?storyid=47364&de=1>