

A 100% web-based architecture is a dramatic shift in enterprise computing with tremendous benefits for you -- the business user -- and the IT staff that must implement and maintain e-business applications. The challenge is to find a vendor that can truly deliver this innovative architecture.

True web-based products offer tremendous business benefits, including:

- **Saving time and money through centralized application implementation and maintenance.** With a web-based architecture, the application is implemented and maintained on a centralized server and not on end-users' PCs or mobile devices. The "client" is simply an HTML user interface accessed through a standard web-browser like Internet Explorer. Because no software is installed on the client, the IT staff can implement and maintain the product in one central application server, rather than running around to hundreds or thousands of users' PCs or mobile devices to install software or make modifications. Once the application has been installed and customized on a central server, rolling it out to users is as easy as emailing a URL to them. The business benefits of centralized implementation and maintenance include substantial savings of IT staff time and expense.
- **Greatly improving interoperability with other applications.** Another benefit of a web-based architecture is that integration with other applications is greatly improved. Because web-based applications are deployed only on a central server, all other applications can integrate directly at the application server. This is substantially easier than integrating to the hundreds or thousands of end-user clients, in addition to an application server. Because most businesses are challenged to integrate numerous disparate applications, the centralized, web-based, model of application integration is a dramatic advantage. The business benefits of this integration architecture include substantial savings of IT staff time and expense as well as increased application value through more efficient integration.
- **Extending access to the application beyond the enterprise.** Because the interface to a web-based application is simply a web page accessed through a browser, access to the application can be extended to anyone with Internet access. These external users might be employees on the road or external constituents like partners or suppliers. A great example is Partner Relationship Management (PRM) where the end-users may include thousands of partners outside the organization. Imagine the difficulty of installing software on the PCs of every employee at every partner in a PRM implementation! The business benefits of extended access include the improved communication, collaboration and efficiencies that enable such revolutionary new business applications as customer self-service, PRM and business-to-business net marketplaces.
- **Making enterprise applications easier to use.** The rich experience of interacting with a web page makes web-based applications substantially easier to use. We are all aware of the difficulties users experienced learning to use complex client/server Enterprise Resource Planning (ERP) and Sales Force Automation (SFA) systems. This is because the user interface consisted of multiple client/server screens with features hidden behind numerous drop-down menus and multi-tab dialog boxes. With a rich HTML interface, the application can include graphically rich icons with descriptive text, guided navigation and the familiar "Back" and "Forward" arrows. The business

benefits of an easy-to-use application include rapid end-user adoption and the elimination of unused "shelfware."

- **Investing less money on client side hardware.** The "thin client" user interface in a web-based architecture requires substantially less processing power on the client hardware. Because there is no application processing executed on the client machine, there is no requirement to continuously upgrade hardware with more processing power and memory. The business benefits here include substantial cost savings and less IT resources focused on complex upgrade cycles.

These are just some of the business benefits of a web-based architecture. Web-based architecture is revolutionizing the business applications market. The challenge is to identify which vendors offer a truly web-based architecture and which are simply putting "lipstick on a pig" through a web-enabled architecture.

The definition of a web-enabled architecture. A web-enabled architecture is typically offered by vendors that began developing their architecture during the client/server heydays of the early 1990s and now face competitive pressure to offer a web architecture. Their answer is to jury-rig a browser-based user interface to their client/server application -- not rewrite their application in a web-based architecture.

- **A typical implementation of this web-enabled variant is to "recreate" the heavy client within the user's browser using Java Applets or Microsoft ActiveX Components.** These large pieces of software code **take considerable time to download, must be maintained by IT staff** and, because **processing is executed on the end-user client, complicate integration with other applications.** They also make **less attractive the option to extend access to users beyond the enterprise by requiring significant network bandwidth** not only to download the components, but also to transfer the data required for client-side processing. Also, the "recreation" of the client/server application usually **results in a level of complexity for the end-user similar to a client/server interface.** Finally, the client-side processing executed within these applets or components **requires significant client hardware processing power.** Basically, all five of the main benefits of a web-based architecture have been eliminated in this "fat client" web-enabled architecture.
- **Another variant of web-enabled architecture is a thin client, HTML interface to the client/server application that is limited in its ability to provide full functionality.** In this variant, the vendor typically offers customers a thin client to meet a requirement for web access, but persuades the customer to implement a client/server version of the software to get "full functionality." In reality, this is more of a bait-and-switch than an architecture, but it is not uncommon. The reason that this web-enabled architecture cannot offer full functionality is that it is **only accessing the application server or database server of the client/server architecture. It does not access the client-side processing capability** that provides most of the end-user functionality in a client/server application. It also performs less efficiently because the client/server application infrastructure is **not designed to rapidly generate the HTML pages that serve as the user interface.** As a result, this web-enabled

architecture simply does not offer the features, functions and performance that the end-user requires and is therefore not a viable alternative to a web-based architecture.

Web-based applications should have the following characteristics:

- **An application server that manages the execution of ALL business logic and workflows and was developed from the ground up to do so;**
- **A thin-client user interface that is accessed through a web-browser and downloads only lightweight HTML and limited amounts of JavaScript -- only in very unique situations should Java Applets or ActiveX controls be downloaded to the browser;**
- **A full feature set, rather than a limited feature set that is a subset of capabilities available in an "optional" client/server version.**
- **A high performance interface that provides a rich interaction with the user without waiting for screen refreshes.**