

# PureWeb®: Architectural and Competitive Review

---

## *Technical and Architectural Overview*

### **Introduction**

CSI's PureWeb® technology facilitates the re-purposing of legacy applications to be used on the Internet, or the rapid and advanced development of new fully web-enabled, rich GUI applications that manipulate large data sets and/or provide interactive imaging capabilities. PureWeb provides a migration path for existing legacy applications to be used on the Internet, and for advanced mobile computing on a variety of hand held and wireless devices. The technical investment in PureWeb is preserved over the long term, as the client and application tiers are isolated from each other. This means that the legacy application can be maintained, upgraded, and modernized over a much longer cycle than the client tier implementations, which can be quickly implemented for new client platforms and devices that arrive in the market.

There are three tiers to the PureWeb solution:

### **The Client Tier:**

The Client Tier of the PureWeb architecture enables a reproduction of the original user interface (UI) on a remote client device, regardless of its richness or complexity. This is the level of the PureWeb application platform where the UI of the legacy application is reproduced or “re-factored” to use a ubiquitous Rich Internet Application (RIA) platform, such as Microsoft's Silverlight, Adobe's Flash, Oracle's Java or Apple's iPhone. Client Tier UIs are built using modern tools and techniques facilitated by the PureWeb Client Software Development Kit (SDK). The cost of reproducing or re-factoring the UI (which now runs remotely on the client device) is low because only the user-facing aspects of the interface need to be reproduced. The modern development environment allows rapid assignment of the original GUI features to identical widgets within the RIA toolboxes. Unique icons or UI art is readily preserved and transferred to the client UI to ensure the conservation of brand look and feel. The PureWeb Client SDK facilitates the re-use of the application logic in the existing application running on a server to drive the full status of the UI running on a client. This unique tier allows for the full re-factoring of advanced, rich GUI applications so that the same business logic can be used remotely, with full feature sets and highly functional workflow, while allowing adjustments to be made to layout to accommodate even the small screen spaces of hand held devices. In addition, where new tools and interface capabilities have been introduced since the development of the original application (such as the advanced touch screen command and control capability of the new generation of hand held devices) these UI controls can be introduced as appropriate.

### **The Server tier:**

The PureWeb Server software forms the middle tier of the architecture and connects the Client Tier to the Application Tier. The Server Tier, comprising PureWeb software combined with off-the-shelf web server technology, provides connection marshalling and application process management across a

cluster of physical and/or virtual servers. The PureWeb Server software is responsible for fielding connections from remote clients, finding a server with an available slot to run the legacy application, starting the legacy application, and then brokering the ongoing communication session between rich client and legacy application. When the session is over, the PureWeb Server software ensures that the legacy application slot is made available for another session.

## **The Application Tier:**

The Application Tier hosts the legacy application on standard hardware appropriate for the legacy application's requirements. This same hardware supports the Server Tier, which adds a very small computational burden to the total workload. The legacy application's source code is modified slightly to utilize the PureWeb Server SDK: a lightweight and efficient set of software libraries that handle all communications and messaging between the legacy application and the PureWeb Server Tier. The PureWeb Server SDK provides virtualization of the image-intensive parts of the legacy application by marshalling mouse and keyboard events from the client to the legacy application, and by efficiently compressing and transmitting resulting images from the legacy application to the client. Because the PureWeb Server SDK is linked directly into the existing application, it is possible to adjust the optimization of image compression, quality, and transmission for different use cases, therefore the applications can be highly tuned for optimum low bandwidth performance, including very small bandwidth, high latency wireless 3/4G networks. The PureWeb Server SDK keeps the client UI synchronized with the full status of the legacy application so that the well-tested existing application logic is used (without modification) to drive the status of the rich client UI.

PureWeb provides a rapid, permanent migration path for existing legacy applications, or those under construction, to be used, fully featured and fully functional, at high performance over the Internet, on the office network, at home, or on wireless networks, including 3/4G, from an array of wireless devices for truly mobile, secure, remote high performance computing. The investment in PureWeb is preserved over the long term as the various client and application tiers are isolated from each other. This means that the legacy application can be maintained, upgraded, and modernized over a much longer cycle than the Client Tier UI implementations, which can be quickly implemented or refreshed for new client platforms and devices that arrive in the market.

## *Alternative/Competitive Technology Architectures*

Until PureWeb, which is very unique, there have been three primary ways of re-purposing existing software applications to be used on the Internet:

## **Desktop Virtualization Technology:**

There are a number of desktop virtualization technologies available in the market: Microsoft Application Virtualization (App-V); VMware ThinApp; Citrix XenApp; HP remote Graphics Software (RGS); and Symantec Software Virtualization Solution (SVS). These technical approaches allow an existing software application to be accessed from a remote client. The tools work by "wrapping" an application on a central server, capturing its output and delivering this output 1:1 to a client which posts it on the client display. These technologies are optimized for business productivity software virtualization, where file sizes are smaller, and often document based, such as Microsoft Office. However, there is usually a heavy O/S specific proprietary client software agent to initially download, install, configure and maintain on the end-users' devices, and the remote performance of the interactive imaging or graphics components of such applications are often quite poor, particularly for applications of a "heavy lifting"

nature, where large datasets are in use, or those with rich and highly interactive UI's or heavy use of computer resources. The methods by which these technologies detect that the application is ready to have a frame delivered often rely on kernel level interruptions to the application which can introduce delay and costly computational overhead (as much as 200%) to each copy of the virtualized software. Furthermore, tying so closely to the operating system restricts the manner in which the virtualized application may be deployed and the environment in which it must run. It may be difficult to migrate the virtualized application between operating system versions or to deploy it in locked-down secure environments. To enable better performance, predictive algorithms may be used in combination with local rendering on a client application. For instance, a guess as to the rendered results of new text entered into a word processor may be displayed while the text is returned to the server and the actual rendered screen sent back to the client. While this may be appropriate for business applications focused on documents of various types, there may be no place for such enhancements in other fields. In any case, these acceleration features are specific to well known applications for mass markets and will not be available for high performance or in-house legacy applications. The remote applications are most often also one screen to one screen, disallowing the use of "multi-monitor" advanced software applications, in which various aspects of the application are displayed and used simultaneously across multiple monitors. The remoted applications cannot be centrally hosted and shared by multiple users without the need for stacking up copies of the applications on virtual machine type software. Finally, it is not possible to re-factor the application to look like or behave the way users might expect either in modern UI tools or in a form factor adapted to the client device. For example, if a Windows application is remoted using Citrix to an iPhone client, the application will still look and feel like a Windows application designed for a much larger screen. This renders any remoted advanced software applications functionally useless at the hand held device level, and thus, does not enable the mobile computing marketplace.

### **Service Oriented Architecture (SOA) technologies:**

This technical approach has been popularized by companies like BEA, Web Methods, HP, IBM, and others. This usually involves throwing away huge parts of an existing legacy application as the application is rewritten from scratch, often using completely new coding languages and technologies that aren't interoperable or compatible with those used to originally build the legacy application. In most cases, legacy applications must be re-written in .net or java prior to being implemented on an SOA platform. This is usually a risky, time consuming, and expensive endeavor, requiring massive new investment in development, testing and quality assurance (QA) to ensure that application and business logic is correctly re-implemented on the new SOA platform. In addition, the final UI resulting from this implementation is not the rich GUI interface of an advanced software application, but may appear much more like a web page using only simple browser widgets and plain text. Overall, these methods are very strong in displaying and using information from database type applications or store fronts, but are very poor when graphics intensive applications are deployed, or rich GUIs are required. Lastly, this implementation does very little to address the need for fully mobile computing across a variety of hand held platforms.

### **Traditional Hardware Virtualization:**

In addition to the virtualization of the desktop and associated applications the entire workstation may be virtualized by products such as VMWare vSphere and Microsoft Hyper-V. These solutions, rather than "wrapping" the applications, create entire virtual machines (VM) from the operating system upwards. Supposedly, applications running in these virtual machines are not "aware" that they do not

own the entire physical system. Within VM environments, it is still necessary to transfer the expected display state to a remote client desktop. Thereby the VM approach can suffer the same issues as desktop virtualizations – additional overhead, poor relationship between what is displayed and what is delivered, additional licensing, management and cost overhead, etc. With particular reference to image intensive applications the VM does not do an acceptable job of virtualizing the graphics processors (GPU). In fact, many VM approaches intercept applications use of the GPU and perform the operations in software at a grievous cost to performance. These issues make hardware virtualization largely unacceptable for image and graphics intensive applications, and do not solve the market requirements for the mobilization of advance classes of software applications.

## Conclusion

PureWeb® is unique, deep IP, with a number of strong, targeted patents currently pending, with others triangulating around the same. PureWeb® can be thought of as a “new generation” hybrid between SOA and desktop virtualization, which takes advantage of the scalability of the SOA approach on the client side, while preserving the investment in development and testing of the legacy application on the server side, resulting in a permanent “transformation” of the target software application into a high performance web and mobility-enabled technology. PureWeb® allows the UI of the legacy application to be built on the PureWeb® client either as a faithful recreation of the original, or re-factored with a new look and feel, as well as being re-factored for high performance use on wireless and hand held devices. In either case, the flexibility of PureWeb® enables the possibility to virtualize only certain parts of the legacy application. Likewise, it is possible to develop clients that expose only certain parts of the virtualized legacy application. Multiple clients can be implemented on multiple platforms in ways that are most appropriate for their respective platforms, or that target different user groups and use cases. It is even possible to combine features from multiple legacy applications into a consistent web and mobile based UI, by simply harmonizing disparate applications (with different UIs, code bases, and architectures) into a consistent web and mobile based UI accessible from a single web browser log in, or actually re-combining a variety of usable aspects of the disparate applications into one new, high performance web and mobile application. PureWeb is the new state-of-the-art technology platform for the high performance virtualization and mobilization of advanced software applications.