



## The Curl™ Client/Web Platform

## Executive Summary

Today's leading corporations use the Web to deliver enterprise applications, reducing the costs of deployment and maintenance. Yet the inefficient server-centric infrastructure has limited the power of Web-based applications. Compared to desktop apps, Web applications provide poor performance and usability.

Driven by the business imperatives of increased productivity, reduced costs and shorter time-to-market, a new computing architecture is now emerging that will enhance usability and system efficiency. Curl™ Corporation solves the existing infrastructure problems by adding client-side technology. Combining the best of the old Client/Server model and the current Web-enabled systems, this new Client/Web architecture enables "rich client" applications that are deployed over the Web but run on the client device.

Curl has the only complete purpose-built Client/Web platform. It represents a technology revolution that can transform Web-enabled enterprise applications. It can be rapidly developed and deployed; it will seamlessly interface with any existing back-end system. Available today, it delivers business value now.

## A New Computing Architecture Emerges: Client/Web

Enterprise computing has been deeply affected by the Web revolution. Most major corporations have invested heavily to Web-enable their enterprise applications by adding a Web interface in front of their backend server infrastructure, using such products as IBM's WebSphere or BEA's WebLogic. All enterprise applications are then accessed through a single client—the Web browser. The major selling point for this architecture is centralized control and administration, which eases deployment and maintenance and lowers total cost of ownership (TCO). Moreover, the Web offers further promise ahead: network access to standardized software components—Web services.

Yet many enterprises find themselves bumping hard against limitations imposed by the current Web infrastructure. The Web as it now exists was built to distribute information, mostly static content. As a means of distributing applications, it's still underdeveloped.

### *Today's Web is Server-Centric*

The current Web is geared for servers to send down "pages" of minimally interactive content. In this architecture, almost all the computing power is applied on the server; the code executes on the server. The so-called "thin client" is really a "dumb client" and the computer on the user's desk becomes an expensive display device.

Ironically, as processor power has grown exponentially, the desktop machine has never been more capable. A business user expects a typical desktop application to offer instant responsiveness and the capacity to tailor the user interface and make quick changes to the data on display—dragging-and-dropping and cutting-and-pasting and re-sizing tables.

Users don't expect such performance from the Web. They have to wait while applications running on powerful servers process all output through a Web server. The end user display is a Web page with content pre-formatted on the server; any interactivity or customization of the page requires further round trips to the server. It's a grossly inefficient use of server power, network bandwidth and the capacity of the local machine. Too often, the outcome is wasted time and productivity: "executive wait."

And of course, all access to corporate applications stops when the network connection is down. To an executive struggling with a spotty 2400 baud Internet connection in a Latin American hotel the claim of access "from anywhere" will ring hollow.

### *The Need for Client-Side Web-Based Apps*

Somehow the Web has to transition from information delivery to application delivery. Forrester Research predicts the coming "X Internet," the first stage of

which it terms the “Executable Net,” so-called because applications—executable code—will be delivered over the network not to a thin display client but to a client rich enough to run them.

A consensus has emerged that a new software architecture is needed, one that complements rather than replaces today’s Web architecture. As long as applications run on servers, delivery over the Web will be marred by latency problems. One clear solution is to augment server-side apps with Web-based apps that run on client devices.

Enterprise applications that are delivered on the Web but execute on the desktop machine can achieve desktop app performance while remaining centrally administered. The result will be major operational efficiencies and a significant boost to user productivity. Such a shift brings the performance benefits of the Client/Server world to the controlled environment of today’s Web-enabled systems.

A hot new market has already developed as a range of companies offer various levels of “rich client” solutions. Analyst firm Gartner is predicting that new technologies in this field will make it hotter still, leading to “a resurgence in rich-client application deployment.”

In the evolution of enterprise computing from Client/Server through to today’s Web-enabled systems, the next level is Client/Web.  
(See Figure 1.)

Every major leap forward in business systems has required a “Technology Revolution.” The transition from the mainframe era to Client/Server systems delivered efficient computing to every desk, transforming productivity. The transition to Web-enabled systems drove down TCO. The imminent transition to Client/Web systems, which promises *both* higher productivity and control of TCO, likewise requires a new technological leap.

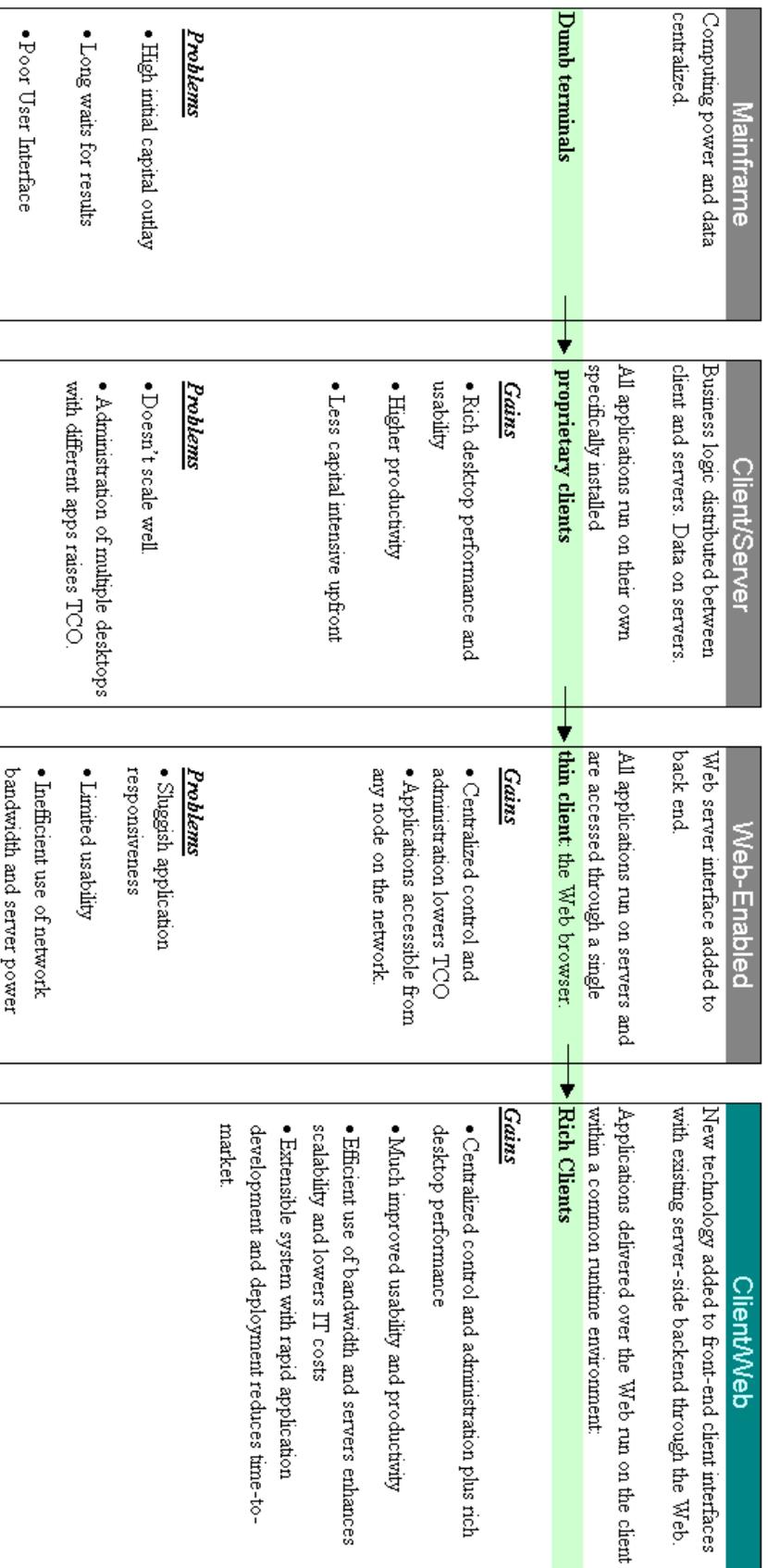
Curl Corporation is the only player in this emerging marketplace with a complete solution built specifically to deliver this new architecture. While others offer adaptations that improve upon the current server-centric technology, Curl has a comprehensive **Client/Web Platform**.

## The Curl Platform: New Technology

Much of the backend infrastructure we have today for Web-based enterprise apps is built upon Java’s J2EE. Microsoft’s .NET project, which is targeted as an alternative to the Java platform, will deploy initially on the server side also.

Curl provides new client-side technology that is entirely complementary to both these server-side alternatives. The unique Curl Client/Web Platform—a

## The Evolution of Enterprise Computing



## The Curl Client/Web Platform

Curl applications execute on the **Surge™ Runtime Environment**, a fast, stable client platform. It includes a just-in-time compiler executing on-the-fly; a multimedia engine enabling unprecedented integration of text, graphics, animation, audio and end-user input; an advanced GUI system for creation of interfaces either standardized or highly customized; an XML parser for direct interpretation of data from any database; and standard HTTP networking for connectivity with any Web-based backend. The runtime operates within a strict security model that ensures system administrators have sole control over security features.

The Curl Client/Web Platform is completed with the **Surge Lab™ Integrated Development Environment (IDE)**, a comprehensive set of development tools, including a visual layout editor, a robust source editor, full debugging capability, and extensive language documentation.

The IDE, and in particular its visual layout editor, enables easy use of our extendible programming language, which encompasses all the capabilities of the full spectrum of languages deployed on the Web today. Curl therefore supports programming at all levels: It enables both the easy formatting and layout associated with mark-up languages such as HTML and the event-driven interactivity and simple dynamic content found in scripting languages such as JavaScript; but because it is also a fully robust object-oriented language, like Java or C++, Curl can be used to build applications with complex business logic, rich graphics, full interactivity, networking, and XML-driven dynamic data presentation.

Purpose-built for Web applications, Curl source code is extremely compact. And since Curl applications are compiled on the client, they are delivered over the network as source code rather than in expanded bytecode or binary form. Curl application downloads are therefore considerably smaller than comparable Java applets or ActiveX controls.

runtime engine and an integrated development environment that includes an extendible programming language—was built from the ground up for the creation of rich client-side enterprise applications, centrally controlled and delivered over the Web. Its strict adherence to Web standards like XML for transmission of data and HTTP for communication enables seamless integration with either a Java-based or a .NET-based backend.

## *Enterprise Apps that are Fast, Efficient, Secure, Extensible*

A Curl-based application works like this: Upon an initial request from a user, the back-end server will send down the app and, optionally, its associated data in a compact file that can be an order of magnitude smaller than a comparable HTML-based application. From that point, the full power of the application is literally in the hands of the user; it runs on the user's local machine, not the server. It will request additional data from the network only as needed.

The server is freed from much of the processing it typically does because the application offloads all of the presentation work and appropriate business logic tasks onto the client. Graphics, for example, instead of being sent down as large files from the server, are generated on the client. The network is freed of the constant back-and-forth requests and sending down of fresh Web pages that is typical today. The huge savings in both server power and network bandwidth make Curl-based applications enormously scalable.

The object-oriented power of the language also means that any Curl-based app is completely extensible. The runtime environment includes more than 4000 APIs that programmers can use out of the box; developers can build their own features on top, with no limits.

## The Business Value of a Client/Web Platform

When enterprise applications are faster and richer and an IT system is more efficient, more scalable and easily extensible, business benefits flow immediately. The direct results are increased productivity, reduced costs and shorter time-to-market.

All of that translates into higher profitability upon deployment. Curl-based enterprise applications deployed across business-to-business extranets will spread the business benefits, enhancing supplier and partner loyalty. In short, Curl technology will position an enterprise as a market leader.

### *Increased productivity*

Jakob Nielsen, a leading usability consultant to major organizations, has estimated the total cost to businesses worldwide of bad intranet usability at around \$100 billion. Huge amounts of IT investment drain away uselessly as employees waste time trying to find data and information. Nielsen urges corporations to stem this flow from their bottom line by thinking about their intranets as productivity tools.

This is the precise focus of Curl technology, which enables more efficient, higher performing enterprise users with the data and information they need at their fingertips.

First of all, Curl-based enterprise applications are **faster**; doing the computing work on the desktop rather than in the network means no "executive wait" time. Just as important, Curl-based apps have **much richer user interfaces** than regular Web interfaces. The Curl language has the capacity for extensive data manipulation built in. It allows for an easily customized UI, to suit the needs of the individual user. Instead of forcing a user to work through page-by-page sequences, Curl-based applications allow single-screen interactions. Users can drag and drop and re-configure the screen as easily as they would on a desktop spreadsheet.

In addition, Curl enables **disconnected operation**. Applications can be built locally so that you can unplug from the network and work with an application offline. That's useful not only when flying but also in situations where the network connection is poor. In that Latin American hotel with the slow Internet line, the harried executive can connect for a short time, download the small compressed files needed (e.g. 30 KB for an executive information system), then disconnect and get to work.

One Curl customer, a division of Siemens, faced a problem common to many enterprises: when decisions were needed, executives couldn't analyze in a timely way the mountain of real-time business data collected by the IT system. Siemens used the Curl platform to create an Executive Information System that gives senior managers a unified view of business results from numerous operating groups across the globe. They can slice and dice the data as they choose and display the results in a variety of two- and three-dimensional visual graphics, each created on the local machine. The presentation is also persistent: next time an executive logs on, the same personally customized view of the data will display.

This is the effect of the productivity boost that Curl technology gives to Web-enabled applications: executives working more profitably.

### *Reduced IT Costs*

A rich client architecture intrinsically provides more efficient network bandwidth and system use. This reduces IT costs in a variety of ways.

- **Large savings in bandwidth.** Forrester cites "up to 98 % reduction in use of bandwidth" in test environments.
- **Large savings in server capacity.** With the backend infrastructure used so much more efficiently, Siemens justified its EIS investment with the cost savings from slimming down its server-side infrastructure. This efficiency also means lower costs to scale the solution. Siemens is now looking into the possibility of deploying the EIS system company-wide.
- **Lower maintenance costs.** A single unified environment for layout, scripting, graphics and object-oriented programming dramatically reduces maintenance costs. And changes to the back end infrastructure don't necessarily require any change to the Curl front end.
- **Lower cost per client.** The optimal use of the client system maximizes the value of existing desktop machines.

### *Reduced Time-To-Market*

The Curl language was designed to offer both the power and flexibility of traditional object-oriented languages as well as the rapid UI development capabilities of presentation languages. This is distinctly different from other object-oriented languages that are designed for logic programming and only offer slow, difficult UI programming. By delivering both of these capabilities in one unified environment, the Curl platform enables more rapid development and easier maintenance than

existing technologies. And once in place, applications built on this platform can easily be extended or modified.

Curl-based applications interface with an organization's existing backend infrastructure. No major integration work is necessary on the servers. And unlike Java or C++, the Curl language—JIT-compiled directly on the client—allows the kind of real-time development and testing associated with simple scripting languages. That, too, enables rapid prototyping and shorter development cycles.

A Curl solution fully custom designed and engineered for a leading asset management firm took three months for a team of three to implement and test.

## Curl Technology is Unique

The two major server-side platforms do not provide for rich client applications. Java on the client-side has a long and failed history riddled with deployment obstacles including a proliferation of different virtual machine implementations.

And Microsoft's massive resources are currently focused on developing .NET servers and tools. A client-side .NET application environment might ship in the next consumer version of Windows, code-named Longhorn. But this release is at least 18 months out. Until then, Microsoft has no platform for client-side Web applications.

In response to the growing frustration with the existing Web-enabled infrastructure—and the market opportunity that this creates—various “rich-client” solutions are now being proposed. Macromedia has positioned itself in this marketplace via Flash MX. But while Flash is appropriate for lightweight applications, in particular consumer-oriented ones, it was not designed for true business applications.

Many other small rich client solutions either adapt, retro-fit or extend the current server-side technology. A typical solution adds another layer to the server-side infrastructure by inserting a “UI server” or “presentation server.” This new server interfaces with a client piece that is essentially a small browser written in Java with smarter pre-built UI components than a regular Web browser. Such a mini-browser might, for example, use XML rather than HTML to send down data in such a way that the interface displays a sortable table instead of a static table. This does give some client-side processing efficiency: no round trip to the server is necessary to re-sort the data in the table.

However, this is a rigid new freedom. The new improved UI is inflexible. You get what comes out of the box and that's it: a suite of standard components and the option to change some settings. A programmer cannot extend it to create his own UI elements.

A more fundamental issue is the reduced scalability and fault tolerance introduced by having this **UI server as a bottleneck** on the network. Extending existing

server-side technologies compromises scalability and introduces new points of failure.

In contrast, the Curl solution to the problems of server-centric systems offers a client-side approach that's completely extensible. It's the first and currently the only technology designed specifically as a Client/Web Platform.

## You Can Do It Now: Business Value Today

In today's economic circumstances, enterprises are keen to invest time and money only in technology that yields business value today.

Much current corporate IT work focuses on the integration of enterprise applications and on tapping the power of Web services. The Curl platform provides the option to leverage such server-side work by dramatically enhancing client-side usability.

For example, another Curl customer, a leading financial services company, used Microsoft's .NET platform to build a Web service that sent down historical securities information. But what use is rich data pulled down off the Web if it isn't in usable form at the presentation end? The customer added at the front end a Curl-based stock charting application with a dynamic user interface that offers rich data visualization and manipulation in a highly interactive, personalized setting. Further premium services can easily be added later, extending the product line.

Many large organizations are also seeking to enhance usability on the server side by building enterprise portals. Typically these are multi-paned digital desktops designed to integrate and accumulate corporate data and information in a streamlined, standardized way.

But you cannot entirely solve the usability problem with a server-side model. A rich client application is a logical extension to enterprise portal work. The Curl Client/Web Platform delivers exactly the usability such enterprises are searching for.

Curl technology is available today. It can be rapidly developed and deployed. It will seamlessly interface with any existing back-end system. Because of the power and extensibility of the language, it's a future-proof investment: a rich Client/Web Platform to support next-generation, Web-deployed enterprise applications.

## Conclusion

Many enterprises today are struggling to get the best out of their Web-enabled applications, frustrated by poor performance, low usability and the inefficiencies of a server-centric system.

Curl has a solution: enterprise applications that move computing power to the client device, though the app is still delivered across the Web. It's a way of preserving the gains of Web-deployment—central control and IT administration with low TCO—while leveraging the power of the client to dramatically enhance productivity.

While some offerings attempt to deliver similar benefits by adapting, retro-fitting or extending the current server-side technology, Curl has designed from the ground up a whole new architecture: next-generation client-side technology that complements the existing infrastructure. The Curl Client/Web Platform enables client-side Web applications that reach seamlessly into any back-end system, leveraging those server-side Web application investments to optimal effect.

The resultant UI improvements will drive productivity up while the dramatic system efficiencies will lower costs and reduce time-to-market.