Do it Once. Do it Right.
MAGIC SOFTWARE

WHITE PAPER

Overcoming RIA Development Challenges:

With End-to-End Application Platform Technology

By Avigdor Luttinger,
VP Corporate Strategy

April 2009

Introduction

The significant benefits that Rich Internet Applications (RIA) convey upon the Enterprise have already been well documented and are fully recognized, ranging from lower cost of ownership, to overall application availability, better security and enhanced user experience.

However, while Enterprise RIA's do provide sizeable advantages (vital to maintaining competitiveness), these benefits come at a price: system complexity. **This white paper examines the challenges to effective RIA development and offers a navigational route to RIA designed to avoid current problems and maximize the potential benefits.**

Until recently, there have been two ways of effectively navigating full RIA delivery:

1. **Use a hosted Platform-as-a-Service (PaaS)**. Examples include Force.com, Google, Microsoft, Amazon and LongJump. If this option is chosen, then consider that while all parts of the system are already provided for you, that platform remains somewhere in the Cloud.

With RIA you gain from:

Lower total-cost-of-ownership

With RIA there's no need to install Client-side software or maintain every new user. The application is set up automatically and transparently on the Client side and all application and platform updates take place on a centralized server.

SaaS enablement

SaaS applications will account for 25% of all new business software by 2011*. RIA is an essential component in developing a successful Software-as-a-Service (SaaS) offering.

Better customer user experience and productivity

Unlike typical internet applications, RIA's don't need Client-Server communication for every data process or activity. Your application runs faster and smoother, your employees get their jobs done faster and your customers get a better service.

More application availability

RIA's can run remotely from anywhere and be accessed at any time via a wide range of hand-held mobile devices. Employees can access work data from home or on the road.

Better corporate security and protected intellectual properties

RIA's feature multi-tier architecture allowing you to more effectively hide the sensitive elements of your application and avoid corporate theft, customer asset abuse and malware intrusion.

2. **Use a Client-Side platform**. For Enterprises at odds with the concept of their assets sitting in the Cloud, the only available means of acquiring a RIA-build capability was, until recently, to do it yourself using a toolkit of the Client RIA Platform variety (such as Adobe Flex/Air, Java FX or Microsoft Silverlight).

(*Gartner Report: SaaS Delivery Challenges On-Premise Software, 26th September, 2006)

Behind the Hype: The Reality of the Cloud

For smaller or growing businesses, Cloud computing conveys considerable advantages. New IT capabilities such as storage and processing can be added incrementally instead of incurring the entire expense of servers along with associated maintenance costs. Servers also take up room. They require data centers, managers, air conditioning, electricity supply, back-up, security and constant upgrades, system performance optimization, patches and hardware replacements when they go wrong. Cloud computing offers an attractive and simple way of outsourcing these IT burdens in return for a simple subscription that has a low entry and exit threshold for businesses new to the game.

However, once you opt to transfer your systems online, moving from local storage and processing to the Cloud computing model, you have effectively outsourced your mission-critical data to an outside company. Behind the hype, the 'Cloud' still consists of servers, hard drives and network connections. The Cloud is real. Cyberspace is an actual place and therefore susceptible, in the same way that local hardware is, to outages, downtime and shoddy service from providers.

As we become more reliant upon the Cloud, vulnerability may also increase as data protection becomes more of a challenge. Also important to remember is that providers are only companies, just like any other - and they can make mistakes. What happens when spreadsheets, customer data and sales presentations suddenly become inaccessible because of a problem at the side of the hosted platform provider?

- ***Recession vulnerability***

As the effects of this recession continues to reverberate into all sectors of our business economy we see that even the largest and seemingly stable enterprise names can file for bankruptcy and disappear overnight – sometimes with no hints or warnings to their customers who are then left to pick up the pieces. What if tomorrow it was your platform provider?

The answer is therefore, to maintain the option of operating your mission-critical IT systems in-house, if the cost of doing so makes this feasible.

But there again, companies are being inhibited by the limited scope of Client-side RIA build platforms.

The Fallacy of Client Side Only Solutions

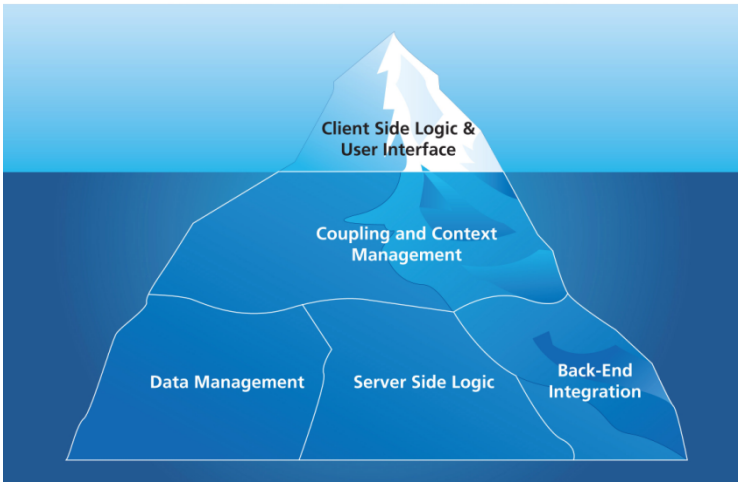
While CIO's may rightly be attracted to the concept of Enterprise RIA and want to experiment and gain hands-on experience within their organizations, they must first understand the implications of build-it-yourself RIA platforms that are Client-side only. Beyond creating a great looking user interface – which can be viewed as the tip of the iceberg - there's far more under the RIA surface to consider.

Beneath that Client surface is a multitude of complex moving parts that can become a major development stumbling block that ends up killing expectations, forcing CIO's to shelve projects and give up on the promised benefits. This is bad news for IT departments looking to improve their services and give the enterprise the support it needs to compete in the present business climate.

The RIA Iceberg – The Most Challenging Development Process Yet

RIA's represent one of the most challenging development processes yet. They involve three clear tiers – a ***Client tier*** (taking care of the presentation, interaction logic and some business logic), a ***Server tier*** (taking care of most of the business logic, the data and backend integration), and a ***Session tier*** (the inter-lying communication layer between the Client and the Server that requires system programming skills).

Developing the Client Side Logic and User Interface Represents Just the Tip of the Iceberg for RIA Development



So, a typical RIA development effort requires building and maintaining a number of different teams to work on the different aspects of the application. As a result, the design, planning and management of the project becomes riskier and of course, more expensive. As with any system – where there are more moving parts, there is more chance of a breakdown occurring somewhere along the line.

- ***The Coupling Challenge***

Traditional Client-Server applications involve fairly simple architecture, relying upon a permanent connection between the Server and the Client. With a tightly coupled design, there is no need to explicitly manage or preserve various logic states.

Conversely, web applications, which centralize their processing in the Server, leave the Client essentially decoupled, or loosely coupled. As long as web applications feature short and simple logic processes, and a limited richness of interactivity, they can be usually implemented with standard web architectures and simple session management.

RIA architecture however, attempts to provide the comfort of a tightly-couple and highly interactive user experience in a loosely-coupled setting. This requires sophisticated system design and programming skills, which are rarely found in development teams focused on business solutions.

At present, the only platforms which support this type of architecture without having to resort to system-level programming are the so-called **metadata-driven platforms** – such as Magic Software's uniPaaS, Force.com, or Cordys.

- ***Making up for Bandwidth Limitations***

As bandwidth has steadily increased, more resources are now available for web applications, bringing us closer to the level of Client-Server or LAN interaction styles with technologies such as Ajax. We see this with the 'Google Suggest' function that brings up a range of popular search terms for every successive letter of the search term you key into the search field.

But while this may be fine for browsing purposes, broadband internet is not sufficient for tightly coupled business applications with tens of interactive fields per screen.

To get around this limitation, RIA's must support more logic at the Client end and must partition processing between the Server and Client. So developers end up working with two physically separate and logically dependent processes – that also run in tandem on both the Client and the Server.

The Client now requires a 'brain' and 'memory' to deal with the constant instruction sequences from the Server. Keeping the session coherent therefore requires sophisticated state and session management.

So while traditional Client-Server applications required just a business developer, to create a non-hosted RIA using a Client-side platform means adding system programming skills to the team, adding considerably to the final cost of the solution.

The Development 'Wall'

Because Client-side RIA platforms are popular with web developers today, we see a number of companies beginning RIA initiatives that are conceived in the web department. By nature, web developers tend to focus on the rich experience of the application's user interface, and for this, technologies such as Flex or JavaFX certainly do the job.

However, web developers generally lack a full system view, and so they end up running into architectural difficulties as soon as they attempt to move beyond the Client. Some attempt to use Ajax, but this also has performance limitations, and in many cases the project will end there.

Typically, a web team may spend up to half a year developing a Client-side web user interface. Then they run into a 'wall'. While a Client-side toolkit will provide the Client 'brain' and 'memory' in order to get around the bandwidth limitations of the web, it won't provide the connectivity or business logic – skills that are also not usually available in such teams.

Even those organizations that persevered with the full three-tiered development process go on to experience difficulties in maintaining the solution and keeping up to date with evolving technologies and standards. This is a huge inconvenience that organizations experimenting with RIA are only now beginning to discover.

Performance & Optimization

Developing and deploying RIA involves multiple communication scenarios between the Client and Server sides of the application. **When opting to subscribe to an end-to-end hosted platform, the platform provider invests the necessary resources to make sure the application is tuned and optimized for full performance.**

However, when an organization opts to use one of the Client-side toolkits mentioned, they will have to develop their own system performance in-house. Since this also requires the specialist skills which most business organizations are not generally familiar with, the organization will once again be forced to invest more time and

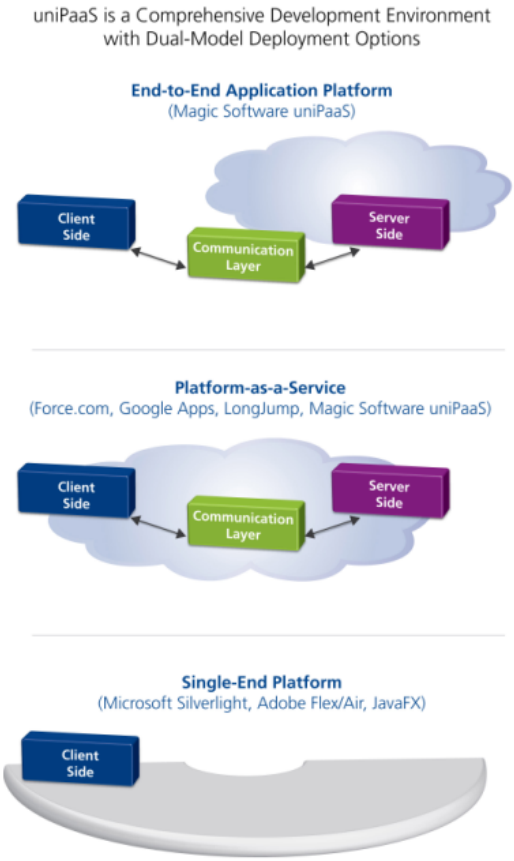
resources trying to tweak an application with inferior performance and optimization to the hosted platform alternative.

Tuning involves reducing the communication scenarios between Server and Client to a minimum and optimizing data package size to limit contention on the network and achieve fast application response time. Using Client side tools means that Server and Client development take place separately - using two different paradigms and teams. As a result there's more chance of network bandwidth wastage.

A further problem is that this lack of optimization only comes to light after full deployment of the application has taken place, and only when customers and users begin to complain about sluggish performance. The developers will then have to go back to re-optimize their communication scenarios – wasting more time, incurring more costs, and gaining a bad reputation with customers in the process.

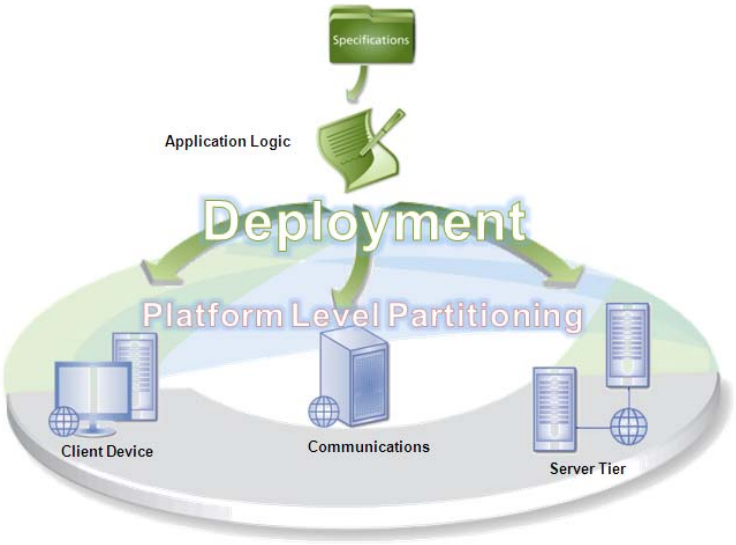
The 'End to End' Solution

There is now a new breed of end-to-end RIA application platforms that can provide a comprehensive answer to the challenges of RIA development and deployment. **One such platform is Magic Software's uniPaaS, which gives users the choice to deploy both on-site and/or in the Cloud as a hosted Platform-as-a-Service (PaaS) offering.**



The key difference is that this type of 'full' platform provides all the parts of the solution – including the hidden part of the 'iceberg' – without requiring separate development. Hence 'end to end'. **The solution uses metadata rather than hard coding so all that's really needed is to describe the application's business logic and design the compelling user interface.** The platform then takes care of the rest, similar to the capabilities of the hosted PaaS – **but without the limitation of cloud-only deployment.**

uniPaaS Automatically Partitions and Deploys Application Logic Across the Client, Server and Communication Sides of the Application



Using a unified development paradigm, uniPaaS can deliver advanced RIA and SaaS web applications incorporating all aspects of the development and deployment process under one roof.

By combining both the Client and Server side development logic within the same paradigm, businesses can avoid the cost and effort of gathering the separate resources to build their RIA web applications, thus reducing project risk and eliminating much of the maintenance costs once the application is deployed.

uniPaaS can also manage all the setting and controlling of the Client side and Server side logic, in addition to the communication between the Client and the Server, including the consumption and manipulation of back-end services - all using the same language.

It is also the only solution to support the entire application delivery spectrum – desktop, client/server, web, RIA and SaaS, with the same application copy.

UniPaaS Application Platform	***Popular RIA Client Toolkits***
End to End platform	*Client Side platform only*
Metadata (with optional code)	*Requires intensive programming*
Platform managed Application Update (transparent)	*Requires explicit coding of Application Update via API*
Platform session and context management	*Requires explicit coding of session and context management*
Single skill set required for both front end and back end	*Requires 3 different skill sets for full project development*
.NET platform support	*Cross platform Client support*

Conclusion

With an end-to-end, metadata-driven application platform, navigating the Enterprise RIA challenge was never easier and the rewards never made more sense: businesses with a RIA offering can give their software users an experience surpassing the richest Client-Server applications, plus the incredible fact that the application is available over an http connection – thus ensuring that employees and partners can benefit from fully functional applications wherever they happen to be. A win-win scenario for Enterprises, ISV's and their customers!

For more information about Magic Software and the uniPaaS application platform visit www.magicsoftware.com