



FITMENT OF RIA IN ENTERPRISE ECOSYSTEM



WHITE PAPER

Rahul Deo

December 2008

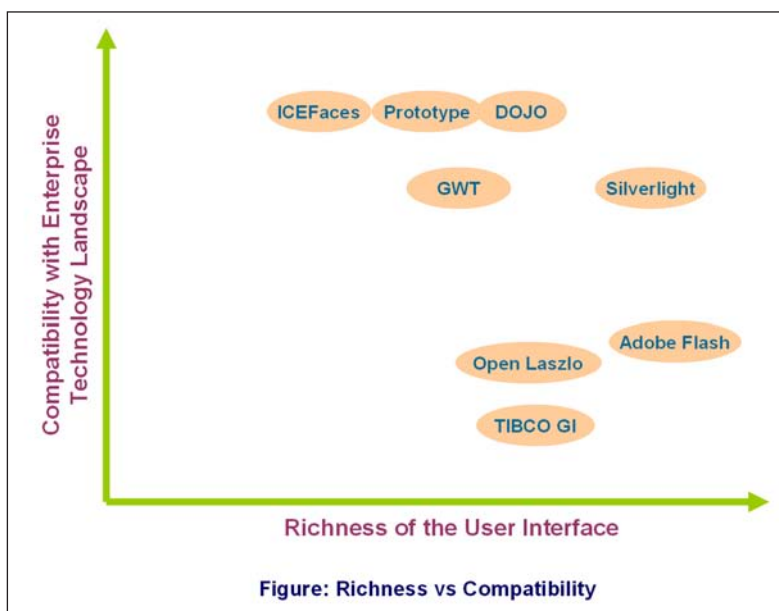
1. Introduction

After several years in existence the Internet is mature and the primary platform for Graphical User Interfaces. Though it has delivered well, people missed the richness of the old fashioned desktop application. In order to address this issue, very first steps were taken using AJAX. This was the evolution of Rich Internet Application(RIA).

Initial adopters solved the problem using the combination of AJAX and DHTML. This approach gave a major boost to the web application development and became the front runner in the RIA technology.

Though this approach brought a lot of richness, it was still not close enough to the richness of the desktop application. So, finally evolved the era of browser plug-ins. Few plug-ins were already in existence, hence adaptation of them was easy. Competition brought more plug-ins to the market. Along came the constraints and complexities involved in the decision making.

This document tries to discuss some of the prevalent technologies, their features, implementation nuances and the possible future of Rich Internet Applications.



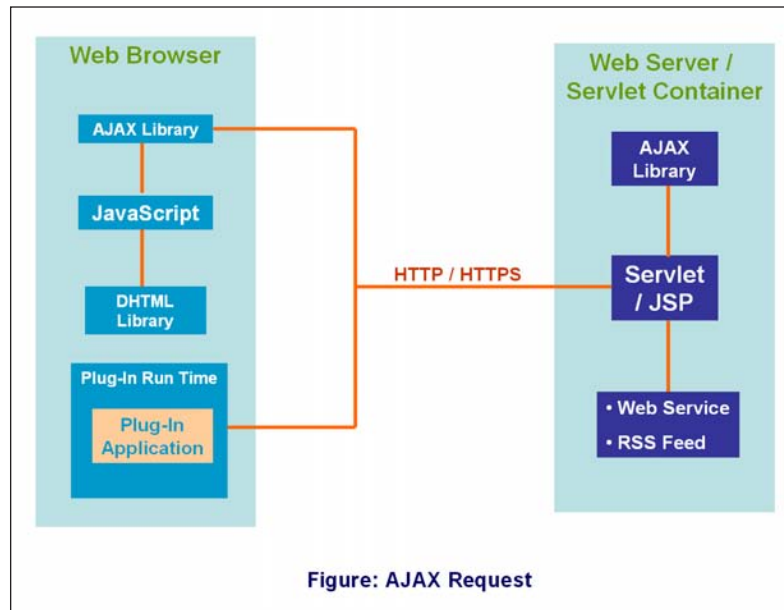
2. Anatomy of Ria

From a high level architecture perspective, the design of a RIA application is almost the same as any web application, except for the fact that the server request doesn't refresh the whole page. Each request is sent in an asynchronous way and handled accordingly by the client and the server.

Plug-in based applications generally run within a plug-in runtime. This runtime provides mechanism for the server communication.

For non plug-in based applications, server communication is generally handled by an AJAX library.





3. Prevalent implementations

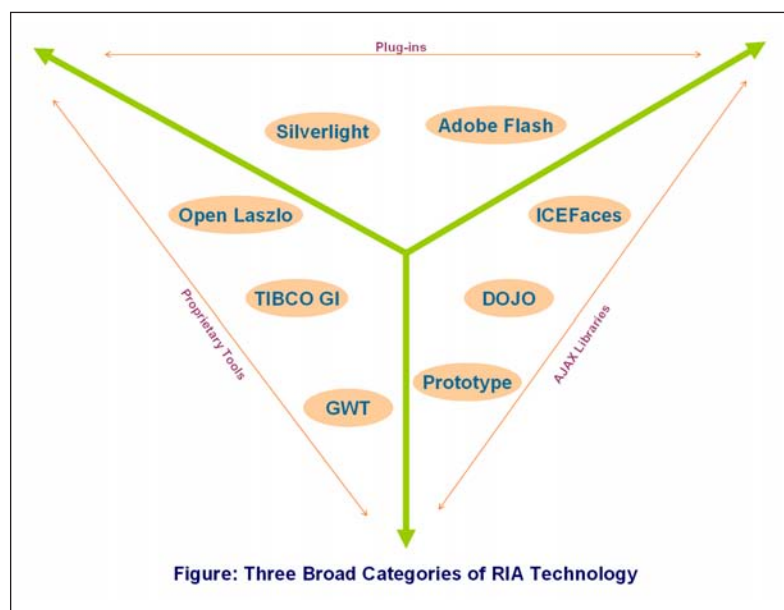
There are many implementations available today in the market. They vary from simple AJAX JavaScript libraries to full fledge AJAX framework and plug-ins.

If we have to categorize the available products in the market, they can be categorized in three broad categories:

- **AJAX libraries:** These libraries provide graphical (DHTML) components along with JavaScript libraries. While graphical components provide the richness, JavaScript libraries provide the asynchronous communication. Most of these products can easily be integrated with the existing enterprise technology landscape.

- **Plug-ins:** These products provide the best richness in terms of graphical user interface. Product vendors use their proprietary language for development and delivery of content. Though the content is delivered over HTTP, the structure is not understood publicly.
- **Proprietary Tools:** Some of the tools in this category produce results which are publicly understood while others are not. Some use their proprietary language" while others use an industry standard language, like Java. From a development stand point, these products are close to plug-in based products. But compiled end products run seamlessly on a browser. In fact, some give an option to compile in DHTML or plug-in consumable format.

A few of the market leaders in each segment are displayed in the chart below.



1. Google Web Tool Kit (GWT)

1. Overview

GWT is a proprietary tool. It allows users to write code in Java and when ready, convert the whole application into a web application. The GWT compiler generates JavaScript.

2. Features

High performance JavaScript: GWT produces AJAX apps that load faster than hand-written JavaScript applications. It uses smaller, more compact, cacheable code and supports IE, Firefox, Mozilla, Safari, and Opera. GWT uses the browser's "back" button correctly.

Better development tools: Since you're writing in Java, you can use IDEs that you love like Eclipse, IntelliJ, and NetBeans, which provide full-featured debugging, with variable watches and breakpoints. Unit tests (based on JUnit) both in a debugger and in a browser.

Google APIs and reusable UI Components: GWT comes equipped with useful libraries. Built-in UI components serve as cross-browser building blocks for your app. GWT Remote Procedure Call (RPC) helps with client-server interaction. JavaScript Native Interface (JSNI) simplifies integrating GWT code with existing JavaScript code.

3. Constraints

- Extensive use of JavaScript
- No options available for browsers that have disabled JavaScript
- UI/UE developers have limited control over it
- GWT has its own compiler to convert Java to JavaScript. Historically automated code generation has not been well accepted by developers

2. ICEFaces

1. Overview

ICEFaces is an integrated AJAX application framework that enables Java EE application developers to easily create and deploy thin-client rich Internet applications in pure Java. It qualifies as an AJAX Library, not a proprietary tool.

ICEFaces leverages the entire standards-based Java EE ecosystem of tools and execution environments. There are no applets or proprietary browser plug-ins required. ICEFaces applications are Java Server Faces (JSF) applications, so Java EE application development skills apply directly and Java developers are isolated from doing any JavaScript related development.

2. Features

Rich User Experience: Leverage the ICEFaces Component Suite of 50+ AJAX JSF components and create a new class of enterprise applications that will transform the user experience.

Open Source: The ICEFaces developer community is extremely vibrant, already exceeding 32,000 developers in 36 countries worldwide (Sep 2008).

Standards-based: ICEFaces is a pure Java enterprise solution so developers can continue to work the way they are use to. Existing Java IDEs and test tools can be leveraged during development.

Transparent AJAX: ICEFaces is much more than a component suite; it is an AJAX application framework that provides ease of development for Java/JSF developers.

Compatibility: ICEFaces supports the broad array of Java Application Servers, IDEs, third party components, and JavaScript effect libraries of any enterprise AJAX framework on the market.

AJAX Push: ICEFaces support AJAX Push enabling server/application initiated content rendering to the browser. With AJAX Push, developers can create collaborative and dynamic enterprise applications like never before.

Security: It is compatible with SSL, prevents cross-site scripting, malicious code injection and unauthorized data mining. ICEFaces is a server-based AJAX solution; so it doesn't expose application logic or user data plus. it as it especially effective at preventing fake form submits and SQL injection attacks.

Scalability and Clustering: The Asynchronous HTTP Server (AHS) provides high-scalability support for ICEFaces applications that utilize AJAX Push and must be deployed to high volumes of concurrent users - a critical requirement often not addressed by other solutions. ICEFaces also now supports third-party Application Server Asynchronous Request Processing (ARP) APIs provided by Glassfish (Grizzly), Jetty, Tomcat, and JBoss.

Professional Enterprise Software: ICEFaces offers enterprise level support packages that are second to none in the industry. Some of their supported customers include JPMorgan, RIM, T-Mobile, EDS, Siemens, and CSC, just to name a select few.

3. Constraints

- JSF based solutions produce their own markup which takes away flexibility from the hands of HTML developers.
- JSF libraries provided by vendors are always at a risk of the vendor going out of business.

3. TIBCO General Interface (GI)

1. Overview

TIBCO General Interface™ is an open source project focused on enabling rapid creation of reliable AJAX applications, components and portlets with the look and feel of desktop graphical user interface applications. It qualifies as a proprietary tool.

2. Features

- Over 100 ready-made AJAX components, and hundreds of more functions
- Visual AJAX tools for rapid assembly, coding, and connectivity to services
- A “server-less” architecture ready to use with what you already have
- Ease of Deployment: GI applications are composed of JavaScript, XML, and CSS files. To deploy, you simply include a reference to these files in an HTML page that can be accessed from any server.
- Support for Industry Standards - General Interface uses industry standards and widely accepted technologies in both its underlying code (JavaScript, XML, and CSS) and overall architecture (Model-View-Controller and multi-tier design).
- Scalable Architecture - Unlike traditional server architectures, TIBCO General Interface runs on the client, using the client CPU for data presentation and interaction. Once the TIBCO General Interface application is accessed, only raw data is requested from the server.

3. Constraints

- Proprietary development tool may pose challenges while developing an enterprise grade application which uses industry standard IDE e.g. Eclipse.
- Markup is auto generated by the tool which takes away control from the HTML designers.

4. Adobe Flash

1. Overview

Adobe® Flash® provides best RIA platform available in the industry. It is plug-in based, and data is transferred in the chunk of XML. Penetration of its plug-in is 99% (through industry research reports) among all internet users. It started with graphic content (Movie, Animation, Audio, etc), but now the same platform has been extended as a leading platform for Rich Internet Application.

2. Features

- **Design with ease and efficiency:** There are multiple features when used can bring design to life. Few such tools are animation tools like Object-based animation, Motion presets, Motion editor, Motion paths, etc. Other tools are around Rich drawing capabilities, Intuitive user interface, workflow, coding and scripting tools.
- **Explore new creative possibilities:** Creative designers can realize their expertise using animation, drawing and interactive design tools. One can also enrich their projects with advanced video and audio capabilities.

- **Collaborate across workflows:** There are multiple time saving opportunities using common Adobe tools and interfaces. They can easily exchange assets between applications with powerful new file formats.
- **Deliver for a wide reach:** Lately internet has been available beyond Desktop browsers. Adobe AIR (Adobe Integrated Runtime) provides access of content through web, mobile and now desktop. With desktop version of AIR, it is now possible to achieve richness of traditional desktop application and still leverage the advantages of internet based applications.
- **Flex Development:** Through Flex, Adobe added great deal of Application Development capabilities to its RIA offering. Flex uses MXML (declarative XML-based language) to describe UI and ActionScript to create client logic. Flex includes more than 100 proven UI components for RIA along with a debugger. RIAs built using Flex can run within browser using Adobe Flash or on the desktop within Adobe AIR (cross-operating system runtime). Using AIR, Flex applications can now access local data and other system resources on the desktop. While the SDK is free, Adobe provides an Eclipse based IDE named Flex Builder. This IDE is available with a price tag.

3. Constraints

- Require a plug-in to be downloaded for the supported browser
- Flash plug-in is weak on accessibility capabilities. Screen readers for blind can't read flash files
- Search engines can't index data contained in Flash binary data

5. OpenLaszlo

1. Overview

OpenLaszlo is an open source platform for creating zero-install web applications with the user interface capabilities of desktop client software.

OpenLaszlo programs are written in XML and JavaScript and transparently compiled to Flash and, with OpenLaszlo 4, DHTML. The OpenLaszlo APIs provide animation, layout, data binding, server communication and declarative UI.

2. Features

- Visual provided by the UI Components are excellent.
- Final product can be compiled in DHTML or Flash.
- The compiled application can be deployed on a Servlet Container or App Server.
- Application architecture fits well with the J2EE Application Architecture.

3. Constraints

- Developer need to learn an XML based language LZX. The language not being popular, it is difficult to build expertise around this technology.
- Applications compiled in Flash must download Flash Plug-In.
- No IDE provided is by the vendor. But some third party eclipse Plug-Ins can be used to develop LZX based applications.

6. Windows Presentation Foundation (WPF)/E - Microsoft Silverlight

1. Overview

Microsoft Silverlight is a cross-browser, cross-platform, and cross-device plug-in for delivering the next generation of .NET based media experiences and rich interactive applications for the Web.

2. Features

- Application can be developed using .Net technologies. Since .Net is a popular technology, it is easy to build a team around this technology.
- Silverlight comes with Plug-in for Visual Studio.
- Compiled applications work on multiple browsers and multiple operating systems.
- Cross domain network is accessed through policy files.

3. Constraints

- Plug-In must be installed on the browser in order to access the Silverlight Application.

7. DOJO

1. Overview

DOJO is a JavaScript library. It also provides DHTML based visual components which can be used for a rich user interface.

2. Features

- Fits well with the existing enterprise technology landscape
- Browser detection, JSON encoding/decoding, powerful AJAX support
- Asynchronous programming support (dojo.Deferred)
- High-performance CSS3 query engine
- Memory leak protection, Firebug integration, Unified data access (dojo.data)
- Drag and drop, i18n support, localizations, back button handling

3. Constraints

- Final HTML Mark-up is auto generated which takes away the control from HTML designers.
- Few features may not work effectively on a slower connection.

8. Prototype & Script.aculo.us

1. Overview

Prototype is a lightweight JavaScript Framework which aims to ease development of dynamic web applications.

It features a unique, easy-to-use toolkit for class-driven development.

Add-on to Prototype named script.aculo.us compliment this JavaScript library with easy-to-use, cross-browser user interface.

2. Features

- Supports AJAX call to the server; Deals well with the data returned by the server and provides helper classes for polling
- The biggest part of this framework is its DOM extensions
- Prototype 1.5.1 features JSON encoding and parsing support.
- Support for the creation of classes and also support inheritance
- Through an add-on named script.aculo.us Prototype provides excellent UI support
- Few of the features are "animation framework", "drag and drop", "AJAX controls", "DOM utilities" & "unit testing"

3. Constraints

- Final HTML Mark-up is auto generated which takes away the control from HTML designers.
- Though these products claim to work with most of the browser, it is advisable to test the final solution on the desired browser.

4. Leaders

The year 2007 has been very exciting for the RIA market. Each category discussed above attracted good traction. Based on customer demand, we could find following technologies in the leadership segment.

Plug-in: Adobe AIR

AJAX Libraries: DOJO

Proprietary Tools: GWT

5. Developers' view

1. Development Tools

Each technology has a different way of handling development. Some provide eclipse plug-ins while others depend on their proprietary tools. Details for each product can be found in the comparative study section.

2. Documentation

Commercially licensed tools have good documents whereas the open source is evolving. Being first is the key to the vendors of this technology. They admit that focus is to get the features released first, followed by the documentation.

3. Learning Curve

Though learning of a technology depends on the developers' existing skills and learning capabilities, a quick evaluation of these technologies allowed to come up with the following projection. This projection assumes that the developer has average knowledge of

his/her domain, be it Java or .Net.

4. Testing

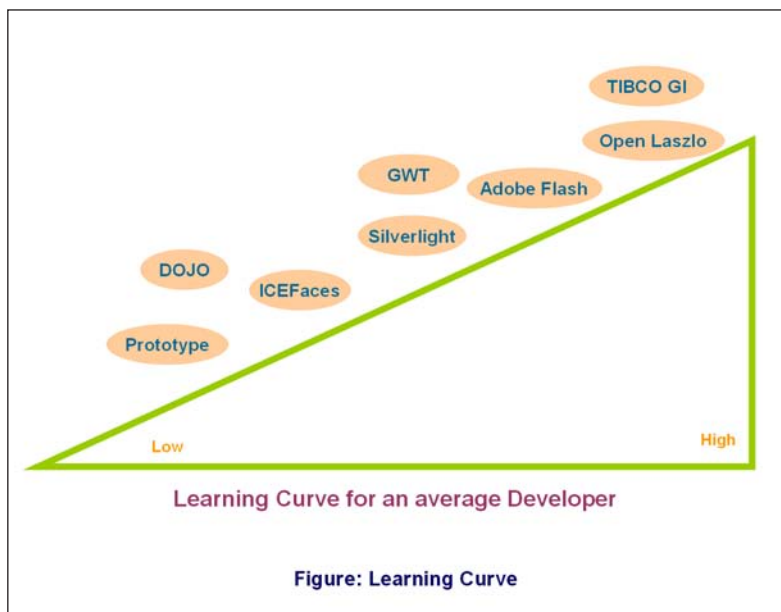
Testing has always been a very important part of the development cycle. Testing a RIA application depends on its underlying technology. Quick observations reveal that technologies which are compatible with an enterprise technology landscape have a more robust testing mechanism. Details of each product can be found in the comparative study section.

5. Source Control

Observation on testing holds good for source control as well. Technologies which are more compatible to the enterprise architecture support better source control.

6. Scalability

Most of the RIA technologies have been around only for a short while. Vendors admit that focus was to get the features out and not on documentation and robustness of their internal architecture. But lately leaders of RIA have committed towards scalability, and data is available towards their respective level of scalability. Leaders like Adobe and DOJO have demonstrated that their products can scale to enterprise grade.



6. Commercial view

1. Licensing

Given the demand for open source, most of the RIA technology surfaced by riding the open source wave. But some of the popular ones are available only through commercial licensing. However, license cost is primarily for the development part of it and not for the delivery. More details can be found in the comparative study section.

2. Availability of skills

RIA is new to the market, so skills are not readily available today. Most of the companies are developing their own skills or leveraging resources close to these technologies. Again developing skills depend on the learning curve and future of these technologies.

3. Development Cost

Most of the products come with robust visual components and JavaScript libraries which cut down

the effort/cost of development. Also some of them allow auto generation of code which dramatically cut down the development effort. The flip side of such auto generation is, they are very hard to integrate with the existing architecture and debugging is always a pain.

4. Support Cost

Support cost is mostly dependent on how well the system has been designed and developed. But, if we have to compare the available products, one which allows more auto generated code will be high on support cost.

7. Future direction

1. Research Reports (reference Adobe.com)

Forrester - "Most of the RIA software specialists we interviewed for this report tell us that 40% to 60% of all of the companies that buy their products and services are building employee-facing apps or turbo-charging their enterprise portals with RIA technology." [Click for Full report](#)

Forrester - "Rich Internet applications are well suited for financial services interactions that are extremely complex, rely on graphics and models, and often require data from multiple sources." [Click for Full report](#)

2. Market Support

The market is really excited about these technologies. CIOs have started making RIA part of their IT strategy. There are many new businesses which are possible now with RIA in existence.

Software bigwigs like IBM, Oracle and Microsoft are committed to make RIA part of their products. AJAX and DHTML is core of all the non plug-in based products but each of them follows its own standards when it comes to implementation.

3. Acceptance in Developer Community

For developers, it is another opportunity to add to their skill sets. RIA being close to user interface and markups, traditional application developers find it difficult to adapt this. Evolution of RIA has also brought a new breed of developers who do more of UI specific coding.

8. References

1. **XUL:** <http://www.xulplanet.com>
2. **Microsoft Silverlight:** <http://silverlight.net>
3. **Google Web Toolkit:** <http://code.google.com/webtoolkit>
4. **ICEFaces:** <http://www.ICEFaces.org/main/home/index.jsp>
5. **TIBCO GI:** <http://tibco.com/devnet/gi>
6. **Adobe Flash:** <http://www.adobe.com/devnet/>
7. **OpenLaszlo:** <http://www.openlaszlo.org>
8. **Dojo Toolkit:** <http://www.dojotoolkit.org>
9. **Prototype:** <http://www.prototypejs.org/>
10. **CTP Advisory:** <http://ctp-advisoryservices.blogspot.com/2008/09/future-of-enterprise-portals.html>

9. Comparative Study

	Development Tools	Language	Testing	Pricing	Documentation	APIs	Industry Support	Adherence to Standards	Client Plug-In
GWT	Eclipse	Java, JavaScript	JUnit	Open Source	Good	Java	Given the Google brand, it will attract industry support gradually.	May not fit well with the Enterprise Java Architecture	No. Final produce is in the form of Java, JavaScript and HTML
ICEFaces	Eclipse	Java	JUnit	Open Source	Good	Java	Good. Compliance to JSF will allow industry to support this product.	Excellent as based on JSF which fits well with Enterprise Java Architecture	No. Final produce is in the form of HTML.
GI TIBCO	TIBCO	XML	NA	Open Source	Good	JavaScript	Good. Compliance to existing standards e.g. XML, CSS, JavaScript will allow industry to support this product.	Adherence to XML, CSS & JavaScript allows the compiled result to be deployed easily	No
Adobe Flash	Adobe/ Open Source/ Eclipse based	AJAX, FLEX(MS-XML, ActionScript), FLASH	Adobe	Commercial License. Plug-in is free	Excellent	AIR	Excellent. Adobe plug-in is widely popular.	Apart from delivering content over HTTP it follows no standard.	Yes
Open-Laszlo	Open-Laszlo Server	LZX	Open-Laszlo	Common Public License Version 1.0	Good	LZX	Compiled result is delivered in DHTML and Flash which has good industry support. However proprietary language may not get good support.	Final product can be compiled to DHTML or FLASH.	No, if compiled to DHTML
Microsoft Silverlight	Visual Studio	C#	Visual Studio	Commercial License	Excellent	C#	Supported by Microsoft and its partner	Final product can be deployed on multiple platforms.	Yes
DOJO	NA	Java	DOJO	Open Source	Excellent	Java	Excellent. Compliance to J2EE gives this product excellent industry support.	Fits very well with J2EE	No
Prototype	NA	JavaScript	Prototype	MIT (Source Code)	Excellent	JavaScript	Excellent	Fits very well with Enterprise Web Architecture	No

Contact us

USA

460 Park Avenue South
Suite #1101, New York
NY 10016, USA
Tel.: +1 212 686 6655
Fax: +1 212 686 2422

UK

88 Wood Street
London EC2V 7RS,, UK
Tel.: +44 20 85281000
Fax: +44 20 85281001

Australia

410 Concord Road
Rhodes, NSW 2138, AUS
Tel.: +61 290 221 146,
Fax: +61 290 221 134

INDIA

Bagmane Technology Park
Byrasandra Village,
C.V. Raman Nagar
Bangalore 560 093, India
Tel.: +91 80 4004 0404,
Fax: +91 80 4004 9999

About Mphasis

Mphasis is a leading Applications, Infrastructure Technology, and BPO services provider.

The company delivers real improvements in business performance for clients through a combination of technology know-how, domain and process expertise. With currently over 36,000 people, Mphasis services clients in Financial Services, Healthcare, Communications, Media & Entertainment, Transportation & Logistics, Energy & Utilities, Consumer & Retail, and Governments around the world. To know more, visit www.mphasis.com.

Mphasis and the Mphasis logo are registered trademarks of Mphasis Corporation. All other brand or product names are trademarks or registered marks of their respective owners. Mphasis is an equal opportunity employer and values the diversity of its people. Copyright © Mphasis Corporation. All rights reserved.

