

White Paper:

PHP Leads Web 2.0

A Closer Look at the Hidden Drivers and Enablers
of the Second Internet Revolution



By Andi Gutmans, Co-Founder and Vice President of Technology,
Zend Technologies, Inc.

September 2006

Everyone is currently talking about Web 2.0, it is the next big thing in the IT industry. But most people have only a vague idea of what Web 2.0 is about – and what it is not. They tend to think of Web 2.0 as a collection of websites and fancy web-based applications. What they don't see is the shift of paradigm that Web 2.0 brings – and the emergence of new technologies under the surface.

Defining Web 2.0

If you ask 20 people in the IT industry to define what Web 2.0 is, it is most likely that you will receive 20 different answers. This shows one of the main problems when talking about it: Web 2.0 is not a fixed standard or product, and the viewpoint of the individual influences his perception of Web 2.0. For consumers or journalists outside the industry, Web 2.0 is a number of applications, websites and interfaces – Google, Flickr.com or even eBay. From a developer's perspective, it is a collection of APIs, formats and code. And a CIO or CTO of a large company might see a new approach for software architecture that helps him to improve his application landscape.

But it is hard to write an article about a new trend without giving a proper definition of it, isn't it? Though my definition is definitely subjective, I tried to find some least common denominators. In my opinion, Web 2.0 usually includes one or more of these elements:

- Rich Web Applications are most likely built with AJAX technology. Even though Web 2.0 is not only for public websites or applications, there is a large movement to build internal enterprise applications with web front ends to achieve platform independence and make it easier to deploy, manage and access business applications.
- SOA (service oriented architecture). SOA means that a website or web application (or even a server-only application) exposes functionality via a service most commonly in our world, a web service. This makes the reusing services and the creation of new applications, the so called mash-ups, very easy.
- Social Web elements. Almost all popular Web 2.0 applications offer collaborative or social functions that allow users to commit themselves and create new content. The user takes part in content creation, whether actively or passively.

A Look Under the Hood – the Gears and Engines of Web 2.0

At this stage, most people perceive Web 2.0 through the rapidly increasing number of dynamic websites like Flickr, Facebook, MySpace or Yahoo!. Compared to classic websites they are far more interactive and behave more and more like classic desktop applications, mostly thanks to AJAX, a client-side technology that resembles traditional user interface behavior in a browser. A good example is Zimbra¹, a web based PIM application that offers the functionality of classic programs like Outlook or Thunderbird. It looks like a desktop application, it reacts almost as quick (or slow) as a desktop application – so is it a desktop application? Yes and no. Yes, because it performs all the tasks and actions you demand from it. No, because the underlying technologies and deployment methods are completely different.

Let's have a look at classic methods of developing, deploying and using applications. Usually, applications are written for a specific platform, be it Windows, Linux or the Mac, in a platform specific programming language like C++. The developer compiles the program and distributes it to the user. The user installs and runs the application. So far so good. If the user wants to use the application on more than one computer, he needs to install (and maybe licence) the application on all computers he wishes to use. Every time the application is updated, the user needs to install the update (on all computers). Now translate that into an enterprise with tens of thousands of employees and the ongoing maintenance of such applications becomes a real challenge.

In the age of Web 2.0, this process is very different: The developer writes the application in a platform independent manner. This means a) independent from the platform the application resides on (the server) and more important b) independent from the platform the application is used on (the PC of the user). For a) a combination of open source server platforms and programming languages has become very popular, called LAMP (Linux, Apache, MySQL, PHP/Perl/Python). For b) the best way to achieve platform independence is to write

¹ <http://www.zimbra.com/>

applications that run on the server and represent their interface and results in a browser using standard HTML or XHTML plus AJAX. This kind of application needs no active distribution or installation – the user just uses it through his browser, and, if it is commercial software, subscribes to its service for a period of time. Plus: the user always works with the latest version of the software – no needs for updates or bug fixing. In addition, data migrations when upgrading applications have to be managed by the service provider and are therefore usually painless to the end user.

This new application model of Web 2.0 has also changed the way the applications are developed: Instead of big, monolithic colossi of compiled applications with millions of lines of code, most Web 2.0 programs consist of a number of rather lean modules. Changes, extensions and updates of these programs are easier and faster, leading from a “one major release a year” scheme (Bugwriter 97, 98, 2000, XT, etc.) to a constant cycle of innovation with monthly or often even weekly new features.

PHP - Programming Language of Choice for Web 2.0

For programming Web 2.0 applications, PHP is certainly the number one language. About 20 million sites use PHP², among them some of the biggest names in the Web 2.0 sphere like Yahoo!, Flickr, Facebook, Friendster, Technorati, Zillow.com or Tagged.com. While there are other languages in the competition for web applications, like ASP, Perl, Python, Ruby to name a few, PHP leads the field here and has a market share of more than 30 percent³.

The renowned analyst Forrester Research evaluated 13 leading open source software projects across approximately 40 criteria and found that six of the projects stand out as examples of excellence and are ready for corporate use⁴. PHP was the only dynamic programming language in this group and stood beside MySQL, Eclipse, Apache HTTP Server, Apache Tomcat, and the JBoss Application Server. As the US magazine eWeek attested, the LAMP-stack (and PHP in particular) delivers excellent performance compared to commercial alternatives such as Microsoft's .Net: “This stack's performance numbers suggest what many who have been using PHP for some time now (including some of the busiest blogs on the Web) know to be true – that a pure LAMP-based PHP system can easily handle enterprise-class traffic and loads.”⁵

But what makes PHP so attractive for Web 2.0? It's the same features that make Web 2.0 itself so attractive: It's easy to learn, easy to use, lightweight while offering full functionality and it can easily be extended. Modules and libraries make PHP work with a large number of APIs and other Web 2.0 technologies. For example, there are a large number of PHP toolkits that support creating AJAX-based Web applications.

Another factor for the continuing success of PHP is the professional environment companies like Zend have created for the adoption of PHP: The Zend Core stack for example is specifically designed to create applications for IBM or Oracle databases⁶, and the open source Zend Framework will make PHP application development even easier. In addition, Zend is offering commercial development and management solutions for enterprise usage. This hybrid pattern of open source, cost free and commercial offerings is typical for Web 2.0 and supports rapid adoption and integration on the one hand and sustainable and manageable business models on the other hand.

What Web 2.0 Means for the Enterprise

When IT managers look at Web 2.0, it often appears to them as consumer hype. Will this hype last? Will it have an impact on their business? Will it change the fundamentals of their traditional enterprise applications? Yes, yes and definitely.

² Source: PHP.net, <http://www.php.net/usage.php>

³ Source: Nexen.net http://www.nexen.net/chiffres_cles/phpversion/php_statistics_for_june_2006.php#sglobal

⁴ Source: “The Forrester Wave Project Summary, Q2 2006”.
<http://www.forrester.com/Research/Document/Excerpt/0,7211,36338,00.html>

⁵ Source: <http://www.eweek.com/article2/0,1895,1983364,00.asp>

⁶ http://www.zend.com/products/zend_core

But how? One explanation is the change in how technology penetration and acceptance work. In the past, most new IT technologies trickled down from the enterprise to home users — think of the PC, word processing or PDAs. This has changed in the past few years: Companies are faced with gadgets and technologies their employees bring to work: USB sticks, instant messaging or web based applications. Now people start using Web 2.0 tools within companies: They set up Wikis to document the progress on projects. They use internal and external blogs or podcasts to present ideas or to deliver training. They use eBay to sell unused office furniture. In short: Whenever an enterprise is facing a new gap in their existing applications, it is very likely that a Web 2.0 technology, application or service is used.

Another factor that increases the importance of understanding Web 2.0 for business is the change in business applications themselves. Wikis, podcasts or blogs might be useful tools, but they are not considered mission critical applications. But CRM systems for example, are mission critical. One of the fastest growing CRM vendors, Salesforce.com, leverages heavily on basic Web 2.0 principles: The software is completely web based and licensed per user for periods of time. In addition, Salesforce.com has made APIs (application programming interfaces) public and enabled third party vendors (or the customer's IT staff) to write extensions and enhancements to suit the program to individual needs. An example is an applet that shows a customer's location on a map by interfacing with Google Maps. Not only does this eco-system of solution providers around Salesforce.com provide more value to its customers but it also enables its customers to leverage the open-standards based Web services to easily integrate into their business environment, e.g. integrating CRM data with their financial systems.

What Salesforce.com and other companies do, is called ASP (application service providing) or software on demand. This principle has been around in IT for a while, but the breakthrough comes now, when the technologies are mature and bandwidth is no longer an issue. For customers, it offers not just lower prices, but eliminates the need for application administration. In addition, they need less time for training and they don't have to worry about upgrading their employees PCs when the next version of the program comes out.

But Web 2.0 means more to business than just using web based software. Two basic principles of Web 2.0 are the use of open standards and its ability to expose the functionality of other applications. That helps to exchange data between applications or to make one application call another program to service it, like using Google Maps from CRM software. In the future, this will lead to the use of smaller, more specialised applications. Whenever a company needs a certain function, it can buy or rent it as a service and integrate it in their web based applications.

The Technology Drivers of Web 2.0

In previous times, the major trends in the IT industry were led and set by a small number of big players — like IBM or Microsoft. With Web 2.0, this has changed: Now the impetus comes from start-up companies, open source projects or corporate think tanks and is either standardized through various councils, initiatives and foundations or other times they just become de-facto standards without going through a heavy standardization process. In terms of technology development, this means that Web 2.0 was not there over night; instead it evolved (and continues to evolve) out of dozens of parallel technology and solution developments like XML, SOAP, PHP, MySQL, Linux, Apache, etc. Most of them were not built with the intention to create something like Web 2.0, instead most of them were created to perform a certain task or to fulfill a certain need in the market: Linus Torvalds needed a UNIX alternative — thus Linux was born. The Linux community needed a web server — Apache was born. Web developers needed an easy and performing scripting language that works with Apache — PHP was born which has grown into a full featured programming language within the last decade. That said, it's important to note that LAMP is not the only incarnation of the Web 2.0 PHP stack. As previously mentioned PHP is cross-platform and has tight integration with other open-source and commercial operating systems, Web servers, and databases. Therefore many PHP users are mixing and matching various open-source and proprietary technologies, these incarnations include OLAP (Oracle, Linux, Apache, PHP), WAMP (Windows, Apache, MySQL, PHP) especially for development and other technologies, especially from the big vendors such as IBM, Oracle, Sun, and Microsoft.

For businesses that want to stay up to date in terms of trends and technologies, this means that they have to listen to new sources: Emerging companies like MySQL (Databases), SugarCRM (PHP-based CRM) or Zend (PHP programming) are the main technology stakeholders now. This isn't an either/or choice between emerging and

established vendors, but it's important to note that a big bulk of the innovation is now being driven by the respective domain experts in these fields. Those are often emerging vendors, who excel in understanding how these technologies are changing the Web paradigm.

This becomes increasingly important as Web 2.0 is being adopted by more and more companies for creating their own set of applications. The ability to mash up (combine) existing applications and services and exposing them in an easy to deploy and manage fashion makes an extremely strong value proposition. This is the reason why decision makers should keep an eye on these technologies and the various aspects of Web 2.0, as they can have significant value to their businesses.

Last, it is also critical for IT managers to recognize the immense value in the cultural and social aspects of these new applications. By enabling our users to not only be end users but also be participants in these applications, the ability to use their feedback as a knowledge multiplier inside the Enterprise is immense. Whether it's sharing, tagging or ranking such information, the problem of solving the ultimate knowledge sharing problem in the Enterprise is finally being dealt with more efficiently.

Contact Information

United States and Canada

Zend Technologies, Inc.
19200 Stevens Creek Blvd., Suite 100
Cupertino, CA 95014

Tel.: 1-888-PHP-ZEND (1-888-747-9363)
Fax: 1-408-253-8801

International Operations

Zend Technologies, Ltd.
7 Abba Hillel Street
Silver Building
Ramat Gan, Israel 52136

Tel.: +972-3-613-9665
Fax: +972-3-613-9671

Europe

Zend Technologies GmbH
(Germany, Switzerland and Austria)
Bayerstraße 83
80335 Munich, Germany

Tel.: +49-89-516199-0
Fax: +49-89-516199-20

Zend Technologies S.a.r.l.
5 Rue de Rome - ZAC Les Algorithmes
93110 Rosny Sous Bois, France

Tel.: +33-1-48550200
Fax: +33-1-48123132

Zend Technologies UK
50 Basing Hill
London NW11 8TH, United Kingdom

Tel.: +44 20 8458 8550
Fax: +44 20 8458 8550

Zend Technologies Italy
Largo Richini 6
20122 Milano, Italy

Tel.: +39 02 5821 5832
Fax: +39 02 5821 5400