Peer-to-Peer Sharing of Web Applications

Roberto J. Bayardo Jr.[*], Adina Crainiceanu[+], Rakesh Agrawal[*]

[*]IBM Almaden Research Center
San Jose, CA 95120
bayardo@alum.mit.edu, ragrawal@acm.org

[+]Cornell University
Department of Computer Science
adina@cs.cornell.edu

ABSTRACT

We describe a novel plugin architecture for securely extending personal webserver functionality and supporting development and deployment of meta p2p networks.

1. INTRODUCTION

Web applications are typically professionally administered, stand-alone programs whose operations are confined to the server on which they are installed. In this paper, we describe a system whereby web applications can spread virally through site independent deployment and peer-to-peer distribution. In addition to simplifying the task of extending a (personal) webserver with new applications, we argue such an architecture enables web applications to easily cooperate to form meta p2p networks.

This work is implemented as an extension to the YouServ p2p web hosting system [2], which is currently deployed within the IBM intranet where it is in active use by almost 1500 unique users. YouServ endows each peer with an easy to use webserver supporting basic HTTP commands for serving designated files. Atop this basic HTTP server engine, YouServ provides multiple features that make webserving useful to a much wider community.

The architecture we propose here keeps with the philosophy of bringing web hosting to the masses by simplifying the creation of sites that provide useful applications through dynamic content generation. This architecture allows anyone to write modules (in Java) that intercept and respond to specific HTTP requests. Viewed as such, these YouServ *plugins* are on the surface quite similar to application server scripts (JSP, ASP, CGI, PHP -- see [3]). There are, however, several notable differences, including the following:

- Plugins are all in one, *site-independent* applications that require only simple web-based configuration (if any) in order to work on any node. In contrast, almost all web applications built with, say, JSP or CGI, involve complex dependencies and configuration that can only be navigated by knowledgeable web administrators.
- Plugins installed on one site are, by default, available for download and installation by other users. As such, instances of a popular plugin can easily propagate across the network without the need for a centralized distribution point. Through digital signatures and sandboxing, users need not be expected to understand the code or trust the site offering the plugin for download.
- Plugins have access to a novel API supporting the development of *secure* applications whose instances can cooperate across multiple nodes. In general, security is a first-class function of the architecture, not a feature left to the application developer.

Though some aspects are still under development, most of the architecture described here is fully deployed and already used to support critical aspects of the YouServ system. The YouServ "portal" site is in fact itself implemented atop this architecture. This website receives thousands of hits per day, many for dynamic content. For example, the site must handle single sign-on authentications for the entire network, provide a dynamic online sites list and up-to-date usage statistics, and support all other portal related functions.

The reader may refer to an extended report [1] on this architecture for several screen shots and additional details.

WWW 2003, May 20-24, 2003, Budapest, Hungary.
ACM xxx.

2. RELATED WORK

There are quite a few technologies already available for creating dynamic web content [3]. The goal of the architecture described in this paper is not to develop yet another web scripting language. In fact, we expect to offer existing scripting methods such as JSP as components. The focus of our work is instead on features that support easy and secure web application deployment and distribution, in addition to p2p execution and interaction. With regard to ease of deployment, the closest related work of which we are aware is the Zope application server's facility for packaging web applications into what are known as products [4]. Zope products facilitate the deployment of web applications to other Zope installations. Zope products, however, are still server-centric in that there is no support for p2p distribution, and once installed, they are not intended to cooperate across installations. Additionally, products offer no facilities to deal with the potential of malicious code that could be embedded within them.

Systems such as JXTA [5] also provide infrastructure for simplifying development of peer-to-peer applications. Our architecture focuses on not just development of such applications, but also their distribution and deployment. This focus is important because the most difficult part of creating a successful p2p application is achieving a critical mass of users. We believe our architecture successfully addresses the important user and security issues that are ignored by application frameworks alone.

YouServ plugins should not be confused with Java Applets, which are client-side applications that can be downloaded and dynamically invoked by Java-enabled web browsers. YouServ plugins can be viewed, in some sense, as "server side analogs" to applets. Even though the term servlet might imply such a designation, a servlet is a lower-level script rather than a complete application like an applet or a YouServ plugin.

3. DEPLOYMENT AND DISTRIBUTION

A plugins is easily deployed by a site owner simply by copying the plugin file to the designated plugin directory, and activating the plugin through a special web form provided by each site called the *plugin manager.* The plugin manager also provides links directly to the plugin-specific configuration pages, should the plugin require any. The plugin manager also allows other users to download the plugins for their own use. Users who visit a site containing a plugin they find useful can immediately download it to deploy on their own site, unless the site owner has explicitly forbidden its download (for example, to protect proprietary code). This allows useful plugins to spread virally, with no need for a central distribution point.

4. EXAMPLE PLUGINS

We have created several plugins for the purpose of demonstrating the various API features in order to encourage development of novel functions by other technical members of the YouServ

community. The MP3PlayList plugin is one such "proof of concept" that assembles URLs to all music (MP3) files shared on the site, and provides a convenient interface for generating tailored playlists (.m3u files) from them. Playlists, when served by a website, cause the browser to launch the default music player to stream the music listed within the playlist. Thus, by visiting the MP3PlayList home page and clicking the submit button, a user can quickly sample the music available on the site. Keywords can optionally be specified to customize the musical selection. Each plugin can provide special configuration pages that are accessible only to the authenticated site owner. The MP3PlayList plugin allows configuration of specific web-accessible folders that are to be mined for audio files. It also allows listing specific users that can access the playlist generator to prevent copyright infringement.

Another basic but useful plugin we provide is the SimpleContentEditor, which allows the site owner to (remotely) edit site content using any web browser.

Plugins such as MP3PlayList and the SimpleContentEditor are *stand-alone* in that their instances operate completely independently. *Cooperative* plugins are a much more interesting class which enable meta p2p applications. For example, though not deployed due to copyright concerns, we have extended the MP3PlayList plugin so that its multiple instances aggregate a global database of publicly accessible music files. The playlist generation function then assembles playlists by selecting from this master list instead of the local song list, in effect providing a peer-to-peer song streaming service.

We are currently prototyping another cooperative plugin that will provide distributed content caching to allow low-bandwidth sites to increase their effective bandwidth. This would be similar to how today's heavily accessed sites contract with a service such as Akamai, only without the expense. A site installing such a plugin would perform demand-based caching and redirecting for other sites, forming a dynamic content caching network coordinated by a designated server and domain name system.

5. SECURITY

Plugins, like other applications installed by a user, involve running code that has the potential of causing damage. Plugins in our implementation currently have the full power of the Java virtual machine at their disposal, thus malicious plugins could wreak havoc on the host machine by deleting files, installing back doors, and so on. This section overviews the features implemented to minimize risks due to malicious plugins: digital signatures and code sandboxing.

Recall that our architecture allows plugins to be downloaded from any of a number of participating sites that are not necessarily trusted by the downloader. This feature, key to successful propagation of the most useful applications, also raises the possibility of malicious plugins being offered for download, or reputable plugins being tampered with before distribution. To eliminate this possibility, the system supports plugin signing and verification. Typically code will be signed by the creator, but users who dissect the plugin implementations could also sign the plugin as an endorsement of the code.

Every authenticated peer is automatically assigned an X.509 security certificate by the YouServ certificate authority. The original purpose of the certificate authority was to provide each site with immediate support of HTTPS/SSL for secure site access. The plugin architecture reuses this certificate for the purpose of plugin signing and verification through a utility program that allows a developer to sign a plugin JAR file in a single step.

When a plugin is first detected, its signatures are verified, and those signatures corresponding to unrevoked, YouServ-issued certificates are displayed by the plugin manager. The user can use this information to help decide whether to activate the plugin (thereby invoking its code).

Another security technique that is complementary to digital signatures is code sandboxing. Sandboxing permits specified code, such as an untrusted plugin, to use only a limited set of operations to prevent or limit any damage should the code turn out to be malicious. The YouServ plugin sandbox is being implemented by leveraging the Java 2 sandbox features. The current sandbox design involves multiple permissions that (untrusted) plugins can be granted by the user with reasonably easy to understand consequences. These permissions and their consequences are detailed through a running example in the extended technical report [1].

6. PLUGIN DEVELOPMENT

YouServ plugins are Java programs and associated resources that are encapsulated in the standard Java JAR package format. Along with a servlet-like interface for intercepting certain HTTP requests, plugins have access to special APIs that provide access to application state useful for enabling site independence, application security, and integration with other aspects of YouServ (for example, for maintaining a consistent look and feel).

One important class of functions provided by the API allows the plugin to conveniently exploit YouServ's security features such as single sign-on authentication, user group membership verification, and HTTPS/SSL enabled connections. Through the API, plugin developers are free from having to write any code dealing with user-authentication, a common source of security holes. Similarly, not all sites support HTTPS due to firewalls or other issues, which could complicate building a site-independent plugin. The API offers methods for constructing URLs and obtaining user identity as securely as possible given the capabilities of the hosting site. This frees developers from having to precisely specify when, where, and under what circumstances to use encrypted versus non-encrypted connections.

Another important class of functions allows plugins to work properly on any site without the need for excessive end-user configuration. These functions allow the plugin to query the set of all (private or public) files made web-accessible by the current site, and obtain their URLs or file handles. They also allow the plugin to query the site owner's user ID, the site's domain, the location of the shared web folder, the name assigned to the plugin, and so on.

7. CONCLUSIONS

Plugins allow easy and secure extension of personal webserver capabilities. In addition, by addressing user and security issues, and providing site-independent development and deployment tools, our architecture allows plugins to propagate and cooperate to form meta p2p networks. This addresses what is perhaps the most difficult part of successful p2p application development: achieving a critical mass of users.

8. REFERENCES

[1] R. J. Bayardo Jr., A. Costea, and R. Agrawal. Peer-to-Peer Sharing of Web Applications. *IBM Research Report RJ 10268*, Nov. 2002.

[2] R. J. Bayardo Jr., A. Somani, D. Gruhl, and R. Agrawal. YouServ: A Web Hosting and Content Sharing Tool for the Masses. In *WWW-2002*.

[3] Fieler, J.: *Application Servers: Powering the Web-Based Enterprise*. Morgan Kaufmann Publishers, December 17, 1999.

[4] M. Pelletier, A. Latteier: *The Zope Book*, New Riders Publishing, 2001.

[5] Sun Microsystems, Inc., *Project JXTA: An Open, Innovative Collaboration*. `http://www.jxta.org/project/www/docs/OpenInnovative.pdf`, April 25, 2001.