Studies in Software as a Service

CONNECTIONS

You can find a web service that does almost anything your home computer can do. This month we study some examples of Software as a Service. **BY JOE CASAD**

Years ago, it was common for an application to run on a central system with a ring of users operating through terminals or terminal-style network connections. This vision lives on today in the form of web-based applications providing software as a service. The web-application paradigm means no one has to burn the software onto a CD, package it in a colorful box, and ship it to a retail shop. In fact, you don't even need to provide a download link or offer a help line for users who can't get through the installation. All the end user needs to do is start up a browser and click on a link.

This month we study the world of Software as a Service (SaaS). We examine some examples of the software services available to Linux users, and we look at the surrounding tools and technologies of the SaaS environment.

We introduce you to the openSUSE Build service, an online service for building binary packages from source code. Next, we look at the Prado PHP-development environment – a powerful framework for building your own web-based applications. Also, we examine the problem of adding an online credit card payment service to a website, and we finish with a look at some tools that let you integrate Google SaaS applications into the ordinary desktop environment.

Why SaaS

Several factors have contributed to the return of centralized computing embodied in the Software as a Service model. One factor is the abundance of inexpensive processing power and data storage. The birth of the data center has led to huge advantages of scale. Also important is the growth of fast Internet connections through broadband and other technologies. The rise of the web browser as a standard interface also has helped to popularize SaaS by creating a climate where you really can connect to a well-designed service from almost anywhere.

Another factor driving the rise of the SaaS model is the Internet economy itself. Almost as soon as the Internet was born, people were trying to figure out how to make money with it. After many missteps, bankruptcies, bursting bubbles, and crashing markets, a leading methodology emerged: sell advertisements.

The commercial web is expanding again, this time around the well-proven premise that Internet traffic translates into advertising dollars. Some companies host sophisticated Internet services just for the privilege of selling ads. Sometimes companies don't even want outside advertisers but operate the site as a tool for their own branding and customer outreach. Even if these companies don't receive direct revenue from the site, the result is the same because their Internet presence saves the money they would have needed to advertise in other media.

The SaaS model also lets new companies leap-frog over older competitors and reach the customer directly – without having to worry about sales channels and

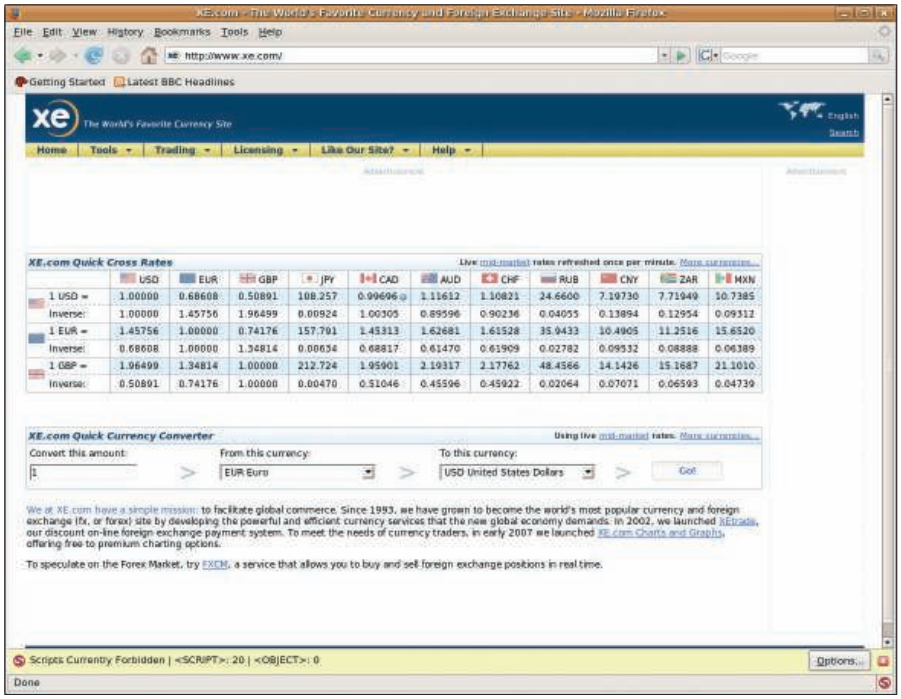

Figure 1: The web is full of handy online tools that deliver free services just to let you view a logo or click on an ad.

brand identity. When Google rolled out the free Google Docs office toolset, for instance, they placed themselves in competition with Microsoft's Office Suite, but they issued the challenge from their own home turf of the ad-driven web, rather fighting the battle on a local desktop dominated by Microsoft.

The Fading Desktop?

Many observers have wondered where this evolution leaves the once-coveted desktop, which was ground zero for its own round of turf battles only a few years ago. Experts have been saying for years that the sunrise of the browser would be a sunset for the local desktop, but the two technologies seem to have settled into peaceful co-existence.

Figure 2: The abundance of cheap storage has changed the way we think about the web. The famous Flickr site will even store your home photos for you. Note the remarkable usage statistics in the center band.

Although a dizzying array of services are available in the browser window, most users aren't ready to give up their desktop applications. Even as users experiment with the best of the new service technologies, there simply are not enough reasons to give up the best of what they already have. Our article on "Integrating Google Tools" shows how you can merge theses online tools with ordinary desktop applications.

Of course, the Software as a Service concept scales to small networks as well. Efficient web-service applications are often implemented at the local network level to consolidate data, centralize processing, and reduce overhead. Our article on the Prado Framework shows how to build a small-scale web applications in PHP.

What About the GPL?

One complication of Software as a Service model is that it departs from the traditional free-software distribution cycle. The GPL and other free-software licenses focus on the *distribution* of software, not its use. Because the program runs on the server and is not actually distributed, the owner of the service can make changes to the program without distributing the source code. Even if the application is completely licensed under the GPL, the remote user is still in the exact same position as a user of proprietary software.

An alternative free software license called the Affero General Public License (AGPL) [1] addresses the problem of running open source software on the server side. The AGPL is nothing more than the GPL with a single clause added. According to Affero's version of the license:

> *2.d) If the Program as you received it is intended to interact with users through a computer network and if, in the version you received, any user interacting with the Program was given the opportunity to request transmission to that user of the Program's complete source code, you must not remove that facility from your modified version of the Program or work based on the Program, and must offer an equivalent opportunity for all users interacting with your Program through a computer network to request immediate transmission by*

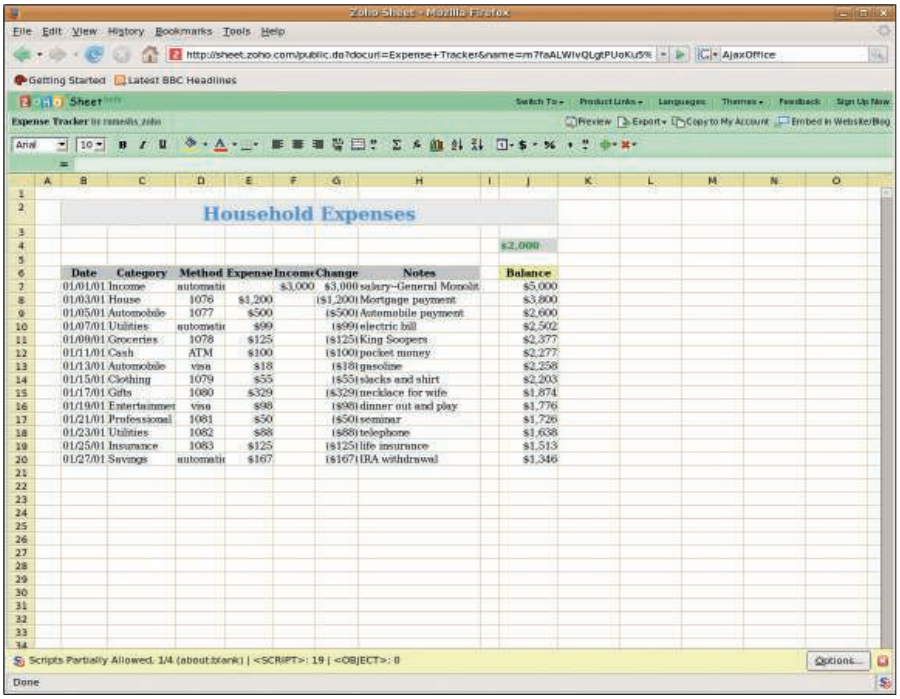

Figure 3: Several companies offer office applications on the web.

HTTP of the complete source code of your modified version or other derivative work.

This clause specifies that whoever implements AGPL-licensed software as a network service must retain a function that allows the user to request delivery of the application source code via the HTTP protocol at any time (if this function is already present in the source code).

The Free Software Foundation (FSF) initially wanted to close the loophole for network-based applications in GPLv3, but it subsequently decided to retain the status quo. Instead the FSF put out their own version of the AGPL [2]. The latest FSF version is based on GPLv3. Section 13 of the GNU APGL corresponds to section 2d in Affero's version:

Notwithstanding any other provision of this License, if you modify the Program, your modified version must prominently offer all users interacting with it remotely through a computer network (if your version supports such interaction) an opportunity to receive the Corresponding Source of your version by providing access to the Corresponding Source from a network server at no charge, through some standard or customary means of facilitating copying of software.

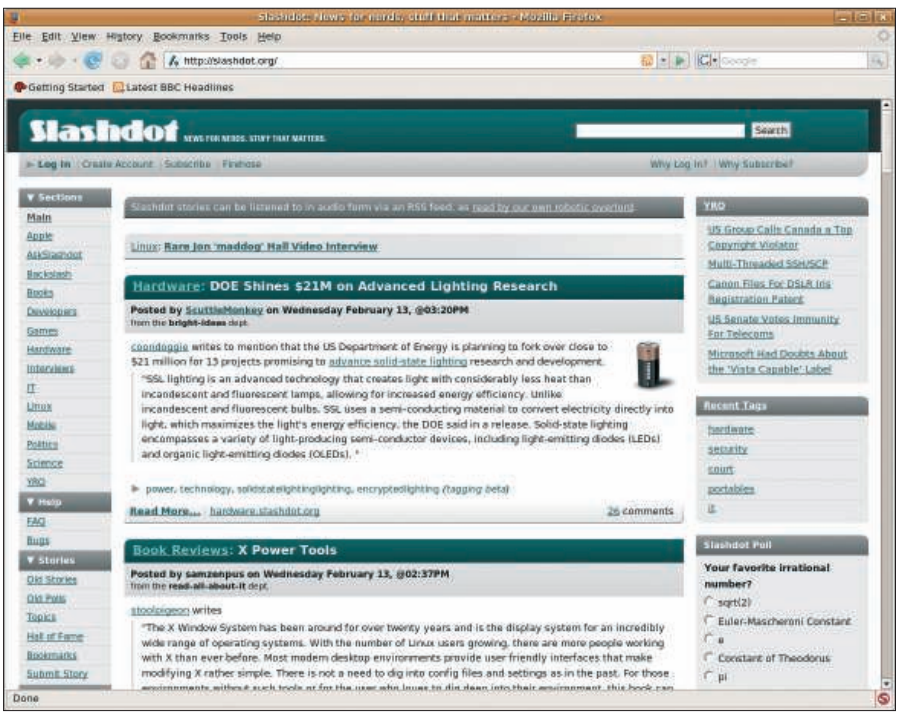

Figure 4: Some online services are happy to share their tools. The blog software used at Slashdot.org, which is known as Slash, is available through SourceForge as an open source project.

If Your Version Supports It

The AGPL essentially states that the user *must not remove* this source code distribution feature if it is built into the program. This provision highlights an important point. The license is chosen by the software developer, not by the user who maintains the software as an Internet service. If the original developer wants users who implement the tool as a software service to distribute the source code, the developer can build this distribution feature into the code and license the project under the APGL. If the developer doesn't want to require distribution for web-based implementations, the the project can remain under the GPL

If the trend for SaaS continues, you can expect to see other licenses that protect the famous "copyleft" protection in the realm of web services. In any case, the developer is always free to voluntarily give the software back to the community regardless of the license. The software that runs Slashdot, for instance, is available for users as an independent open source project (Figure 4).

Conclusion

The fact that even the FSF is reluctant to modify their license to close the web-service loophole is evidence of how the power and momentum of the Software as a Service model. Read on for more on some promising projects from the world of Software as a Service. ■

INFO

[1] Affero General Public License: *http://www.affero.org/oagpl.html*

[2] GNU AGPL: *http://www.fsf.org/licensing/licenses/agpl-3.0.html*