

This article first appeared in the March 2008 issue of *Novell Connection* magazine.

www.novell.com/connectionmagazine

Help the Help

The Next Generation of Hotel Technology

by Carolyn Ford and Jim Sermersheim

Novell.

My favorite hotel is in Seattle, discovered years ago during one of my many business trips. It isn't my favorite because it has gold gilded toilet seats or anything like that; it is my favorite because of their attention to detail, such as Mathew, the front desk guy, makes sure there is bottled water and strawberries—my favorite, waiting in my room for me when I arrive. This personal touch I associate with hotels in general is the reason I found it slightly odd when I first heard of Hotel Technology Next Generation (HTNG), the hotel industry standards group. I found this amusing because, as usual, I was thinking of the hotel business as it revolves around ME. I like to believe that “personal touch” is just for me and applied to me because I am so memorable not because information about me is kept in a cold database somewhere.

My rose-colored glasses recently got a bit of a polishing while attending the HTNG Conference in Seattle. While there, I was surprised to realize hotels have “systems,” not just “Mathews.” And hotels face the same challenges as every other organization. They have, disparate systems, sometimes as many as 100, all with different management interfaces, all required to meet compliance mandates such as PCI and Sarbanes Oxley.

Hotels have systems such as payment, heating and cooling, phones, and that magical channel on the TV that always knows my name. I always envisioned someone going to my room prior to my arrival and manually programming my TV for that extra special touch. After listening to some sessions at the HTNG conference, I think I was not far off, manually programming my TV may be just the tip of the iceberg as far as manual processes go for hotels.

Wouldn't it be cool for my favorite hotel if, when I make a reservation, details for my comfort are set into motion? All my personal favorites transmitted to all those disparate systems, right down to what temperature I like my room and it just magically happens?

Imagine, I make my reservation online, my TV welcomes me by name, with information relevant to my stay, such as upselling the hotel's services and the surrounding area's offerings. The billing system prepares an

On-Ramp to Identity?

What does that mean exactly—on-ramp to identity? Maybe the best way to explain it is to start with the case of Delaware North. They have to solve this problem of connecting all kinds of disparate systems. They need to do it in a way that allows them to provision and manage identities, create workflow policies and produce compliance records.

Well, Delaware North happens to already be a Novell customer, and in fact, what they need to do could be done using Novell Identity Manager by creating some custom connectors.

The real problem is that they need to interact not only with disparate systems in their own data centers, but also with systems owned by partners, subcontractors, customers and possibly peer vendors. Now, as awesome as Novell is, getting all these vendors to purchase a full-blown Novell Identity Manager solution just doesn't seem practical. So how are they going to interact with those systems? The answer lies in the emerging standards coming out of the HTNG group.

By agreeing on groups of standard SOAP messages that perform functions such as “IsUserInRole” or “createUser”, anyone's system can be made to talk a common identity language. By building a reference implementation of some of these messages, the Bandit team has not only proven that you can connect to legacy systems, but has also shown that there can be an easy, open source option for anyone wanting to talk this common identity language without having to commit to an enterprise-scale identity management system.

In this way, the Bandit project provides an easy to access on-ramp to a world where identity is spoken in a common tongue.

estimate of my stay for me. Mathew gets an e-mail alert regarding my arrival date so my room can be prepared with fresh linens that smell like an ocean breeze and a nice Pinot Grigio with strawberries and cheese. Everything is orchestrated, waiting for my arrival. Ahhh, just thinking of it reminds me of my childhood nickname “Queenie.” Of course, now I prefer “Queen Carolyn.”

It may sound a little George Jetson, but system-wide integration with an identity management solution, creating a common administrative interface could make all this happen. Not only that, it would also make it possible to actually monitor systems and ensure everything is within business policy and meeting government regulations. The problem is, many of those systems are just plain old and weren’t engineered to interface with anything. They require high overhead manual administration and have been impossible to keep current with business policy and government mandates. Some of them are simply unreachable.

That is, until now. The Novell Bandit team has proven these systems are NOT unreachable. Using open source, Bandit developers created reference code that builds on existing HTNG standards, extending them with an open message set for identity compliance. This code cost-effectively connects disparate systems to streamline administration, comply with regulatory requirements and, most significantly MY DESIRES!

The code bridges various systems and platforms in an enterprise, including legacy systems, with commercially available identity management software. The first implementation of the code is currently being initiated by Delaware North Companies, a global hospitality and food service provider. In short, the manual processes that have been a royal pain for hotels, and companies in general, can be automated through integration with an identity management solution, making me (OK, all customers) royalty!

Some might argue that “this is only four little pieces of code” and I am making a bigger deal of it than it is. But, allow me to remind you, it was only five little letters Helen Keller finally understood to spell “water”—just one word. One word that opened up a whole world to her, one word gave a, once “unreachable” little girl the power to communicate. Four pieces of code did precisely the same thing for deaf and dumb legacy systems.

I admit: the Helen Keller comparison may be a little dramatic, but you get the idea. The Bandit code makes it possible for these once unreachable systems to integrate with a central identity management solution; that code is the on-ramp to identity! They are the on-ramp to making my hotel stay that much more delightful. I have been focusing on hotels because Delaware North is the first company to use the open source code this way; but the truth is, every company faces similar challenges. They either have legacy or home-grown systems that have to be managed manually that have been considered a burden necessary to bare.

Using open source creates a foundation that is vendor- and to a great extent, system-agnostic. This code can be leveraged by any company with legacy system challenges, by a simple download. (See *On-Ramp to Identity*.) The code Bandit developed for Delaware North probably won’t be an exact fit for your company’s challenges, but the really cool thing about open source is that it has a massive community that is eager to make the code a perfect fit for whatever need you have. And, of course, that need will be to make my experience with your company, hotel, bank, or whatever that much better! I like dark chocolate and pink Champaign. **N**

Getting Your Hands Dirty

So, you want to see just what's going on here? Maybe you're like me and find it much easier to understand something by playing with it than reading about it. If that sounds like you, here's what you can do to get the code and play with it:

1. Check out the code. The code for this project can be viewed at code.bandit-project.org/trac/browser/trunk/HTNG-POC. It can also be checked out using svn from code.bandit-project.org/svn/trunk/HTNG-POC/. If you use the Eclipse IDE, there's already a .project file there to make life a little easier for you.
2. Configure and build. This project produces a war file to be dropped into a Tomcat deployment. Follow the instructions in *readme.build* to create the war file. These instructions will also give you some basic information on the things you can do with the configuration file. (Find that file at ConfigurationFiles/htng-config.xml.)
3. Once you have a war file, drop it into your Tomcat deployment and you should now have a listening server which services some key HTNG SOAP messages.
4. Start the test LDAP server. To test this code, we have mocked up an LDAP server deployment which includes a few identities. This is included in the project, and needs to be started before the tests are run. Do this by running the following command.

```
java -jar test/lib/ldapts.jar
```

5. Send some messages to the SOAP server by running anteaater from the test directory. Follow the instructions in *test/readme* if you need more information regarding how to run them and what they do.

If you run into any problems or have questions, get in touch with a Bandit. Subscribe to the Bandit developer's list at code.bandit-project.org/mailman/listinfo/bandit-dev, or say hi on the Bandit IRC channel at [irc://irc.freenode.net/#bandit-project](http://irc.freenode.net/#bandit-project). We're bound to chime in.