



Print Article



Close Window

From: [www.cio.com](http://www.cio.com)

## Cloud Computing and Patent Trolls: How To Prepare Now

– David Taber, CIO

February 22, 2011

Here's the scenario: Your IT team writes a web service, and part of its WSDL interface includes a [hash algorithm](#) the team came up with on their own. You publish the API and your business partners use your clever little hash in integrating with across cloud services. Years later, you get a letter from a lawyer from a town in Texas you've never heard of, claiming you've infringed on a patent you never heard of. Your team scrambles to replace that hash algorithm, but that means a change to your API and some of your business partners resist making the change. It doesn't matter though: the infringement has already occurred, and you're going to pay somebody quite a bit of money even if you can prove your innocence.

This is going to be difficult, because you're in the domain of patent trolls, legal firms that specialize in very tight arguments and skillful identification of their prey.

Before I go any further, this disclaimer: I am not a lawyer, I don't even play one on TV. This article must not be construed as legal advice, and you need to consult with your attorneys before taking action on the issues raised here. Yes, my lawyer made me write this.

### Patent Mills

All too often, the Patent and Trademark Office grants patents that are overly broad and seem to ignore much of prior art. I used to work for a software company that had been issued a patent that could have covered most of distributed computing.

Recently, I came across a single patent that seemed to cover all the most interesting parts of CRM and ecommerce. The fact that much of what the patent claimed as a new innovation had been bundled in Windows 95 didn't seem to bother the PTO's patent examiner. Yet this patent is now being used to shake down ecommerce companies all across the U.S. The fact that none of them knew about this patent when they built their ecommerce systems doesn't matter: the royalties and the damages probably will be paid. Even if the alleged infringers win their cases, at \$500 an hour the lawyers and expert witnesses get expensive.

That's why large software companies are patent mills. On any given Tuesday, both Microsoft and IBM are probably issued a dozen patents. These patents aren't used as much to protect their innovation as they are to ward off patent trolls — first with patents that may contradict what the trolls have, and second with a mountain of IP that will be used as cross-licensing fodder during the settlement negotiations.

### Avoiding the Hash Problem in the First Place

So what was the problem, anyway? Your team developed the hash algorithm on their own. Sure, but

that doesn't matter: you don't have to copy IP to infringe on a patent. All you have to do is use the same "novel method" that is protected by somebody's patent. Even though you have the ultimate "clean room" implementation — you don't even know about the patent you're infringing — the rent will come due if the patent holder finds out about it. And of course, it's not just hash calculations: lots of ideas and methods are patentable.

With software you're writing entirely for internal use, there's not much risk of somebody external discovering at infringement. But in the cloud, the whole point is to allow your web services to be used by others. You build an API based on a hash algorithm, and then you publish it to the world. Now the trolls can find you. And the more external parties use your clever little hash algorithm, the larger the royalties and damages can become. Lovely.

Of course, hash algorithms and protocols and business process automation mechanisms are commonplace in advanced business software. It's not like you can stop using them just because you're doing work in the cloud. But what you can do is buy, rather than build, these externally-visible components. When you buy them, though, the whole point is to get indemnification from IP infringement. Grabbing something from Sourceforge won't cut it — there's no indemnification unless you've got a contract with a vendor who will stand behind it.

If you can't or don't want to buy these components, then you must have an internal coding standard that requires the developers to document the prior art that they based the clever externally-visible parts of their software on. Whether it's a commercial product or your own internal development, you need to have documentation of when the "original" work was first brought to market (because the prior-art argument only holds if the IP was in public view at the time the patent was filed, not years later when it was granted). If there are textbooks or articles that refer to the techniques your team used, cite them in the footnotes. Although this chore will likely irritate your developers, only they can research the provenance of the IP you use. Once that task is done, the project manager or documentation team can do the actual write-ups and annotations.

Finally, the indemnification and documentation won't do you any good unless they're easily accessible and hard to miss. Filing copies of the artifacts along with the source files in your code control system is one good step. Filing duplicates with your firm's legal department is another. Unfortunately, this is a cost of doing business that won't go away any time soon.

*David Taber is the author of the new Prentice Hall book, "[Salesforce.com Secrets of Success](#)" and is the CEO of [SalesLogistix](#), a certified Salesforce.com consultancy focused on business process improvement through use of CRM systems. SalesLogistix clients are in North America, Europe, Israel, and India, and David has over 25 years experience in high tech, including 10 years at the VP level or above.*

**Follow everything from CIO.com on Twitter [@CIOonline](#).**

© 2010 CXO Media Inc.