

## Chapter 7:

### **Patent war--**

Use low-quality patents to prove that all software rips off your company

Patents in the United States last for 20 years from the time of filing. Prior to 1994, the patent term was 17 years from when the patent was issued.

Bill Gates was originally against the use of patents for software, because of the problems they create for developers in every company, including Microsoft:

“Amazingly we haven't found a way to use our licensing position to avoid having our own customers cause patent problems for us. I know these aren't simple problems but they deserve more effort by both Legal and other groups. For example we need to do a patent exchange with HP as part of our new relationship. In many application categories straightforward thinking ahead allows you to come up with patentable ideas.”

This was from a confidential 1991 memo which was published by a court.

From [http://en.swpat.org/wiki/Bill\\_Gates\\_on\\_software\\_patents](http://en.swpat.org/wiki/Bill_Gates_on_software_patents)

It's important to point out that there is no such thing as “patentable ideas” in the United States. Utility patents can be issued for a process, a machine, a “manufacture” or a composition of matter (for example, if you develop a new non-stick coating for pans.)

A patent cannot be issued for something too abstract-- From an IP standpoint, the worst-case scenario is to have a copyrighted program classified similarly to a recipe or worse, a mathematical equation.

When Intel patents a physical device like a CPU, their IP covers some of the processes and physical designs, the goals of which are to carry out computer instructions. When we write computer instructions, we get a copyright on those instructions-- similarly to how a recipe is copyrighted. The copyright is on the expression or implementation of the recipe (the wording) or the program (the code.) There is no copyright on the process or the concepts implemented.

Thus, if people have access to our code they can implement some of our programs in a slightly different way. To prevent other people from creating a

taskbar, or a search engine, or some other software we want to be the sole providers of, we need to find a way to apply patents to our work. One way to do that is to define a “computer implemented process” (a program) using patent terminology.

The CPU is already designed (and granted a patent) for performing its instructions, which makes for an interesting legal challenge to get a second patent on doing things the CPU is already designed to do-- for example, Toyota may already have a patent on a car, but if nobody has a patent on using a car to deliver pizza as part of a business model, there is a possibility of threatening competing pizza places for infringing on that process.

Another limitation we run into is that patents have to be on things that are novel and non-obvious to people in the field. To get around this and the other requirements, we have our legal team apply for countless patents using the most absurd, drawn-out, vague language possible.

If we drag a competitor to court using such patents and they have a competent legal team to defend them, we will probably lose several of these patents and possibly the case against them. However, if we have a very large company and a very powerful legal team, the competitor may decide (upon receiving threats about such a lawsuit and a request to stop infringing on our very large portfolios containing thousands of vague patents or more) to simply give up and cede to our demands for compliance— whether it is to pay royalties or stop offering a particular feature altogether.

Such patent battles have allowed Apple to harass companies with design patents over such trivial matters as rounded corners-- and to claim they invented a “slide-to-unlock” feature— which has existed on barnyard gates for hundreds of years at least, but never before on the screen of a smartphone! In fact you could say that their “slide-to-unlock” feature is really just the “scrollbar widget” that was probably invented by Xerox in the 1970s, but never before was it used to unlock a screen-- you get the picture.

Several efforts were made to reduce the viability of these strategic patent portfolios, and give legitimate companies a chance to thrive despite our larger businesses and more powerful legal teams. Prior to 2000 it was difficult to address these challenges, but with lobbyists working to reshape Washington to suit our needs above others, we can fight against laws that put smaller companies on more comparable legal footing.

Still, as recently as 2014 Microsoft was still enjoying what amounted to an End-User-License-Agreement attached to the .Net framework-- enforced through an agreement to no sue over patents. Even as software patents lose their potency, the agreements companies were coerced to participate in under these patent-protection schemes remain.

As said early, one possible avenue is to continue to offer protection against

future, hypothetical patent aggression, based on the possibility of overturning Alice Corp. v. CLS Bank International (whether or not that could happen) or based on global enforcement of patents in Europe or Asia.

Another possibility is to simply tie the license for certain high-value proprietary components to similar EULA-esque agreements as Microsoft did in the past. With new investments from Apple, IBM and Microsoft into cloudware, overt proprietary licensing has taken a backseat while covert threats of strategic patent litigation are relied on. So if the patent angle falls apart, a simple return to "Shared Source" or cloudware/freemium-like access to certain enterprise features could form the basis of similar agreements in lieu of patent threats.

### **Relevant quotes from the Halloween documents:**

*"The primary threat Microsoft faces from Linux is against NT Server."*

*"UNIX's perceived Scaleability, Interopability, Availability, and Manageability (SIAM) advantages over NT."*

*"Linux can win as long as services / protocols are commodities"*

*"Linux's homebase is currently commodity network and server infrastructure. By folding extended functionality (e.g. Storage+ in file systems, DAV/POD for networking) into today's commodity services, we raise the bar & change the rules of the game."*

*"Via tools such as enterprise agreements, long term research, executive keynotes, etc., Microsoft is able to commit to a long term vision and create a greater sense of long term order than an OSS process."*

From <https://antitrust.slated.org/halloween/halloween1.html>

*"The effect of patents and copyright in combatting Linux remains to be investigated."*

*"It plants the idea that any MIS manager so foolish as to use Linux will find his operating system yanked out from under him by a future patent lawsuit"*

*"Microsoft truly behaves as though it corporately believes that there's only a fixed pool of key ideas, most already discovered, which software designers must squabble over in zero-sum competition until the end of time. In that game, the only definition of 'winning' is cornering enough goodies to guarantee you a monopoly lock."*

From <https://antitrust.slated.org/halloween/halloween3.html>

*“We need to keep hammering on the difference between source that you can see only after signing a Microsoft NDA or non-competition agreement and source that anyone can examine, modify, and redistribute.”*

*“The risk that Microsoft will go on a patent-lawsuit rampage, designed more to scare potential open-source users than to actually shut down developers, is substantial. The language about “concrete actions” in relation to IPR has the same ominous feel”*

*“Seventy-four percent (74%) of Americans and 82% of Swedes stated that the risk of being sued over Linux patent violations made them feel less favorable towards Linux.”*

From <https://antitrust.slated.org/halloween/halloween7.html>

*“Our SCOsource licensing revenue to date has been generated from license agreements that are non-exclusive, perpetual, royalty-free, paid up licenses to utilize our UNIX source code, including the right to sublicense.”*

*“SCO holds no Unix patents; the state and disposition of the Unix copyrights is unclear and presently disputed between SCO and Novell”*

*“When a company decides to release existing proprietary code as Open Source, the show-stopper is almost always the other parties outside of that company who are involved. Such parties become involved through patents that have been licensed, proprietary code that has been produced by a third party and embedded into the product, and existing contracts relating to the product that have been entered into with customers or other vendors.”*

*“because it's possible to infringe a patent you've never heard of: you can never be sure there isn't some patent somewhere that you're infringing among the millions of patents granted annually.”*

*“SCO has also repeatedly made and withdrawn allegations about patents in the trade press.”*

From <https://antitrust.slated.org/halloween/halloween9.html>

*“Caldera/SCO has a long history of lawsuits over obsolete technologies stripped out of dead companies — starting with DR-DOS from Digital Research and continuing through USL's System V into the present with the IBM lawsuit.”*

From <https://antitrust.slated.org/halloween/halloween10.html>

About the author:

Ted MacReilly is a technologist and tech writer concerned with modern trends in software design and development. He does not work for Microsoft, Apple, or Google, but would like them to continue offering proprietary software and cloudware, without getting too cozy with free software developers.

Copyright (c) 2019 Ted MacReilly

All rights reserved.

Permission to use, copy, modify, and distribute this work for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.