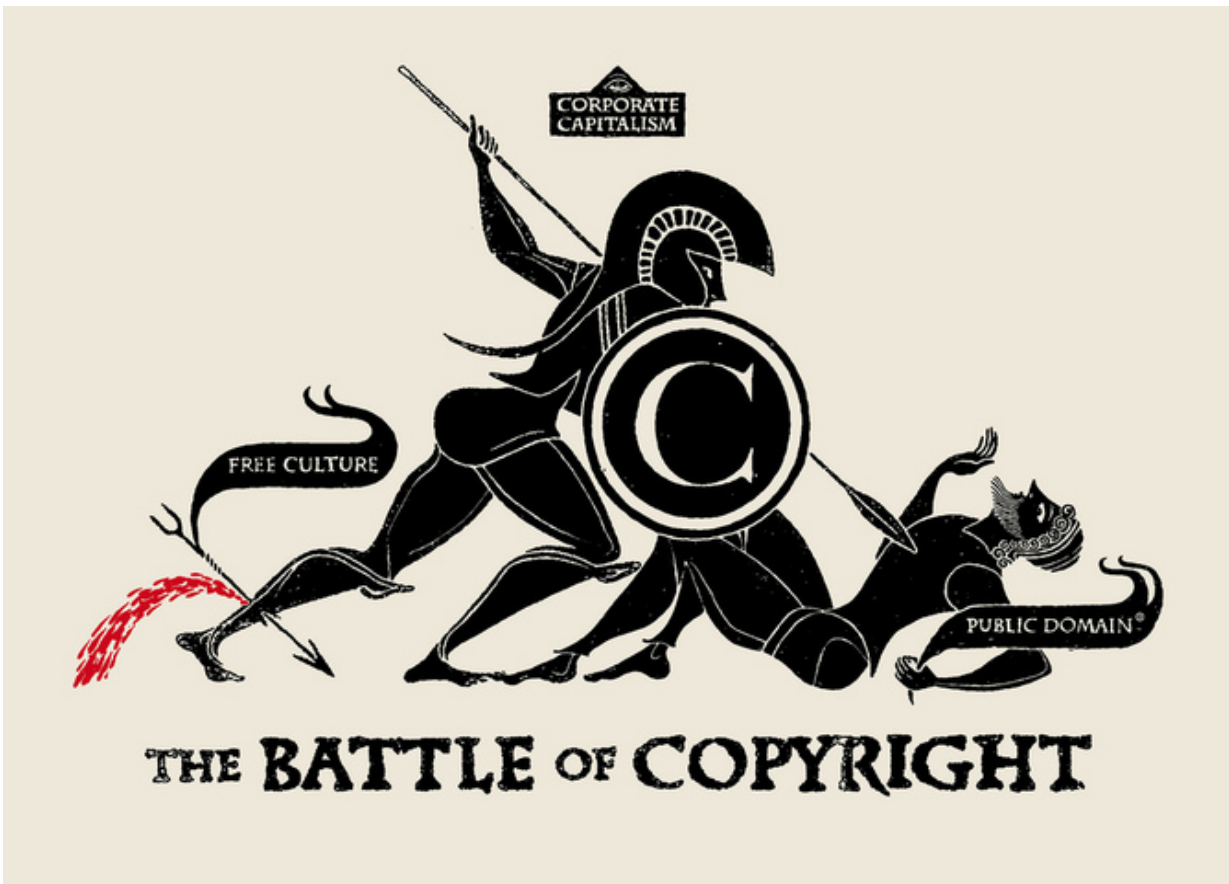# Oracle vs Google case threatens foundations of software design

June 1 2015, by Bill Buchanan

Copyright keeps appearing where it's not wanted. Credit: Christopher Dombres

The Java programming language, which has just turned 20 years old, provides developers with a means to write code that is independent of

the hardware it runs on: "write once, run anywhere".

But, ironically, while Java was intended to make programmers' lives easier, the court case between Oracle, Java's owner, and Google over Google's use of Java as the basis of its Android mobile operating system may make things considerably more difficult.

Google adopted Java for Android apps, using its own, rewritten version of the Java run-time environment (the Java virtual machine or VM) called Dalvik. The Oracle vs Google court case centres around the use of Java in Android, particularly in relation to Application Program Interface (API) calls.

An API is a standard set of interfaces that a developer can use to communicate with a useful piece of code – for example, to exchange input and output, access network connections, graphics hardware, hard disks, and so on. For developers, using an existing API means not having to reinvent the wheel by accessing ready-made code. For those creating APIs, making them publicly and freely accessible encourages developers to use them and create compatible software, which in turn makes it more attractive to end users.

For example, OpenGL and Microsoft's DirectX are two APIs that provide a standardised interface for developers to access 3D graphics hardware, as used in videogames or modelling applications. Hardware manufacturers ensure their hardware is compatible with the API standard, the OpenGL Consortium and Microsoft update their APIs to ensure the latest hardware capabilities are addressed and games developers get a straightforward interface compatible with many different types of hardware, making it easier to create games.

Fight for your right to API

Google designed Android so that Java developers could bring their code to Android by recreating (most of) the standard Java API calls used in the Java libraries and supported by the standard Java VM. The case revolves around whether doing this – by essentially re-creating the Java API rather than officially licensing it from Oracle – is a breach of copyright. If the case finds in favour of Oracle it will set a precedent that APIs are copyrightable, and so make developers lives a lot more legally complex.
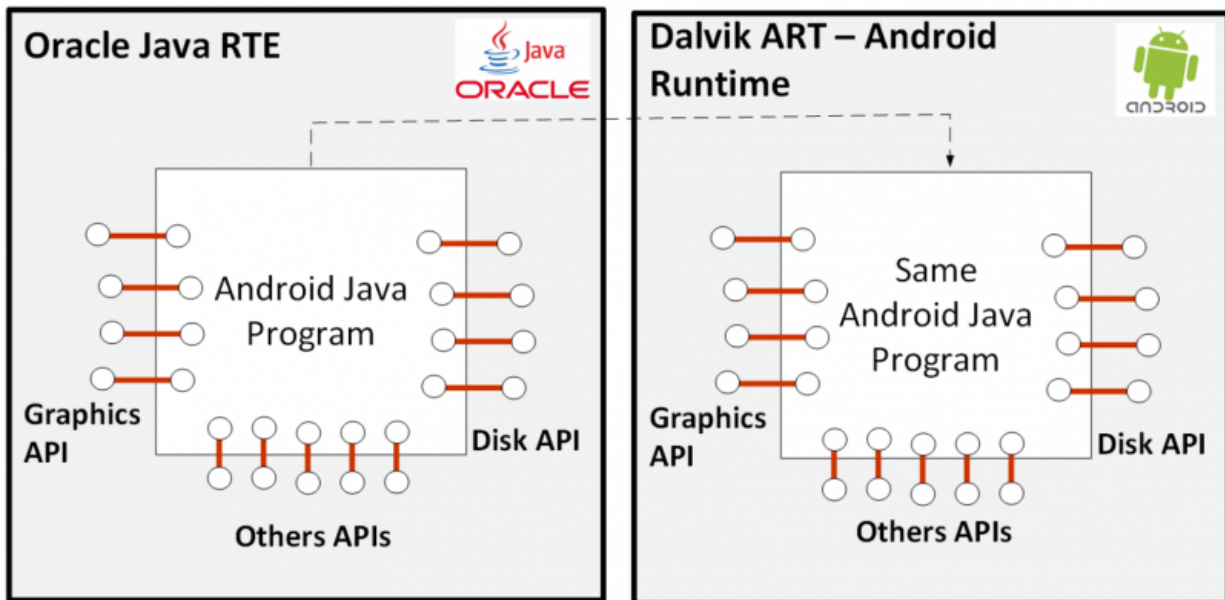
Java runtime and compatible Android equivalent

To be clear, the case doesn't revolve around any claim that Google reused actual code belonging to Oracle, but that the code it produced mimicked what Oracle's Java run-time environment was capable of.

The initial finding came in May 2012, when a US court agreed with

Google's claim that using APIs them falls under fair use, and that Oracle's copyright was not infringed. Then in May 2014, the US Federal Circuit reversed part of the ruling in favour of Oracle, especially related to the issue of copyright of an API. Now, at the US Supreme Court's request, the White House has weighed in in Oracle's favour.

Can you 'own' an API?

For most in the industry, a ruling that it's possible to copyright an API would be a disaster. It would mean that many companies would have to pay extensive licence fees, and even face having to write their own APIs from scratch – even those needed to programmatically achieve only the simplest of things. If companies can prevent others from replicating their APIs through recourse to copyright law, then all third-party developers could be locked out. Also the actual call to the API and its functionality could be copyrighted too, so that the functionality would have to be different too, otherwise it would be a copy.

In the initial trial, District Judge William Alsup taught himself Java to learn the foundation of the language. He decided that to allow the copyrighting of Java's APIs would allow the copyrighting of an improbably broad range of generic (and therefore uncopyrightable) functions, such as interacting with window menus and interface controls. The Obama administration's intervention is to emphasise its belief that the case should be decided on whether Google had a right under fair use to use Oracle's APIs.

It's like the PC all over again

Something like this has happened before. When IBM produced its original PC in 1981 (the IBM 5150), a key aspect was access to the system calls provided by the PC BIOS, which booted the computer and

managed basic hardware such as keyboard, monitor, floppy disk drive and so on. Without access to the BIOS it wasn't possible to create software for the computer.

One firm, Compaq, decided to reverse-engineer the BIOS calls to create its own, compatible version – hence the term "IBM PC compatible" become standard language to describe a program that would run on an IBM model or any of the third-party hardware from other manufacturers that subsequently blossomed. IBM's monopoly on the PC market was opened up, and the PC market exploded into what we see today – would this have happened had IBM been able to copyright its system calls?

So 20 years after the birth of Java, through the groundwork laid by its original creator, Sun Microsystems, Java has become one of the most popular programming languages in the world through being cross-platform and (mostly) open. But now it seems it ends in a trap. The wrong decision in this case could have a massive impact on the industry, where even using a button on a window could require some kind of licence – and licence fees. For software developers, it's a horrible thought. Copyrighting APIs would lock many companies into complex agreements – and lock out many other developers from creating software for certain platforms.

For Google, there's no way of extracting Java from Android now; its runaway success is bringing Google only a whole lot of problems. But as we go about building a world built on software, be assured that one way or another this ruling will have a massive effect on us all.

Source: The Conversation