Oracle v. Google and the Brain Teaser of API Copyright Protection

Flore Brunetti

🕐 MAR 6, 2019

Since the beginning of the digital revolution, copyright law has had to adapt to the advent of new technologies and their impact on the world of creation. Software protection, mainly, raised a lot of issues that still fuel the judicial debate. Indeed, while it is well established that software can be copyrighted as literary works, the extent of the protection keeps on raising questions.

In early cases, courts recognized the copyrightability of original source code and object code, literal elements of the software. The protection was then extended to non-literal elements of computer programs (sequences, structure, organization, user interface…), provided they passed the abstraction-filtration-comparison test which distinguished _process_ from _expression_ in software. But despite these decisions, difficulties remain in the realm of copyright software protection. The main concern of these past few years in the programming community rests on the copyrightability of application programing interfaces (APIs), which can be defined as a "set of core libraries that facilitate the development of applications […] by providing basic system or language functionalities"[1]. APIs give developers a set of ready-to-use functions to avoid them coding existing features several times over, and make programs more compatible with different platforms. As such, they play a crucial role in software development and are necessary to interoperability.

The issue of whether APIs are protected by copyright or not was raised by a lawsuit which arose in 2010 between two tech giants: Oracle Inc., and Google Inc. At that time, Oracle had just bought from Sun Microsystems Inc. the Java Platform, which enables software developers to write programs that are able to run on different types of computers architectures without having to rewrite them for each different type. With its new rights in hand, Oracle decided to sue Google which copied 7,000 lines of declaring code, the identifier that contains the name of each function, and generally replicated the overall structure, sequence, and organization from 37 Java API packages to develop its Android software. It thus reproduced both literal and non-literal elements of Oracle's program. Google's intent in doing so was to allow "programmers to write programs with a uniform way of calling on methods for Java and Android in an effort to promote interoperability and to further the public benefit"[2]. The company, however, wrote its own implementing code, the actual source code that defines the behavior of functions.

This lawsuit gave rise to a series of decisions and the _Oracle v. Google_ saga is not over yet. On May, 31st, 2012, the district court for the Northern District of California gave a first decision, praised by the programming world, in favor of the free use of APIs. Regarding the declaring code, it relied on the merger doctrine and on the short phrases doctrine to hold that it was not copyrightable in this case. Regarding the overall structure, sequence and organization of the Java API packages, the court held that although it was creative, original and resembled a taxonomy, it was "nevertheless a command structure, a system or method of operation – a long hierarchy of over six thousand commands to carry out pre-assigned functions – that is not entitled to copyright protection"[3]. Thus, according to the district court, "the particular elements replicated by Google were free for all to use under the copyright act"[4]. But the case did not stop there, as the Federal Circuit, which did not take into consideration Google's interoperability inquiries, reversed the district court decision and held that "the declaring code and the structure, sequence, and organization of the 37 Java API packages [were] entitled to copyright protection"[5].

According to the Federal Circuit, the district court misapplied the merger doctrine and erred in applying the short phrases doctrine to the declaring code, as the developer could have called the packages any number of things and as Oracle did not try to protect short individual lines of code but 7000 lines of declaring code, combination of elements that could be copyrighted if original. Besides, the Federal Circuit deemed the structure, sequence and organization of the API packages copyrightable as "although an element of a work may be characterized as a method of operation, that element may nevertheless contain expression that is eligible for copyright protection"[6]; the idea is that if a "computer program [was] uncopyrightable simply because it carries out preassigned functions", no computer program would be protectable, which would contradict Congress's express intent[7].

At that stage, Google petitioned for certiorari but the Supreme Court rejected its request. Instead, the case was remanded to the District Court for Northern District of California to determine whether the use made by google of protected APIs was permissible under the doctrine of fair use. On May, 26, 2016, the district court's jury unanimously held in Google's favor. But on appeal, the Federal Circuit, after analyzing the four fair use factors (the purpose and character of the use, the nature of the copyrighted work, the amount and substantiality of the portion used and the effect of the use on the potential market), found that even though a fair use defense could "be sustained in an action involving the copying of computer code", "Google's use of the 37 Java API packages was not fair as a matter of law"[8]. This line of cases, which recognized copyrightability of APIs and led to the conclusion that Google's behavior was not fair use, arguably conflicts with other circuit precedents which favored interoperability. It is widely criticized in the programming industry, as it creates a significant burden for the software development profession and could limit innovation in this area, which would contradict intellectual property protection's aim. Indeed, "reuse of APIs is vital to enabling interoperability among distinct software programs [and] allows software programmers to more rapidly build upon the work of others in developing innovative new software"[9]. As the Electronic Frontier Foundation explains, "the freedom to reimplement and extend existing APIs has been crucial to competition and progress in both hardware and software. Simply put, excluding APIs from copyright has been essential to the development of modern computers and the Internet"[10]. The Federal Circuit's decision in *Oracle v. Google* is thus in contradiction with the "computer industry's long-accepted practice of re-using software interfaces, which provide sets of commands that make it easy to implement common functionality"[11]. It threatens not only the creation of future software but also preexisting programs, as many software developers had, like Google, reused other company's APIs. To try to obtain a decision that would be more protective of innovation in the computer industry, Google just filled, on January 2019, a new petition to try to have the case reviewed by the Supreme Court. Let us hope that the writ of certiorari will be granted to lift the uncertainty that the *Oracle v. Google* saga created over the computer industry.

Flore Brunetti is an LLM candidate, 2019, at NYU School of Law.

---

[1] DARIA VASILESCU-PALERMO, *APIs and copyright protection: the potential impact on software compatibility in the programming industry*, 16 J. Marshall Rev. Intell. Prop. L. 153.,

[2] *Id*.

[3] *Oracle Am., Inc. v. Google Inc.*, 872 F. Supp. 2d 974

[4] Id.

[5] *Oracle America, Inc. v. Google, Inc.* 750 F.3d 1339 (Fed. Cir. 2014)

[6] Id.

[7] Id.

[8] *Oracle Am., Inc. v. Google LLC*, 886 F.3d 1179 (Fed. Cir. 2018)

[9] Pamela Samuelson & Clark D. Asay, *Saving software's fair use future*, 31 Harv. J. Law & Tec 535

[10] Michael Barclay and Mitch Stoltz, *EFF Asks the Federal Circuit to Fix its Terrible API Copyright Decision*, May 30, 2017.

[11] Kent Walker, *Oracle v. Google and the future of software development*