

ALERT

Federal Circuit Patent Bulletin: *CardSoft, LLC v. VeriFone, Inc.*

October 17, 2014

"The person of ordinary skill in the art is 'deemed to read the claim term not only in the context of the particular claim in which the disputed term appears, but in the context of the entire patent,' including the specification and the prosecution history."

On October 17, 2014, in *CardSoft, LLC v. VeriFone, Inc.*, the U.S. Court of Appeals for the Federal Circuit (Prost, Taranto, Hughes*) reversed the district court's judgment entering the jury verdict that the defendants infringed U.S. Patents No. 6,934,945 and No. 7,302,683, which related to software for controlling a payment terminal. The Federal Circuit stated:

Claim terms are generally given their ordinary and customary meaning as understood by a person of ordinary skill in the art. The person of ordinary skill in the art is "deemed to read the claim term not only in the context of the particular claim in which the disputed term appears, but in the context of the entire patent," including the specification and the prosecution history. It can also be appropriate to use extrinsic evidence to determine a term's meaning, but "while extrinsic evidence can shed useful light on the relevant art . . . it is less significant than the intrinsic record in determining the legally operative meaning of claim language."

The district court construed "virtual machine" as "a computer programmed to emulate a hypothetical computer for applications relating to transport of data." That construction is correct, but incomplete. . . . The intrinsic and extrinsic evidence establishes that, at the time the asserted patents were filed, the defining feature of a virtual machine was its ability to run applications that did not depend on any specific underlying operating system or hardware. One problem with the prior art, as the specification notes, was that applications were hardware or operating system dependent. The patent teaches using a virtual machine to solve this problem because a virtual machine "creates a complete portable environment," which "allows programs to operate independent of processor" and allows "[d]ifferent arrangements of hardware [to] be controlled by the same application software."

That the specification would emphasize this aspect of a virtual machine is not surprising in light of the extrinsic evidence. Sun Microsystems, Inc. (Sun) released the famed Java virtual machine in 1996, the year before the earliest possible priority date of the asserted patents. The Java virtual machine acted as an interpreter between a computer application and the computer's underlying operating system and hardware, allowing developers to write one application and run it on multiple different types of computers. Sun marketed Java by emphasizing that the virtual machine allowed a developer to "write once, run anywhere."

And the prosecution history expressly ties this extrinsic evidence—the "write once, run anywhere" Java virtual machine—to the patent's use of "virtual machine." During prosecution of the '945 patent, the applicant stated that the Java virtual machine was a "conventional" virtual machine that allowed "different incompatible computers (incompatible hardware and operating systems)" to "be programmed to emulate the same hypothetical computer" so that "[a]pplications" written for that hypothetical computer "are therefore portable to the previously incompatible computers." The applicant explained that the claims describe "an addition to a conventional virtual machine," not a wholly new structure. In short, the asserted patents use "virtual machine" in exactly the same way Sun used the term—the patents simply optimize the virtual machine for use on a payment terminal.

CardSoft makes two arguments in support of the district court's construction. It first argues that the structure of the claims dictates a broader meaning for "virtual machine" because the claims state that the virtual machine "includes" certain "instructions." CardSoft argues that these instructions are akin to applications, and that because the instructions are "include[d]" in the virtual machine, and the virtual machine can be operating system or hardware dependent, the instructions can also be operating system or hardware dependent. But this conflates the virtual machine itself with applications (or instructions) running on the virtual machine. The defining characteristic of a virtual machine was, and is, that it acts as an interpreter between applications and the underlying hardware or operating system. That the claimed virtual machine "includes" applications, in the sense that it acts as an interpreter for applications, does not mean that the applications can be hardware or operating system dependent. Such a construction would leave "virtual machine" essentially meaningless.

CardSoft next argues that differentiation of independent claim 1 from dependent claims 7 and 8 of the '945 patent mandates a broader construction because these dependent claims state that instructions "do not require translation to the native software code of the microprocessor." But claim differentiation is merely a presumption. It is "a rule of thumb that does not trump the clear import of the specification." Because the ordinary meaning of "virtual machine" is clear in light of the specification and prosecution history, claim differentiation does not change its meaning.

VeriFone contends that, applying the correct construction, it is entitled to judgment of no infringement as a matter of law because the accused payment terminals run applications that depend on a specific underlying operating system or hardware. . . . Arguments that are not appropriately developed in a party's briefing may be deemed waived. By failing to respond to VeriFone's argument in the briefing, CardSoft has effectively conceded that the accused devices run applications that depend on a specific underlying operating system or hardware. Consequently, we find that CardSoft has waived this argument, and we grant Appellants judgment of no infringement as a matter of law.