IN THE UNITED STATES DISTRICT COURT
FOR THE EASTERN DISTRICT OF TEXAS
MARSHALL DIVISION

CARDSOFT (ASSIGNMENT FOR THE BENEFIT OF CREDITORS), LLC	§	
v.	§	CASE NO. 2:13-CV-290-RWS-RSP
FIRST DATA CORP., et al.	§	

SUPPLEMENTAL CLAIM CONSTRUCTION MEMORANDUM AND ORDER

On June 10, 2014, the Court held a hearing to determine the proper construction of the disputed claim terms in United States Patents No. 6,934,945 ("the '945 Patent") and 7,302,683 ("the '683 Patent"). The Court entered a Claim Construction Memorandum and Order on June 24, 2014 ("*First Data Markman*"). Dkt. No. 82.

Previously, the Court construed the term "virtual machine" (among other terms) in the same patents-in-suit in *CardSoft (Assignment for the Benefit of Creditors) LLC, et al. v. VeriFone Systems, Inc., et al.*, No. 2:08-CV-98, Dkt. No. 251 (E.D. Tex. Sept. 23, 2011) ("*VeriFone Markman*"). The *VeriFone* case proceeded to a trial on the merits and a jury verdict of infringement. *See id.*, Dkt. No. 389, 6/8/2012 Verdict Form. The Court entered a Judgment on October 30, 2013. *Id.*, Dkt. No. 483.

After the *First Data Markman*, the Court of Appeals for the Federal Circuit reversed the *VeriFone* judgment, finding error in the construction of "virtual machine." *See CardSoft (Assignment for the Benefit of Creditors) LLC, et al. v. VeriFone Systems, Inc.*, 769 F.3d 1114 (Fed. Cir. 2014) ("*VeriFone Opinion*"), *petition for rehr'g and rehr'g en banc denied* (Dec. 22, 2014), *petition for cert. filed*, No. 14-1160 (Mar. 23, 2015).

After considering the arguments made by the parties in their Joint Statement of Claim Construction (Dkt. No. 140, submitted April 20, 2015), filed pursuant to the Court's March 17, 2015 Order Regarding Parties' Joint Status Report (Dkt. No. 135), the Court issues this Supplemental Claim Construction Memorandum and Order.

THE PARTIES' POSITIONS

"virtual machine"	
Plaintiff's Proposed Construction	**Defendants' Proposed Construction**
"a computer programmed to emulate a hypothetical computer for applications relating to transport of data that processes instructions expressed in a hardware or operating system-independent language"	"a computer programmed to emulate a hypothetical computer for applications relating to transport of data that processes instructions expressed in a hardware and operating system-independent language, allowing different incompatible computers (incompatible hardware and operating systems) to be programmed to emulate the same hypothetical computer so that applications written for that hypothetical computer are therefore portable to the previously incompatible computers. A virtual machine, acting as an interpreter between an application program and a payment terminal's underlying hardware and operating system, has the ability to run applications that do not depend on any specific underlying operating system or hardware because a virtual machine creates a complete portable environment, which allows programs to operate independent of processor and allows different arrangements of hardware to be controlled by the same application software"

Dkt. No. 140 at 2 (citations, internal quotation marks, square brackets, and ellipsis omitted from Defendants' proposed construction).

Plaintiff argues that Defendants "attempt[] to read limitations into the term that are nowhere supported in [the Federal Circuit] decision, but are merely used as background in

arriving at its decision." *Id.* at 9. Plaintiff also submits that "[a]s to the question of whether the proper construction should contain an 'and' or an 'or' in describing the hardware or operating system-independent language, [Plaintiff] asserts that both the Federal Circuit decision and this Court's construction of other terms makes clear that the conjunction should be 'or.'" *Id.* at 10.

Defendants respond that "Defendants' proposed construction quotes verbatim from the appellate opinion." *Id.* at 12. Defendants urge that the required "ability to 'write once, run *anywhere*'" requires that an application "can run on not only hardware X, Y, and Z, but also on operating system A, B, and C—it is independent of both hardware *and* operating system." *Id.* at 13. Defendants argue that "the Federal Circuit's construction is explicitly part of the mandate and may not be revisited," and "[Plaintiff's] litigation-driven construction would lead to the clearly erroneous result whereby the 'virtual machine' would not support 'write once, run anywhere' applications, which the Federal Circuit held is a 'defining feature' of the claimed 'virtual machine.'" *Id.* at 14 & 16.

DISCUSSION

In *VeriFone,* the Court summarized the defendants' arguments as follows:

> . . . Defendants argue that the virtual machine's emulation of the hypothetical computer must somehow overcome incompatibility between both different operating systems and different hardware (processors) that can only understand and process its own specific native code. Defendants contend that the only way that the claimed "virtual machine means" can overcome these incompatibilities is if the virtual machine is programmed and receives instructions in a language that is independent of both the hardware processor and the operating system.

VeriFone Markman at 9. In *VeriFone*, the Court rejected the defendants' arguments and found:

> If both the message processor and the function processor, which are part of the virtual machine, can be implemented in the native software code of the microprocessor, then they do not have to be expressed in "a hardware/operating system-independent language" as Defendants' proposed construction would require.

> . . . [T]he specification . . . criticizes prior art virtual machines for requiring *applications* written in hardware-specific code since such applications would not be portable to different devices. . . . It does not, however, discuss whether the virtual machine itself can be written in hardware-specific code – indeed, the cited portion is silent on the topic of the code used to implement the claimed virtual machine. Likewise, none of the other specification language to which Defendants cite states that the virtual machine, or any part thereof, must necessarily be written in a hardware/operating system independent language in order to be emulatable in different computers.
>
> * * * The portions of the prosecution history cited and relied upon by Defendants . . . do not make any . . . clear disclaimer of virtual machines written in hardware-specific code. * * * [T]he court rejects Defendants' argument that the "virtual machine means" must "process[] instructions in a hardware/operating system-independent language on the communication device."
>
> * * * Accordingly, the court construes "virtual machine means" and "virtual machine" to mean "a computer programmed to emulate a hypothetical computer for applications relating to transport of data."

Id. at 12-13 (square brackets in original).

In the above-captioned case, Defendants proposed that "virtual machine means" means "a computer programmed to emulate a hypothetical computer running applications that are independent of the communication device hardware and operating system." *First Data Markman* at 9. The Court found that "the specification discloses that a virtual machine can facilitate 'portability' of programs," that during prosecution "Plaintiff explained that the claimed invention is different from the well-known 'Java Virtual Machine,'" and that "[n]owhere, however, did the patentee definitively state that all virtual machine applications must be portable or that a virtual machine can run only portable applications." *Id.* at 12-15. "For example," the Court noted, "applications that can be executed on a virtual machine installed on a particular device might not operate, or at least not operate properly, when executed on the same virtual machine on a different device." *Id.* at 15 (citing '945 Patent at 22:21-31). The Court concluded:

> Defendants' proposal that applications must be "independent" of the device hardware and operating system is too narrow. For example, application

> performance and capabilities may vary depending upon the hardware and operating system. Also . . . portability or non-portability of applications is not a limitation of the virtual machine that executes the applications. Instead, portability of applications is merely a desired result of using a virtual machine.
>
> Finally, as Plaintiff has argued, Defendants' concern that Plaintiff's proposed constructions "would encompass even systems requiring absolute one-to-one customization of application to platform" is addressed by the separate claim limitation that "the virtual machine means is emulatable in different computers having incompatible hardwares or operating systems." More specifically, because the parties agree that the "virtual machine means" term requires emulating a "hypothetical computer" for running applications, the "virtual machine means" and "emulatable . . ." limitations together require that applications run on the *same* "hypothetical computer" emulated on *different* computers.

First Data Markman at 16 (citations and internal quotation marks omitted). The Court therefore construed "virtual machine means" to mean "a computer programmed to emulate a hypothetical computer for applications relating to transport of data." *Id.* at 17.

After the *First Data Markman*, the Federal Circuit reversed the judgment in *VeriFone* and found:

> Because the district court's construction does not reflect the ordinary and customary meaning of "virtual machine" as understood by a person of ordinary skill in the art, we reverse.
>
> * * *
>
> 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 district court improperly rejected the Appellants' argument that the "virtual machine" must "process[] instructions expressed in a hardware/operating system-independent language." * * *
>
> The district court's construction improperly conflates the claimed virtual machine with applications written to run on the virtual machine. The claimed virtual machine is operating system or hardware dependent because it must communicate directly with the underlying operating system or hardware. But the applications written to run on the virtual machine are not correspondingly dependent because the applications are written to communicate with the virtual machine, not the actual underlying operating system or hardware.
>
> . . .

> 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. * * *
>
> * * *
>
> [T]he prosecution history expressly ties th[e] extrinsic evidence—the "write once, run anywhere" Java virtual machine—to the patent's use of "virtual machine." * * * 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 [Microsystems, Inc.] used the term [with reference to its Java virtual machine]—the patents simply optimize the virtual machine for use on a payment terminal.
>
> . . .
>
> * * * 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.
>
> * * *
>
> Because the district court erred by failing to give "virtual machine" its ordinary and customary meaning, we reverse the district court's construction of this term.

VeriFone Opinion at 1117-20 (citations omitted).

Plaintiff argues that the *VeriFone Opinion* is consistent with Plaintiff's proposal of "instructions expressed in a hardware *or* operating system-independent language," as noted above.

Although the defendants in *VeriFone*, as well as the Federal Circuit, used the word "or," Defendants have properly countered that for instructions to be *not* dependent on hardware *or* operating system, such instructions must be independent of *both* hardware *and* operating system. The *VeriFone Opinion* is clear in this regard. *See id.* at 1117 ("The district court improperly

rejected the Appellants' argument that the 'virtual machine' must 'process[ ] instructions expressed in a hardware/operating system-independent language.'"); *see also id.* at 1118 ("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"); *id.* ("write once, run anywhere").

Defendants' proposed construction gives effect to the above-quoted findings in the *VeriFone Opinion* but contains far more than is necessary or manageable as a construction for the term "virtual machine."

The Court accordingly hereby construes **"virtual machine"** to mean **"a computer programmed to emulate a hypothetical computer for applications relating to transport of data that processes instructions expressed in a hardware and operating system-independent language."**

CONCLUSION

The Court adopts the construction set forth in this opinion for the disputed term at issue in the patents-in-suit.

The parties are ordered that they may not refer, directly or indirectly, to each other's claim construction positions in the presence of the jury. Likewise, the parties are ordered to refrain from mentioning any portion of this opinion, other than the actual definition adopted by the Court, in the presence of the jury. Any reference to claim construction proceedings is limited to informing the jury of the definitions adopted by the Court.

SIGNED this 29th day of May, 2015.

ROY S. PAYNE
UNITED STATES MAGISTRATE JUDGE