

**United States Court of Appeals
for the Federal Circuit**

CARDSOFT
(assignment for the Benefit of Creditors), LLC,
Plaintiff-Appellee,

v.

**VERIFONE, INC., HYPERCOM CORP.,
and VERIFONE SYSTEMS INC.,**
Defendants-Appellants,

AND

**INGENICO S.A., INGENICO CORP.,
INGENICO INC., AND WAY SYSTEMS, INC.,**
Defendants.

2014-1135

Appeal from the United States District Court for the
Eastern District of Texas in No. 2:08-CV-00098, Magis-
trate Judge Roy S. Payne.

Decided: October 17, 2014

DONALD R. MCPHAIL, Cozen O'Connor, of Washington,
DC, argued for plaintiff-appellee. With him on the brief
were BARRY P. GOLOB and KERRY B. MCTIGUE. Of counsel

on the brief was WILLIAM E. DAVIS, III, Davis Firm P.C., of Longview, Texas.

E. JOSHUA ROSENKRANZ, Orrick, Herrington & Sutcliffe LLP, of New York, New York, argued for defendants-appellants. With him on the brief were MARK S. DAVIES, RICHARD A. BIERSCHBACH, BRIAN D. GINSBERG, and CAM T. PHAN. Of counsel on the brief was ROBERT W. KANTNER, Jones Day, of Dallas, Texas.

Before PROST, *Chief Judge*, TARANTO and HUGHES, *Circuit Judges*.

HUGHES, *Circuit Judge*.

CardSoft alleges that Appellants infringe two patents directed to software for small, specialized computers, like payment terminals. In construing the patent claims, the district court adopted CardSoft's proposed construction for the claim term "virtual machine." Applying the district court's construction, a jury returned a verdict for CardSoft. Because the district court erred in its construction of "virtual machine," and because CardSoft waived any argument that Appellants infringe under the correct construction, we reverse.

I

CardSoft (Assignment for the Benefit of Creditors), LLC (CardSoft) filed suit in March 2008 against Appellants VeriFone, Inc., VeriFone Systems Inc., and Hypercom Corp. (collectively, VeriFone), asserting infringement of U.S. Patent Nos. 6,934,945 (the '945 patent) and 7,302,683 (the '683 patent). The district court held a *Markman* hearing in July 2011 and conducted a jury trial in June 2012. The jury determined that certain VeriFone devices infringed claim 11 of the '945 patent and claim 1 of the '683 patent and that these claims were not invalid. VeriFone moved for a new trial and for judgment as a

matter of law, but the district court denied both motions. VeriFone appeals. We have jurisdiction under 28 U.S.C. § 1295(a)(1).

II

The '683 patent is a continuation of the '945 patent and shares the same specification. Both patents describe software for controlling a payment terminal. *See* '945 patent col. 1 ll. 10–17. Payment terminals are small, specialized computers, and include a processor, peripheral units like a card reader, a display, a printer, or a communications interface, and a software operating system to control the hardware components. *Id.* at col. 2 l. 64–col. 3 l. 1.

According to the patents, prior art payment terminals used a variety of “different hardware/software architectures.” *Id.* at col. 2 ll. 34–37. But this variety of different architectures meant that each application program for a payment terminal needed to be written specifically for that terminal. *Id.* at col. 3 ll. 5–11. “[P]rogramming alterations are not ‘portable’ between different types of devices.” *Id.* at col. 3 ll. 13–14.

To solve this problem, the specification describes a “virtual machine,” acting as an “interpreter” between an application program (like a particular merchant’s payment processing software) and a payment terminal’s underlying hardware and operating system. *Id.* at col. 3 ll. 29–36. Instead of writing a payment processing application for a particular hardware configuration or operating system, a developer can write the application for the virtual machine. *Id.* at col. 3 ll. 41–45. This application can then run on any payment terminal running the virtual machine, creating “a complete portable environment for program operations.” *Id.* at col. 3 ll. 45–46.

The specification acknowledges that the concept of a virtual machine was well known at the time, but argues

that the slowdown in operation created by a conventional virtual machine would create a “performance penalty” that could be a “significant problem” for a payment terminal. *Id.* at col. 3 ll. 35, 47–49. To solve this problem, the specification describes an improved virtual machine optimized for use on specialized portable computers, like payment terminals. This improved virtual machine includes a specialized “virtual message processor” designed to optimize network communications. *Id.* at col. 3 ll. 56–67. It also includes a specialized “virtual function processor” designed to optimize control of the payment terminal itself. *Id.*

Claim 1 of the ’945 patent is representative of the asserted claims:

A communication device which is arranged to process messages for communications, comprising a *virtual machine* means which includes

a virtual function processor and function processor instructions for controlling operation of the device, and

message in[struction] means including a set of descriptions of message data;

a virtual message processor, which is arranged to be called by the function processor and which is arranged to carry out the message handling tasks of assembling the messages, disassembling messages and comparing the messages under the direction of the message instruction means that is arranged to provide directions for operation of the virtual message processor, whereby when a message is required to be handled by the communications device the message processor is called to carry out the message handling task,

wherein the *virtual machine* means is emulatable in different computers having incompatible hardwares or operating systems.

Id. at col. 50 ll. 48–67 (emphases added).

III

VeriFone appeals the district court’s construction of “virtual machine,” found in all asserted claims. It argues that the district court erred by not requiring the claimed “virtual machine” to include the limitation that the applications it runs are not dependent on any specific underlying operating system or hardware. We agree. 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.

A

“Claim construction is a legal statement of the scope of the patent right” that we review de novo. *Lighting Ballast Control LLC v. Philips Elecs. N. Am. Corp.*, 744 F.3d 1272, 1276–77, 1284 (Fed. Cir. 2014) (en banc); *Cybor Corp. v. FAS Techs., Inc.*, 138 F.3d 1448, 1456 (Fed. Cir. 1998) (en banc). Claim terms are generally given their ordinary and customary meaning as understood by a person of ordinary skill in the art. *Phillips v. AWH Corp.*, 415 F.3d 1303, 1312–13 (Fed. Cir. 2005) (en banc). 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. *Id.* at 1313. 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.” *Id.* at 1317 (citations and quotations omitted).

B

The district court construed “virtual machine” as “a computer programmed to emulate a hypothetical computer for applications relating to transport of data.” *Card-Soft, Inc. v. VeriFone Holdings, Inc.*, No. 2:08-cv-98, 2011 WL 4454940, at *8 (E.D. Tex. Sept. 29, 2011). 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.” *Id.* at *7. In doing so, the district court noted that dependent claims 5 and 6 of the ’945 patent expressly require that the “message processor” and “function processor” components of the virtual machine are “implemented in the native software code of the microprocessor in the device.” *Id.* at *7. The district court also noted that the specification does not bar the virtual machine from being “written in hardware specific code.” *Id.* Relying on this, the district court held that the claimed “virtual machine” need not run applications or instructions that are hardware or operating system independent.

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.

1

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. '945 patent col. 3 ll. 5–14, 29–36. 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.” *Id.* at col. 3 ll. 34–46; col. 10 ll. 5–7.

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. *See Oracle Am., Inc. v. Google Inc.*, 750 F.3d 1339, 1348 (Fed. Cir. 2014). 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. *Oracle*, 750 F.3d at 1348; *Nazomi Commc’ns, Inc. v. Nokia Corp.*, 739 F.3d 1339, 1340 (Fed. Cir. 2014); *Nazomi Commc’ns, Inc. v. ARM Holdings, PLC*, 403 F.3d 1364, 1366 (Fed. Cir. 2005). Sun marketed Java by emphasizing that the virtual machine allowed a developer to “write once, run anywhere.” *Oracle*, 750 F.3d at 1348.

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.” JA18849. The applicant explained that the claims describe “an addition to a conventional virtual machine,” not a wholly new structure. *Id.* 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.

2

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.” ’945 patent col. 50 ll. 49–53. 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.” ’945 patent col. 51 ll. 29–31, 36–37. But claim differentiation is merely a presumption. It is “a rule of thumb that does not trump the clear import of the specification.” *Eon-Net LP v. Flagstar Bancorp*, 653 F.3d 1314, 1323 (Fed. Cir. 2011); *see also Marine Polymer Techs., Inc. v. HemCon, Inc.*, 672 F.3d 1350, 1359 (Fed. Cir. 2012) (en banc) (“[C]laim differentiation is not a hard and fast rule and will be overcome by a contrary construction dictated by the written description or prosecution history.”) (citation and quotation omitted). Because the ordinary meaning of “virtual machine” is clear in light of the specification and prosecution history, claim differentiation does not change its meaning.

IV

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. Appellants’ Br. 64–65. CardSoft did not respond to this argument in its responsive brief on appeal. CardSoft recognized the issue: “Appellants argue that, under their construction of ‘*virtual machine*,’ ‘a ruling of noninfringement [sic] is compelled.” Appellee’s Br. 29. But CardSoft never responded. It instead argued that “[b]ecause Appellants’ construction of ‘*virtual machine*’ is wrong” the jury’s verdict should be affirmed. *Id.*

Arguments that are not appropriately developed in a party’s briefing may be deemed waived. *See SmithKline Beecham Corp. v. Apotex Corp.*, 439 F.3d 1312, 1320 (Fed. Cir. 2006) (collecting cases); *see also Procter & Gamble Co. v. Amway Corp.*, 376 F.3d 496, 499 n.1 (5th Cir. 2004) (“Failure adequately to brief an issue on appeal consti-

tutes waiver of that argument.”). 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.

V

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. And because CardSoft waived any argument that Appellants infringe under the correct construction, we grant Appellants judgment of no infringement as a matter of law.

REVERSED