IN THE UNITED STATES DISTRICT COURT
FOR THE DISTRICT OF DELAWARE

EXPRESS MOBILE, INC., Plaintiff, v. GODADDY.COM, LLC, Defendant.	Civil Action No. 19-1937-RGA

MEMORANDUM OPINION

Timothy Devlin, DEVLIN LAW FIRM LLC, Wilmington, DE; James R. Nuttall, Michael Dockterman, Robert F. Kappers, Tron Fu, Katherine H. Johnson, STEPTOE & JOHNSON LLP, Chicago, IL; Christopher A. Suarez, STEPTOE & JOHNSON LLP, Washington, DC, Attorneys for Plaintiff.

Beth Moskow-Schnoll, Brittany Giusini, Brian S.S. Auerbach, BALLARD SPAHR LLP, Wilmington, DE; Brian W. LaCorte, Jonathan A. Talcott, BALLARD SPAHR LLP, Phoenix, AZ, Attorneys for Defendant.

June 1, 2021

/s/ Richard G. Andrews
ANDREWS, UNITED STATES DISTRICT JUDGE:

Before me is the issue of claim construction of multiple terms in U.S. Patent Nos. 6,546,397 (the '397 Patent), 7,594,168 (the '168 Patent), 9,063,755 (the '755 Patent), 9,471,287 (the '287 Patent), and 9,928,044 (the '044 Patent). I have considered the Parties' Joint Claim Construction Brief. (D.I. 64). I held remote oral argument on April 8, 2021. (D.I. 77). The parties argued ten terms there. I asked for supplemental briefing on two of them, which is underway. (D.I. 80). I now decide the other eight terms.

I. LEGAL STANDARD

"It is a bedrock principle of patent law that the claims of a patent define the invention to which the patentee is entitled the right to exclude." *Phillips v. AWH Corp.*, 415 F.3d 1303, 1312 (Fed. Cir. 2005) (en banc) (internal quotation marks omitted). "'[T]here is no magic formula or catechism for conducting claim construction.' Instead, the court is free to attach the appropriate weight to appropriate sources 'in light of the statutes and policies that inform patent law.'" *SoftView LLC v. Apple Inc.*, 2013 WL 4758195, at *1 (D. Del. Sept. 4, 2013) (quoting *Phillips*, 415 F.3d at 1324) (alteration in original). When construing patent claims, a court considers the literal language of the claim, the patent specification, and the prosecution history. *Markman v. Westview Instruments, Inc.*, 52 F.3d 967, 977–80 (Fed. Cir. 1995) (en banc), *aff'd*, 517 U.S. 370 (1996). Of these sources, "the specification is always highly relevant to the claim construction analysis. Usually, it is dispositive; it is the single best guide to the meaning of a disputed term." *Phillips*, 415 F.3d at 1315 (internal quotation marks omitted).

"[T]he words of a claim are generally given their ordinary and customary meaning. . . . [Which is] the meaning that the term would have to a person of ordinary skill in the art in question at the time of the invention, i.e., as of the effective filing date of the patent application."

Id. at 1312–13 (citations and internal quotation marks omitted). "[T]he ordinary meaning of a claim term is its meaning to [an] ordinary artisan after reading the entire patent." *Id.* at 1321 (internal quotation marks omitted). "In some cases, the ordinary meaning of claim language as understood by a person of skill in the art may be readily apparent even to lay judges, and claim construction in such cases involves little more than the application of the widely accepted meaning of commonly understood words." *Id.* at 1314.

When a court relies solely upon the intrinsic evidence—the patent claims, the specification, and the prosecution history—the court's construction is a determination of law. *See Teva Pharm. USA, Inc. v. Sandoz, Inc.*, 574 U.S. 318, 331 (2015). The court may also make factual findings based upon consideration of extrinsic evidence, which "consists of all evidence external to the patent and prosecution history, including expert and inventor testimony, dictionaries, and learned treatises." *Phillips*, 415 F.3d at 1317–19 (internal quotation marks omitted). Extrinsic evidence may assist the court in understanding the underlying technology, the meaning of terms to one skilled in the art, and how the invention works. *Id.* Extrinsic evidence, however, is less reliable and less useful in claim construction than the patent and its prosecution history. *Id*.

"A claim construction is persuasive, not because it follows a certain rule, but because it defines terms in the context of the whole patent." *Renishaw PLC v. Marposs Societa' per Azioni*, 158 F.3d 1243, 1250 (Fed. Cir. 1998). It follows that "a claim interpretation that would exclude the inventor's device is rarely the correct interpretation." *Osram GMBH v. Int'l Trade Comm'n*, 505 F.3d 1351, 1358 (Fed. Cir. 2007) (citation and internal quotation marks omitted).

II. BACKGROUND

The specifications of the ’397 and ’168 patents are “substantively identical” (D.I. 71 at 6), as are, separately, the specifications of the ’755, ’287, and ’044 patents (*id.* at 14). The two common specifications are different from each other; for claim construction purposes, the two sets of patents are unrelated and therefore extrinsic evidence to each other. The inventions display an in-work webpage in real time so a web developer can view the webpage during editing as it would appear to an end user viewing the webpage through a browser. The following claims are the most relevant for the purposes of this Markman:

Claim 1 of the ’397 Patent

> 1. A method to allow users to produce Internet websites on and for computers having a browser and a virtual machine capable of generating displays, said method comprising:
>
> (a) presenting a viewable menu having a user selectable panel of settings describing elements on a website, said panel of settings being presented through a browser on a computer adapted to accept one or more of said selectable settings in said panel as inputs therefrom, and where at least one of said user selectable settings in said panel corresponds to commands to said virtual machine;
>
> (b) generating a display in accordance with one or more user selected settings substantially contemporaneously with the selection thereof;
>
> (c) storing information representative of said one or more user selected settings in a database;
>
> (d) generating a website at least in part by retrieving said information representative of said one or more user selected settings stored in said database; and
>
> (e) building one or more web pages to generate said website from at least a portion of said database and *at least one run time file*, where said *at least one run time file* utilizes information stored in said database to generate virtual machine commands for the display of at least a portion of said one or more web pages.

(D.I. 1-1, Ex. A (“the ’397 patent”), claim 1) (emphasis added).

Claim 2 of the ’397 Patent

2. An apparatus for producing Internet websites on and for computers having a browser and a virtual machine capable of generating displays, said apparatus comprising:

(a) an interface to present a viewable menu of a user selectable panel of settings to describe elements on a website, said panel of settings being presented through a browser on a computer adapted to accept one or more of said selectable settings in said panel as inputs therefrom, and where at least one of said user selectable settings in said panel corresponds to commands to said virtual machine;

(b) a browser to generate a display in accordance with one or more user selected settings substantially contemporaneously with the selection thereof;

(c) a database for storing information representative of said one or more user selected settings; and

(d) a build tool having at least one run time file for generating one or more web pages, said run time file operating to utilize information stored in said database to generate commands to said virtual machine for generating the display of at least a portion of said one or more web pages.

(*Id.*, claim 2).

Claim 9 of the '397 Patent

9. The apparatus of claim 2, wherein said elements include a button or an images (sic), wherein said selectable settings includes the selection of an element style, and wherein said *build engine* includes means for storing information representative of selected style in said database.

(*Id.*, claim 9) (emphasis added).

Claim 1 of the '168 Patent

1. A system for assembling a web site comprising:

a server comprising a *build engine* configured to:

accept user input to create a web site, the web site comprising a plurality of web pages, each web page comprising a plurality of objects.

accept user input to associate a style with objects of the plurality of web pages, wherein each web page comprises at least one button object or at least one image object, and wherein the at least one button object or at least one image object is associated with a style that includes values defining transformations and time lines for the at least one button object or at least one image object; and wherein

each web page is defined entirely by each of the plurality of objects comprising the web page and the style associated with the object,

produce a database with a multidimensional array comprising the objects that comprise the web site including data defining, for each object, the object style, an object number, and an indication of the web page that each object is part of, and

provide the database to a server accessible to web browser;

wherein the database is produced such that a web browser with access to a *runtime engine* is configured to generate the web-site from the objects and style data extracted from the provided database.

(D.I. 1-2, Ex. B. ("the '168 patent"), claim 1) (emphasis added).

Claim 1 of the '755 Patent

1. A system for generating code to provide content on a display of a device, said system comprising:

computer memory storing a *registry* of:

(a) symbolic names required for evoking one or more *web components* each related to a set of inputs and outputs of a *web service* obtainable over a network, where the *symbolic names* are character strings that do not contain either a persistent address or pointer to an output value accessible to the *web service*, and

(b) the address of the *web service*;

an authoring tool configured to:

define a user interface (UI) object for presentation on the display, where said UI object corresponds to the *web component* included in said *registry* selected from the group consisting of an input of the web service and an output of the web service,

access said computer memory to select the symbolic name corresponding to the *web component* of the defined UI object,

associate the selected *symbolic name* with the defined UI object,

produce an *Application* including the selected *symbolic name* of the defined UI object, where said *Application* is a device-independent code, and

produce a Player, where said Player is a device-dependent code;

such that, when the *Application* and Player are provided to the device and executed on the device, and when a user of the device provides one or more input values associated with an input *symbolic name* to an input of the defined UI object,

1) the device provides the user provided one or more input values and corresponding input *symbolic name* to the web service,

2) the *web service* utilizes the input *symbolic name* and the user provided one or more input values for generating one or more output values having an associated output *symbolic name*,

3) said Player receives the output *symbolic name* and corresponding one or more output values and provides instructions for a display of the device to present an output value in the defined UI object.

(D.I. 1-3, Ex. C ("the '755 patent"), claim 1) (emphasis added).

Claim 1 of the '287 Patent

1. A system for generating code to provide content on a display of a device, said system comprising:

computer memory storing a *registry* of:

(c) *symbolic names* required for evoking one or more *web components* each related to a set of inputs and outputs of a *web service* obtainable over a network, where the *symbolic names* are character strings that do not contain either a persistent address or pointer to an output value accessible to the web service, where each *symbolic name* has an associated data format class type corresponding to a subclass of User Interface (UI) objects that support the data format type of the *symbolic name*, and has a preferred UI object, and

(d) an address of the *web service*;

an authoring tool configured to:

define a (UI) object for presentation on the display, where said defined UI object corresponds to a *web component* included in said *registry* selected from a group consisting of an input of the *web service* and an output of the *web service*, where each defined UI object is either: 1) selected by a user of the authoring tool; or 2) automatically selected by the system as the preferred UI object corresponding to the *symbolic name* of the *web component* selected by the user of the authoring tool,

access said computer memory to select the *symbolic name* corresponding to the *web component* of the defined UI object,

associate the selected *symbolic name* with the defined UI object, where the selected *symbolic name* is only available to UI objects that support the defined data format associated with that *symbolic name*, and

produce an *Application* including the selected *symbolic name* of the defined UI object, where said *Application* is a device-independent code; and

a Player, where said Player is a device-dependent code, wherein, when the *Application* and Player are provided to the device and executed on the device, and when the user of the device provides one or more input values associated with an input *symbolic name* to an input of the defined UI object,

1) the device provides the user provided one or more input values and corresponding input *symbolic name* to the *web service*,

2) the *web service* utilizes the input *symbolic name* and the user provided one or more input values for generating one or more output values having an associated output *symbolic name*,

3) said Player receives the output *symbolic name* and corresponding one or more output values and provides instructions for the display of the device to present an output value in the defined UI object.

(D.I. 1-4, Ex. D ("the '287 patent"), claim 1) (emphasis added).

Claim 1 of the '044 Patent

1. A system for generating code to provide content on a display of a device, said system comprising:

computer memory storing:

(e) *symbolic names* required for evoking one or more *web components* each related to a set of inputs and outputs of a *web service* obtainable over a network, where the *symbolic names* are character strings that do not contain either a persistent address or pointer to an output value accessible to the web service, where each *symbolic name* has an associated data format class type corresponding to a subclass of User Interface (UI) objects that support the data format type of the *symbolic name*, and where each symbolic name has a preferred UI object, and

(f) an address of the *web service*;

an authoring tool configured to:

define a (UI) object for presentation on the display,

where said defined UI object corresponds to a *web component* included in said computer memory selected from a group consisting of an input of the *web service* and an output of the *web service*, where each defined UI object is either:

1) selected by a user of the authoring tool; or

2) automatically selected by the system as the preferred UI object corresponding to the *symbolic name* of the *web component* selected by the user of the authoring tool,

access said computer memory to select the *symbolic name* corresponding to the *web component* of the defined UI object,

associate the selected *symbolic name* with the defined UI object, where the selected *symbolic name* is only available to UI objects that support the defined data format associated with that *symbolic name*,

store information representative of said defined UI object and related settings in a database;

retrieve said information representative of said one or more said UI object settings stored in said database; and

build an *application* consisting of one or more web page views from at least a portion of said database utilizing at least one player, where said player utilizes information stored in said database to generate for the display of at least a portion of said one or more web pages,

wherein when the *application* and player are provided to the device and executed on the device, and

when the user of the device provides one or more input values associated with an input *symbolic name* to an input of the defined UI object, the device provides the user provided one or more input values and corresponding input *symbolic name* to the *web service*, the *web service* utilizes the input *symbolic name* and the user provided one or more input values for generating one or more output values having an associated output *symbolic name*,
and the player receives the output *symbolic name* and corresponding one or more output values and provides instructions for the display of the device to present an output value in the defined UI object.

(D.I. 1-5, Ex. E ("the '044 patent"), claim 1) (emphasis added).

III. CONSTRUCTION OF AGREED-UPON TERMS

I adopt the following agreed-upon constructions:

Claim Term	**Construction**
multi-dimensional array(s) / multidimensional array(s)	"a uniquely identifiable indexed set of related elements, wherein each element is addressed by a set of two or more indices, each index corresponding to a dimension of the array"
storing information representative of said one or more user selected setting in a database	"storing data in a database, which data pertains to one or more attributes of an object available for selection by a user"
Transformation	"the changing of an object from one state to another based on a timer control, subject to user settings"
Settings	"attributes of an object available for selection"
authoring tool / authoring tool configured to	a system, with a graphical interface, for generating code to display content on a device screen
device-dependent code	code that is specific to the operating system, programming language, or platform of a device
device-independent code	code that is not specific to the operating system, programming language, or platform of a device
where said application is device-dependent code	where said Application is a device-independent code[1]
for evoking one or more web components	for calling up one or more web components

IV. CONSTRUCTION OF DISPUTED TERMS

I have construed below the terms the parties selected for oral argument. The parties may choose to argue claim construction issues for other terms as necessary to any summary judgment

[1] I am not positive the parties meant to capitalize the "a" in Application. But since they did, I do too.

motions, but, especially given the sparse briefing on many of the disputed terms, I do not decide them here.

1. "at least one run time file / one or more run time files" ('397/1, 2, 37)

a. *Plaintiff's proposed construction*:
   i. "one or more files, including a run time engine, that are downloaded or created when a browser is pointed to a web page or website"
b. *Defendant's proposed construction*:
   i. "one or more files, including a run time engine, that are downloaded or created and executed by a browser when a browser is pointed to a web page or website"
c. *Court's construction*:
   i. "one or more files, including a run time engine, that are downloaded or created when a browser is pointed to a web page or website"

The parties dispute whether the "run time file(s)" must be executable by a browser.

Plaintiff argues that though some "run time files" may be executable, other files downloaded by a browser to display web page content, such as image, audio, or video files, are not executable. (D.I. 64 at 31). Plaintiff further asserts that specifying "executable" is unnecessary because the run time engine file is by its nature executable. (*Id.*).

Defendant agrees that the run time engine file is necessarily executable. (*Id.*). It argues that, because the "files" can, as Plaintiff asserts, include non-executable files, it is important to explicitly require that at least one file be executable. (*Id.* at 32).

I agree with Plaintiff. The construction requires a run time engine file. If a run time engine file is executable, which both parties agree is the case, there is no meaningful difference between the two constructions other than that Defendant's inclusion of "executed" is redundant. I therefore construe "at least one run time file / one or more run time files" as "one or more files, including a run time engine, that are downloaded or created when a browser is pointed to a web page or website."

2. **“build engine” (’397/9, 14, 19, 23; ’168/1, 2)**

    a. *Plaintiff’s proposed constructions*:
        i. no construction necessary; or
        ii. “a software component for processing user input related to building a website/webpage(s) for database storage”
    b. *Defendant’s proposed constructions*:
        i. ’397 patent:
            1. indefinite; or
            2. same construction as for ’168 patent
        ii. ’168 patent
            1. “component that receives data or information regarding the creation, editing and/or display of a web page and updates one or more databases, including a database internal to the build engine, in response to that information or data”
    c. *Court’s construction*:
        i. not indefinite; “build engine” means “build tool” in the ‘397 patent; any construction of “build engine” in the ‘168 patent is deferred

Defendant argues that “build engine” in the ‘397 patent is indefinite because it lacks antecedent basis, so that a POSA would not understand what “build engine” refers to with reasonable certainty. (D.I. 64 at 23–24). Although Defendant acknowledges the USPTO’s Certificate of Correction (COC), issued to correct “build engine” to “build tool” in the ’397 patent (D.I. 65-9, Ex. 1I at 2), it presents two arguments against the use of the COC for claim construction of this term. First, Defendant argues that because the COC was not issued until February 13, 2018 (*id.*), to the extent the COC is applicable, it does not apply before its date of issuance (D.I. 77 at 32). Second, Defendant argues that, in any case, the COC should be disregarded because the specification, when discussing Fig. 3A, illustrates that a “build engine” is a “build tool” component. (D.I. 64 at 24, 27). The two terms therefore cannot be equivalent, Defendant maintains. (*Id.* at 27).

Plaintiff, on the other hand, argues that no construction is necessary because the COC makes clear that the antecedent basis for “build engine” is “build tool.” (*Id.* at 22–23). Plaintiff also argues that no construction is necessary because the specification equates the two terms

when it states the phrase "build engine (i.e. build tool)." (*Id.* at 22) (citing '397 patent at 2:1–4). For the same reason, Plaintiff maintains, the COC properly treats "build engine" and "build tool" as interchangeable terms. (*Id.* at 26).

There are therefore two issues related to the COC. First, whether the COC's corrections only apply after the COC's date of issuance. Second, whether the COC's correction is proper.

The Federal Circuit has held that "for causes arising after the PTO issues a certificate of correction, the certificate of correction is to be treated as part of the original patent—i.e., as if the certificate had been issued along with the original patent." *Southwest Software, Inc. v. Harlequin, Inc.*, 226 F.3d 1280, 1295 (Fed. Cir. 2000). For causes of action that arise after the date of the COC's issuance, therefore, the COC's corrected claim language has effect dating back to the priority date of the patent because "the certificate is considered part of the original patent." *Id.*

The COC was issued on February 13, 2018. (D.I. 65-9, Ex. 1I at 2). Plaintiff alleges Defendant was made aware of its infringement as early as February 28, 2013 but does not specify the date of initial infringement. (D.I. 47 at 16, 25). Relevant to the issue here, Plaintiff alleges infringement before and after the date of the COC's issuance. Because the COC's corrected language is treated as "part of the original patent," for any alleged infringement after the date of the COC's issuance, February 13, 2018, the corrected claim language applies.

The corrected claim language does not apply, on the other hand, to alleged infringement prior to February 13, 2018. If a COC were given effect as of the priority date of the patent for infringement *prior* to the date of the COC's issuance, a party could be held liable for infringing a facially invalid patent. "In such a case, where the claim is invalid on its face without the certificate of correction, it strikes us as an illogical result to allow the patent holder, once the certificate of correction has issued, to sue an alleged infringer for activities that occurred before

the issuance of the certificate of correction." *Southwest Software,* 226 F.3d at 1295–96. The corrected claim language therefore does not apply to any alleged infringement prior to February 13, 2018, the date of COC issuance.

The second issue relates to whether the COC's issuance was proper.

Certificates of correction may be issued for "a mistake of a clerical or typographical nature, or of minor character" at the discretion of the USPTO. 35 U.S.C. § 255. The Federal Circuit has held that although § 255 does "allow broadening corrections of clerical or typographical mistakes," there is clear Congressional intent "to protect the public against the unanticipated broadening of a claim after the grant of the patent by the PTO." *Superior Fireplace Co. v. Majestic Products Co.*, 270 F.3d 1358, 1371 (Fed. Cir. 2001). "It would be inconsistent with that objective to interpret § 255 to allow a patentee to broaden a claim due to the correction of a clerical or typographical mistake that the public could not discern from the public file and for which the public therefore had no effective notice." *Id.* The "public file" used to determine whether a correction is proper consists of "the specification, drawings, and prosecution history." *Id.* at 1372. As for a mistake of "minor character": "A mistake that, if corrected, would broaden the scope of a claim must thus be viewed as highly important and thus cannot be a mistake of 'minor character.'" *Id.* at 1375.

In the '397 patent, the relevant claim recites, "The apparatus of claim 2, wherein said elements include a button or an images (sic), wherein said selectable settings includes the selection of an element style, and wherein said build engine includes means for storing information representative of selected style in said database." ('397 patent at 66:48–49 (uncorrected claim 9)). There is no grammatical error in the use of "build engine" that suggests a mistake on its face. Looking to independent claim 2, however, although dependent claim 9

references "said build engine," claim 2 recites only "a build tool," not a "build engine." This raises the possibility of a clerical or typographical error in the use of "said build engine."

The specification teaches that a "build engine" is an example of a "build tool." ('397 patent at 9:1–29). Although Defendant argues that these two terms are distinguished in Fig. 3A, the relevant specification language stating that a "build engine" is an example of a "build tool" explicitly refers to Fig. 3A. (*Id.*). Looking at the claim language, claim 9 references "said build engine" while claim 2 recites a "build tool." Because the specification states that a "build engine" is an example of a "build tool," it is therefore reasonable to expect that a reader of the patent could discern that "build engine" in claim 9 of the '397 patent is meant to refer to the "build tool" in claim 2.

I therefore find that the COC's correction of "build engine" to "build tool" in the '397 patent is a valid correction of a "clerical or typographical mistake" under § 255.

For the construction of "build engine" in the '397 patent as it applies to infringement prior to the COC's date of issuance, I analyze indefiniteness. "[A] patent is invalid for indefiniteness if its claims, read in light of the specification delineating the patent, and the prosecution history, fail to inform, with reasonable certainty, those skilled in the art about the scope of the invention." *Nautilus, Inc. v. Biosig Instruments, Inc*., 572 U.S. 898, 901 (2014). In order to prove a patent is invalid for indefiniteness, there must be clear and convincing evidence. *Id*. at 912 n.10 (citing *Microsoft Corp. v. i4i Ltd. Partnership*, 564 U.S. 91, 95 (2011)).

I do not think Defendant has proven indefiniteness by clear and convincing evidence for many of the same reasons cited in the COC analysis. The specification states that a "build engine" is an example of a "build tool," specifically in relation to Fig. 3A ('397 patent at 9:1–29), which Defendant argues distinguishes the two (D.I. 64 at 27). No other argument was

presented in the briefing or at oral argument that a POSA would not have been able to determine, with reasonable certainty, that the "build engine" in claim 9 referred to the "build tool" in claim 2. Because the specification notes that a "build engine" is an example of a "build tool," I think a POSA would have understood with reasonable certainty that "said build engine" in claim 9 referred to the "build tool" in claim 2.

I therefore find for the '397 patent that "build engine" is not indefinite, and that "build engine" should be construed as "build tool." Because the parties did not present "build tool" for construction at oral argument, I will not construe "build tool" here. The parties may present any claim construction disputes regarding "build tool" in their summary judgment briefing.[2]

3. **"run time (runtime) engine" ('168/1)**

    a. *Plaintiff's proposed constructions*:
        i. "file that is executed at runtime that utilizes information from the database and generates commands to display a web page or website"; or
        ii. "file that is executed at runtime that reads information from the database and generates commands to display a web page or website"
    b. *Defendant's proposed construction*:
        i. "file that is executed at runtime that reads information from the database and generates virtual machine commands to display a web page or website"
    c. *Court's construction*:
        i. "file that is executed at runtime that reads information from the database and generates commands to display a web page or website"

I previously construed "runtime engine" to mean "file that is executed at runtime that reads information from the database and generates commands to display a web page or website." (D.I. 65-6, Ex. 1E at 5). To the extent Plaintiff disagrees with this construction, it argues that "utilizes information" could mean something more than simply reading information from the

[2] I also do not construe "build engine" in the '168 patent. The proposed construction was barely briefed (D.I. 64 at 22-27) and was barely mentioned at the Markman hearing. (D.I. 77 at 36, 39-40).

database. (D.I. 64 at 38–39). At oral argument, Plaintiff focused its argument on concerns that Defendant may later argue that reading and downloading information should be construed differently. (D.I. 77 at 49:21–50:21). Defendant agreed that "downloading is equivalent to reading over a network." (*Id.* at 46:14–47:3). I therefore focus my analysis on the difference between my prior construction, which Plaintiff accepted as an alternative construction, and Defendant's construction, which construes "commands" as limited solely to "virtual machine commands."

Plaintiff argues that the claim language does not refer explicitly to virtual machine commands. (D.I. 64 at 40). Plaintiff also maintains that the specification "confirms that a run time engine can generate other types of commands to display a web page or website, such as commands for downloading image, audio and video files." (*Id.*) (citing '168 patent at 5:49–68). Defendant, on the other hand, argues that the runtime engine "generates the web-site" using information stored in the database, a process for which the specification describes issuing virtual machine commands. (*Id.* at 42).

I agree with Plaintiff. The term "virtual machine" does not appear in the claim language. Defendant's citation to the specification for the process of generating the web-site is only in reference to an embodiment; it does not indicate that the process must exclusively be done using virtual machine commands. (*See* '168 patent at 35:8–12). The specification does not otherwise limit the commands involved in displaying the web page to virtual machine commands. I therefore construe "runtime engine" as "file that is executed at runtime that reads information from the database and generates commands to display a web page or website."

4. **"registry" ('755/1, 12; '287/1, 15)**

    a. *Plaintiff's proposed constructions*:
        i. no construction necessary; or

ii. "a database or lookup table that is used for computing functionality"

b. *Defendant's proposed construction*:
   i. "a database, XML file, or Portable Description Language file that exists on a computer"

c. *Court's construction*:
   i. "a database that is used for computing functionality"

The parties generally agree that "registry" is a term of art in the computer science and web design fields. (D.I. 64 at 45–46). They dispute whether Plaintiff acted as its own lexicographer to limit the definition of "registry" to that provided in the specification. (*Id.*).

Plaintiff argues that "registry" should not be limited to "a database, XML file, or Portable Description Language" because that description, while in the specification, refers only to "one embodiment that references the 'web component registry.'" (*Id.* at 45) (citing '755 patent at 8:19–22). Other embodiments, Plaintiff argues, disclose a web component registry that could be implemented in ways other than "database, XML file, or Portable Description Language," such as by using a list with relevant information. (*See id.* at 46) (citing '755 patent at 9:16–26, 22:26–29).

Defendant, on the other hand, argues that the specification reference to "web component registry 230" is definitional because it does not contain language limiting the term to a single embodiment. (D.I. 64 at 45).

I generally agree with Plaintiff. The language in the specification referring to "web component registry" is limited to the description of Fig. 2A, which is described as "an embodiment." ('755 patent at 7:63–66). Nothing in the specification indicates the patentee's desire to redefine or limit the term "registry" to only a "database, XML file, or Portable Description Language"; that description is merely used to describe one such manifestation of the element in an embodiment.

I do not, however, agree with Plaintiff's inclusion of a "lookup table" in its construction. Plaintiff cites to the specification to support its assertion that "registry" can take a tabular format, but the cited portions only teach a list with information, not a table. ('755 patent at 9:16–26, 22:26–29). The parties do at minimum agree, though, that the construction of "registry" requires a "database."

It might be helpful to a jury to have a construction. I will adopt Plaintiff's alternative construction, except for "lookup table," and I therefore construe "registry" as "a database that is used for computing functionality."

5. **"web component" ('755/1, 3, 6, 7, 12, 14, 17, 18; '287/1, 3, 6, 7, 15, 17, 20, 21; '044/1, 3, 6, 7, 15, 17, 20, 21)**

    a. *Plaintiff's proposed construction*:
        i. "one or more functionalities associated with one or more web page elements to be displayed on a device"
    b. *Defendant's proposed construction*:
        i. "software object, which has a clearly defined interface, conforms to a prescribed behavior common to all components within an architecture, is meant to interact with other components, and encapsulates certain functionality or a set of functionalities"
    c. *Court's construction*:
        i. "software objects that provide functionalities of a web service"

The parties partially agreed at oral argument to construe "web component" as "software objects that provide functionalities of a web service," disputing only whether to include "certain" in front of "functionalities." (D.I. 77 at 70:22–73:14).

Neither party meaningfully argues why "certain" should or should not be included. The scope of "functionalities" provided by the "software objects" is already limited to the context of a "web service." Further specifying only "certain" functionalities does not change the scope of the construction.

I therefore construe "web component" as "software objects that provide functionalities of a web service."

6. **"web service" ('755/1, 5, 7, 12, 16, 18; '287/1, 5, 7, 15, 19, 21; '044/1, 5, 7, 15, 19, 21)**

    a. *Plaintiff's proposed constructions*:
        i. no construction necessary; or
        ii. "A software system that supports interaction between devices over a network"
    b. *Defendant's proposed construction*:
        i. "A software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format called "WSDL" (web service description language). Other systems interact with the web service in a manner prescribed by its description using Simple Object Access Protocol ("SOAP") messages, typically conveyed using HTTP with an XML serialization in conjunction with other web-related standards."
    c. *Court's construction*:
        i. "A software system that supports interaction between devices over a network."

Plaintiff argues that no construction of "web service" is necessary because it is a well understood term of art for which definitions "abound in computer literature and technical dictionaries." (D.I. 64 at 50). To the extent a construction is required, Plaintiff asserts, its alternative construction is supported by the specification description of a "web service" as "a plurality of services obtainable over the Internet." (*Id.*) (citing '755 patent at 8:18–19).

Defendant instead asserts that "web service" should be construed and that, because during prosecution the application referenced the "World Wide Web" in conjunction with "web service," the World Wide Web Consortium (W3C) definition should apply here. (D.I. 64 at 50–51). Defendant then asserted at oral argument—without having included any prior reference in its three paragraphs of briefing on this term—that the patentee used the W3C definition as part of an amendment to overcome the prior art McCain reference, which Defendant argued would constitute prosecution history disclaimer. (D.I. 77 at 60:4–24).

Plaintiff replied that the reference to the prosecution history was just general background and not asserted by the patentee in a way that would constitute disclaimer. (*Id.* at 63:23–64:3). Plaintiff also argued that the W3C definition is overly limiting because it excludes web service interfaces, such as REST, that are specifically referenced in the specification. (*Id.* at 56:3–15; *see* '755 patent at 28:14–18).

I mostly agree with Plaintiff. The reference to the W3C definition in the prosecution history cited by Defendant appears in a section that generally describes the various components of the application, including a "web service." (D.I. 42-15, Reply to Office Action of April 1, 2012,[3] at 5–6). To the extent that the patentee used this information to overcome the combination of prior art references McCain and Sidman, the patentee did not delimit the type of components constituting the "web service" or use the latter, restrictive portion of the W3C definition in patentee's assertions. (*Id.*).

As for disclaimer, the parties have not provided sufficient argument for me to evaluate this issue. Defendant's argument is essentially that the application, during prosecution, referenced the World Wide Web when referring to "web service," and therefore I should adopt the W3C definition of the term. (D.I. 64 at 50–51). I have already noted that this definition appears in a background section and, consequently, does not constitute disclaimer without other indicia of disavowal. Defendant presents no additional arguments for disclaimer. In fact, the collective briefing on this term covers only two and a half pages, with only a few sentences dedicated to arguing about disclaimer. Prosecution history disclaimer "requires that the alleged disavowing actions or statements made during prosecution be both clear and unmistakable." *Omega Eng'g, Inc. v. Raytek Corp.*, 334 F.3d 1314, 1325–26 (Fed. Cir. 2003). Based on the

[3] The patentee used "Reply to Office Action of April 1, 2012" in the header of the document, but in the text describes a response to an office action dated April 1, 2013 (D.I. 42-15 at 1), which I think is correct.

briefing, it is not apparent to me that the patentee's statements arise to "clear and unmistakable" disclaimer.

Plaintiff's alternative construction better fits the description of "web service" in the specification as a "a plurality of services obtainable over the Internet." ('755 patent at 8:18–19). I therefore construe "web services" as "A software system that supports interaction between devices over a network."

7. **"symbolic name(s)" ('755/1, 12; '287/1, 15; '044/1, 15)**

    a. *Plaintiff's proposed construction*:
        i. no construction necessary
    b. *Defendant's proposed construction*:
        i. "parameters specifying inputs and/or outputs associated with web services"
    c. *Court's construction*:
        i. no construction necessary

Plaintiff argues that no construction is necessary because the claim language itself defines "symbolic names" as "character strings that do not contain either a persistent address or pointer to an output value accessible to the web service." (D.I. 64 at 52) (citing '755 patent, claim 1). Defendant, on the other hand, argues that this definition merely describes what "symbolic names" cannot be and is therefore insufficient. (*Id.* at 54). Defendant also asserts that "symbolic names" should be limited to "parameters" that specify "inputs and/or outputs associated with web services," consistent with patentee's representations to overcome prior art references during prosecution. (*Id.* at 53) (citing D.I. 42-15, Reply to Office Action of April 1, 2012, at 5–6).

I agree with Defendant that the patentee argued that "symbolic names" are parameters associated with inputs or outputs related to web services. But Defendant has not argued how or why this characterization of "symbolic names" subsumes or contradicts the definition set forth in claim 1 of the '755 patent. Nor has Defendant explained why it thinks the definition of

"symbolic names" set forth in claim 1 is impermissibly broader than the characterization of "symbolic names" offered by the patentee during prosecution. Claim 1 affirmatively describes "symbolic names" as "character strings." The claim further limits "symbolic names" to only those "character strings" that "do not contain either a persistent address or pointer to an output value accessible to the web service." Not having a "persistent address" was, in fact, the exact characteristic of "symbolic name" asserted by the patentee to overcome prior art references McCain and Sidman during prosecution. (D.I. 42-15, Reply to Office Action of April 1, 2012, at 5–6). Because the language of the claim expressly limits the scope of "symbolic name" in a manner consistent with patentee's arguments during prosecution—and because Defendant has not otherwise provided reasonable basis for changing the scope based on the prosecution history—there is no need to further construe the term.

No construction of "symbolic names" is necessary.

8. **"application" ('755/1, 12, 22; '287/1, 15, 25; '044/1, 15, 25)**

    a. *Plaintiff's proposed construction*:
        i. "device-independent software code containing instructions for a device"
    b. *Defendant's proposed constructions*:
        i. "device-independent code which contains instructions for a device and which is separate and independent from the Player"; or
        ii. "device-independent code which contains instruction for a device and which is separate and distinct from the Player"
    c. *Court's construction*:
        i. "device-independent code containing instructions for a device and which is separate from the Player"

At oral argument, Plaintiff preserved its objection to incorporating "separate" in the construction, but otherwise agreed that the application is separate from the player. (D.I. 77 at 74:17–75:3). Plaintiff also agreed that the word "software" could be taken out of its construction

without changing the meaning. (*Id.* at 74:12–15). I therefore focus my analysis on whether the construction should read "separate *and independent*" or just "separate."

Plaintiff argues that the specification and prosecution history do not use the "separate and independent" language to delimit the scope of "application." (D.I. 77 at 75:17–20). Plaintiff also asserts that although I included "separate" in my prior construction of "application" (i.e. "device-independent code which contains instructions for a device and which is separate from the Player"), I rejected the requirement that the "application" be "partitioned" from the "Player." (*Id.* at 75:21–76:2). Defendant's requirement that the application be "independent" in addition to merely being "separate" is, Plaintiff maintains, an improper attempt to partition the "application" from the "Player," especially in light of specification text that, referring to an embodiment, explains that the "application" can be extended "on the Player so that it is efficiently integrated into a comprehensive client/server Application." (D.I. 64 at 56) (citing '755 patent at 7:30–33).

Defendant, on the other hand, argues that the code for the "application" and "Player" are "touted as being able to be maintained separately and distinctly," so they must be not only separate lines of code but also separate files. (D.I. 77 at 80:20–82:18). Defendant also points to references in the prosecution history, including the McCain reference, to highlight that the patentee's use of the separation of the "application" and "Player" to argue that the invention was novel supports construction of "application" to mean "separate and independent" from the "Player." (D.I. 64 at 57).

I agree with Plaintiff. Defendant's argument here—based on the patentee's assertion of novelty using the separation of the "application" and "Player"—is substantively identical to the argument it made when I issued my prior construction. (D.I. 65-6, Ex. 1E at 7). As I did there, I acknowledge that "separate" should be included in the construction of "application" because the

patentee's assertion that the separation of the "application" and "Player" is novel constitutes prosecution history disclaimer. (*Id.*). In making this novelty argument, the patentee only asserted, however, that the bodies of code for the "application" and "Player" are separate, not that the "application" and "Player" need to be housed independently or distinctly from each other. (*See id.*). Moreover, including "independent" or "distinct" excludes embodiments such as the one highlighted by the specification in which the "Player" is "efficiently integrated" with the "Application." (*Id.*; '755 patent at 7:30–33).

To be sure, the code for the "application" and "Player" are "separate" sets of code. To the extent "independent" or "distinct" are meant to mean something more than "separate," however, the disclaimer does not support that conclusion. Patentee's statements refer only to "partitioning the code" required for the Application and for the Player (i.e. maintaining a separate body of code for each). (D.I. 65-6, Ex. 1E at 7). This "partitioning" is already conveyed by "separate" in the construction. Adding "independent" or "distinct" would imply that there is some other difference between the code for the Application and the code for the Player that goes beyond there merely being "separate" bodies of code. No such additional difference is supported by the patentee's statements during prosecution. Furthermore, since the code for "Application" and "Player" can be integrated on the same server ('755 patent at 7:30–33), adding "independent" or "distinct" to the construction would serve no purpose other than to confuse.

I therefore construe "application" as "device-independent code containing instructions for a device and which is separate from the Player."

V. CONCLUSION

Within five days the parties shall submit a proposed order consistent with this Memorandum Opinion suitable for submission to the jury.