

**IN THE UNITED STATES DISTRICT COURT  
FOR THE EASTERN DISTRICT OF TEXAS  
TYLER DIVISION**

**PERSONALWEB TECHNOLOGIES, LLC**

**Plaintiff,**

**vs.**

**NEC CORPORATION OF AMERICA, INC.**

**Defendant.**

§  
§  
§  
§  
§  
§  
§  
§  
§  
§  
§

**CASE NO. 6:11-CV-655  
PATENT CASE**

**PERSONALWEB TECHNOLOGIES, LLC**

**Plaintiff,**

**vs.**

**GOOGLE INC. AND YOUTUBE, LLC**

**Defendants.**

§  
§  
§  
§  
§  
§  
§  
§  
§  
§  
§

**CASE NO. 6:11-CV-656  
PATENT CASE**

**PERSONALWEB TECHNOLOGIES, LLC**

**Plaintiff,**

**vs.**

**NETAPP, INC.**

**Defendant.**

§  
§  
§  
§  
§  
§  
§  
§  
§  
§  
§

**CASE NO. 6:11-CV-657  
PATENT CASE**

---

**PERSONALWEB TECHNOLOGIES, LLC**

**Plaintiff,**

vs.

**AMAZON.COM, INC.; AMAZON WEB  
SERVICES LLC; AND DROPBOX, INC.**

**Defendants.**

---

§  
§  
§  
§  
§  
§  
§  
§  
§  
§

**CASE NO. 6:11-CV-658  
PATENT CASE**

---

**PERSONALWEB TECHNOLOGIES, LLC**

**Plaintiff,**

vs.

**EMC CORPORATION, AND  
VMWARE, INC.**

**Defendants.**

---

§  
§  
§  
§  
§  
§  
§  
§  
§  
§

**CASE NO. 6:11-CV-660  
PATENT CASE**

---

**PERSONALWEB TECHNOLOGIES, LLC**

**Plaintiff,**

vs.

**AUTONOMY, INC., ET AL.,**

**Defendants.**

---

§  
§  
§  
§  
§  
§  
§  
§  
§  
§

**CASE NO. 6:11-CV-683  
PATENT CASE**

---

**PERSONALWEB TECHNOLOGIES, LLC**

**Plaintiff,**

vs.

**YAHOO! INC.**

**Defendant.**

§  
§  
§  
§  
§  
§  
§  
§  
§  
§

**CASE NO. 6:12-CV-658  
PATENT CASE**

---

**PERSONALWEB TECHNOLOGIES, LLC**

**Plaintiff,**

vs.

**APPLE INC.**

**Defendant.**

§  
§  
§  
§  
§  
§  
§  
§  
§  
§

**CASE NO. 6:12-CV-660  
PATENT CASE**

---

**PERSONALWEB TECHNOLOGIES, LLC**

**Plaintiff,**

vs.

**FACEBOOK, INC.**

**Defendant.**

§  
§  
§  
§  
§  
§  
§  
§  
§  
§

**CASE NO. 6:12-CV-662  
PATENT CASE**

---

**MEMORANDUM OPINION AND ORDER**

This Memorandum Opinion construes the disputed claim terms in U.S. Patent Nos. 5,978,791 (“the ‘791 Patent), 6,415,280 (“the ‘280 Patent), 6,928,442 (“the ‘442 Patent), 7,802,310 (“the ‘310 Patent), 7,945,539 (“the ‘539 Patent), 7,945,544 (“the ‘544 Patent), 7,949,662 (“the ‘662 Patent), 8,001,096 (“the ‘096 Patent), and 8,099,420 (“the ‘420 Patent).

Autonomy and Hewlett-Packard's Motion for Summary Judgment of Indefiniteness (6:11-CV-683, Docket No. 164) is **DENIED**. Facebook's Motion for Summary Judgment of Indefiniteness (6:12-CV-662, Docket No. 66) is **DENIED**.

### **BACKGROUND**

The Plaintiff PersonalWeb Technologies LLC ("PersonalWeb") sued the following Defendants for infringement: NEC Corporation of America, Inc. ("NEC"); Google, Inc. and YouTube, LLC ("Google"); NetApp, Inc. ("NetApp"); Amazon.com, Inc. and Amazon Web Services LLC ("Amazon"); Dropbox, Inc. ("Dropbox"); EMC Corp. and VMWare, Inc. ("EMC"); Autonomy, Inc., Hewlett-Packard Co., and HP Enterprise Services, LLC ("HP"); Yahoo! Inc. ("Yahoo"); Apple Inc. ("Apple"); and Facebook, Inc. ("Facebook"). The Court heard oral argument on July 18, 2013.

There are nine asserted patents, all claiming priority to a common application. A number of the Patents have been the subject of post-issuance examination. Inter Partes Reviews ("IPRs") have been initiated for several of the Patents, and the Patent Trial and Appeal Board ("PTAB") has issued decisions for at least three Patents. The Patents generally relate to methods for identifying data items in a data processing system.

### **APPLICABLE LAW**

"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) (quoting Innova/Pure Water Inc. v. Safari Water Filtration Sys., Inc., 381 F.3d 1111, 1115 (Fed. Cir. 2004)). In claim construction, courts examine the patent's intrinsic evidence to define the patented invention's scope. See *id.*; C.R. Bard, Inc. v. U.S. Surgical Corp., 388 F.3d 858, 861 (Fed. Cir. 2004); Bell Atl. Network Servs., Inc. v. Covad

*Commc'ns Group, Inc.*, 262 F.3d 1258, 1267 (Fed. Cir. 2001). This intrinsic evidence includes the claims themselves, the specification, and the prosecution history. See *Phillips*, 415 F.3d at 1314; *C.R. Bard, Inc.*, 388 F.3d at 861. Courts give claim terms their ordinary and accustomed meaning as understood by one of ordinary skill in the art at the time of the invention in the context of the entire patent. *Phillips*, 415 F.3d at 1312–13; *Alloc, Inc. v. Int'l Trade Comm'n*, 342 F.3d 1361, 1368 (Fed. Cir. 2003).

The claims themselves provide substantial guidance in determining the meaning of particular claim terms. *Phillips*, 415 F.3d at 1314. First, a term's context in the asserted claim can be very instructive. *Id.* Other asserted or unasserted claims can also aid in determining the claim's meaning because claim terms are typically used consistently throughout the patent. *Id.* Differences among the claim terms can also assist in understanding a term's meaning. *Id.* For example, when a dependent claim adds a limitation to an independent claim, it is presumed that the independent claim does not include the limitation. *Id.* at 1314–15.

“[C]laims ‘must be read in view of the specification, of which they are a part.’” *Id.* (quoting *Markman v. Westview Instruments, Inc.*, 52 F.3d 967, 979 (Fed. Cir. 1995) (en banc)). “[T]he 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.’” *Id.* (quoting *Vitronics Corp. v. Conceptor, Inc.*, 90 F.3d 1576, 1582 (Fed. Cir. 1996)); see also *Teleflex, Inc. v. Ficoso N. Am. Corp.*, 299 F.3d 1313, 1325 (Fed. Cir. 2002). This is true because a patentee may define his own terms, give a claim term a different meaning than the term would otherwise possess, or disclaim or disavow the claim scope. *Phillips*, 415 F.3d at 1316. In these situations, the inventor's lexicography governs. *Id.* Also, the specification may resolve ambiguous claim terms “where the ordinary and accustomed meaning of the words used in the claims lack

sufficient clarity to permit the scope of the claim to be ascertained from the words alone.” *Teleflex, Inc.*, 299 F.3d at 1325. But, “[a]lthough the specification may aid the court in interpreting the meaning of disputed claim language, particular embodiments and examples appearing in the specification will not generally be read into the claims.” *Comark Commc’ns, Inc. v. Harris Corp.*, 156 F.3d 1182, 1187 (Fed. Cir. 1998) (quoting *Constant v. Advanced Micro-Devices, Inc.*, 848 F.2d 1560, 1571 (Fed. Cir. 1988)); see also *Phillips*, 415 F.3d at 1323. The prosecution history is another tool to supply the proper context for claim construction because a patent applicant may also define a term in prosecuting the patent. *Home Diagnostics, Inc., v. Lifescan, Inc.*, 381 F.3d 1352, 1356 (Fed. Cir. 2004) (“As in the case of the specification, a patent applicant may define a term in prosecuting a patent.”).

Although extrinsic evidence can be useful, it is “less significant than the intrinsic record in determining the legally operative meaning of claim language.” *Phillips*, 415 F.3d at 1317 (quoting *C.R. Bard, Inc.*, 388 F.3d at 862). Technical dictionaries and treatises may help a court understand the underlying technology and the manner in which one skilled in the art might use claim terms, but technical dictionaries and treatises may provide definitions that are too broad or may not be indicative of how the term is used in the patent. *Id.* at 1318. Similarly, expert testimony may aid a court in understanding the underlying technology and determining the particular meaning of a term in the pertinent field, but an expert’s conclusory, unsupported assertions as to a term’s definition is entirely unhelpful to a court. *Id.* Generally, extrinsic evidence is “less reliable than the patent and its prosecution history in determining how to read claim terms.” *Id.*

The patents-in-suit also contain means-plus-function limitations that require construction. Where a claim limitation is expressed in means-plus-function language and does not recite

definite structure in support of its function, the limitation is subject to 35 U.S.C. § 112(f). *B. Braun Med., Inc. v. Abbott Labs.*, 124 F.3d 1419, 1424 (Fed. Cir. 1997). In relevant part, 35 U.S.C. § 112(f) “mandates that such a claim limitation ‘be construed to cover the corresponding structure . . . described in the specification and equivalents thereof.’” *Id.* (quoting 35 U.S.C. § 112(f)). Accordingly, when faced with means-plus-function limitations, courts “must turn to the written description of the patent to find the structure that corresponds to the means recited in the [limitations].” *Id.*

Construing a means-plus-function limitation involves multiple inquiries. “The first step in construing [a means-plus-function] limitation is a determination of the function of the means-plus-function limitation.” *Medtronic, Inc. v. Advanced Cardiovascular Sys., Inc.*, 248 F.3d 1303, 1311 (Fed. Cir. 2001). Once a court has determined the limitation’s function, “[t]he next step is to determine the corresponding structure described in the specification and equivalents thereof.” *Id.* A “structure disclosed in the specification is ‘corresponding’ structure only if the specification or prosecution history clearly links or associates that structure to the function recited in the claim.” *Braun*, 124 F.3d at 1424.

Software-implemented means-plus-function limitations generally require more than the disclosure of a general purpose computer as the supporting structure. *Aristocrat Techs. Austl. PTY Ltd. v. Int’l Game Tech.*, 521 F.3d 1328, 1333 (Fed. Cir. 2008). An algorithm for performing the claimed function must be disclosed. *WMS Gaming Inc. v. Int’l Game Tech.*, 184 F.3d 1339, 1349 (Fed. Cir. 1999). The algorithm may be disclosed “in any understandable terms including as a mathematical formula, in prose, or as a flow chart, or in any other manner that provides sufficient structure.” *Finisar Corp. v. DirecTV Group, Inc.*, 523 F.3d 1323, 1340 (Fed. Cir. 2008) (internal citations omitted). However, as a narrow exception, disclosure of a general

purpose computer may be sufficient if the “function[ ] can be achieved by any general purpose computer without special programming.” In re Katz Interactive Call Processing Patent Litigation, 639 F.3d 1303, 1316 (Fed. Cir. 2011).

Defendants also contend that some claims at issue are invalid for indefiniteness. A claim is invalid under 35 U.S.C. § 112(b) if it fails to particularly point out and distinctly claim the subject matter that the applicant regards as the invention. The party seeking to invalidate a claim under 35 U.S.C. § 112(b) as indefinite must show by clear and convincing evidence that one skilled in the art would not understand the scope of the claim when read in light of the specification. Intellectual Prop. Dev., Inc. v. UA-Columbia Cablevision of Westchester, Inc., 336 F.3d 1308, 1319 (Fed. Cir. 2003).

## ANALYSIS

### “data item(s)”<sup>1</sup>

PersonalWeb proposes “sequence of bits.” Defendants propose “the contents of a file, a portion of a file, a page in memory, an object in an object-oriented program, a digital message, a digital scanned image, a part of a video or audio signal, or any other similar entity which can be represented by a sequence of bits.”

Both sides cite the following passage from the specification:

In general, the terms “data” and “data item” as used herein refer to sequences of bits. Thus a data item may be the contents of a file, a portion of the file, a page in memory, an object in an object-oriented program, a digital message, a digital scanned image, a part of a video or audio signal, or any other entity which can be represented by a sequence of bits.

---

<sup>1</sup> For all construed terms, the Court’s construction reflects use of the term in the claims cited in the parties’ Joint Claim Construction Statement. See 6:11-CV-655, Docket No. 85, Ex. B. Not every Defendant proposes a construction for every term. See *id.* For some terms, one or more Defendants propose no construction is necessary. *Id.* Further, many Defendants propose their own individual constructions for numerous terms. *Id.* For the majority of terms, the Court uses Defendants’ “exemplary construction” to explain its analysis. However, it considered all proposed constructions from every Defendant for every term in its analysis. To the extent any Defendants’ specific construction is not discussed in this order, it was still considered by the Court.



'791 Patent, at 1:54–60. PersonalWeb contends this passage indicates a “data item” is simply a sequence of bits, so any further limitations would be improper. 6:11-CV-655, Docket No. 86 (“Brief”) at 15. PersonalWeb argues Defendants’ list of examples is non-exhaustive and would confuse the jury. *Id.* at 16. Additionally, PersonalWeb asserts that other portions of the specification cite different examples of data items, such as directories and records in a database. *Id.* Thus, Defendants’ list of examples is incomplete because it does not even include every example taught in the specification. *Id.* Finally, PersonalWeb challenges Defendants’ inclusion of the word “similar,” which PersonalWeb asserts has no support from the specification. *Id.*

Defendants argue PersonalWeb makes contradictory arguments regarding the patentee’s lexicography. 6:11-CV-655, Docket No. 89 (“Response”) at 29. Defendants contend PersonalWeb cannot argue the patentee expressly defined “data item” as a sequence of bits, then ignore the remainder of the patentee’s definition (which included the cited examples). *Id.* Defendants also believe their list of examples highlights that a “data item” must be an item. *Id.* at 30. Defendants contend the word “item” indicates a “data item” must be a discrete object, not an unbounded sequence of bits. *Id.*

The specification plainly and unambiguously states that a data item is a sequence of bits. '791 Patent, at 1:54–55; *see Thorner v. Sony Computer Entm't Am. LLC*, 669 F.3d 1362, 1365–66 (Fed. Cir. 2012) (noting a patentee may act as its own lexicographer). Any further construction would be unnecessary. Defendants include several illustrative examples in their proposed construction, but the specification makes clear that those examples are merely permissive. *See* '791 Patent, at 1:55 (“Thus a data item may be....”) (emphasis added). Further, the specification includes other examples of data items that are not included in Defendants’ proposal. For example, data items can be “files, directories, records in the database, objects in

object-oriented programming, locations in memory or on a physical device, or the like.” See *id.* at 1:66–2:2. It would be misleading to include a non-exhaustive list of examples in the construction when the specification teaches further examples as well. See *Dealertrack, Inc. v. Huber*, 674 F.3d 1315, 1322 (Fed. Cir. 2012) (declining to limit a term to a list of examples in the specification).

A “data item” is a “sequence of bits.”

### **“data file(s)”**

PersonalWeb proposes “a named data item having one or more data segments.” Defendants propose “a named data item that appears in a directory in a file system and which is a data file.”

The specification contains the following passage:

A file system is a collection of directories. A directory is a collection of named files—both data files and other directory files. A file is a named data item which is either a data file (which may be simple or compound) or a directory file. A simple file consists of a single data segment. A compound file consists of a sequence of data segments. A data segment is a fixed sequence of bytes.

‘791 Patent, at 5:44–52. Each side contends this passage supports its construction. PersonalWeb argues its definition comes directly from the specification because “a file is a named data item.” Brief at 17. A data file can be either simple or compound, and its construction reflects that. *Id.* PersonalWeb further challenges Defendants’ inclusion of “appears in a directory file system.” *Id.* at 17. PersonalWeb asserts that there is no requirement that all data files be located in directories. *Id.*

Defendants argue the portion of the specification relied upon by PersonalWeb cannot be read independently of the two preceding sentences. Response at 31. Defendants contend the paragraph as a whole indicates a “data file” is part of a collection of named files that are part of a

collection of directories. *Id.* Defendants assert that this construction makes sense in light of the content-based approach of file identification described in the Patents. *Id.* Defendants also contend their seemingly redundant “and which is a data file” limitation is appropriate. *Id.* Defendants argue the specification distinguishes between a “data item” and a “file,” but PersonalWeb’s construction blurs that distinction. *Id.* at 32. Defendants believe under PersonalWeb’s construction, a mere segment of data could be a “file.” *Id.*

The specification teaches that a data file may be simple or compound. ‘791 Patent, at 5:48–49. “A simple file consists of a single data segment. A compound file consists of a sequence of data segments.” *Id.* at 5:49–51. This passage indicates why PersonalWeb’s “having one or more data segments” limitation is unnecessary. A file can have one segment or multiple segments, so there is no reason to include a limitation about the number of segments. The limitation does not change the claim scope.

There is no support for Defendants’ inclusion of “that appears in a directory in a file system.” The specification states that a directory includes both data files and other directory files. *Id.* at 5:46–47. Defendants use this statement to argue a data file must be located in a directory. This is incorrect. A directory must contain data files, but there is no requirement that all data files be located in a directory.<sup>2</sup> See *id.* at 5:45–52. Instead, a data file may be located elsewhere. Further, there are no disclaimers limiting the location of a data file to a directory, and the remainder of the paragraph supports this construction.

Defendants also argue the Court’s construction should include the directory limitation because PersonalWeb proposed a similar construction during an IPR. See Response at 30.

---

<sup>2</sup> For example, if A contains B and C, B is not required to be found within A. There is nothing preventing B from being located in D or E.

However, the PTAB did not adopt the directory limitation, and Defendants cite no authority that this Court should adopt a construction rejected by the PTAB. See Docket No. 89, Ex. G, at 11.

“Data file(s)” is “a named data item(s).”

**“file system”**

PersonalWeb contends no construction is necessary. In the alternative, PersonalWeb proposes “A collection of directories. A directory is a collection of named files.” Defendants propose “a collection of directories, which each include a collection of named files.”

PersonalWeb contends the phrase requires no construction because it is readily understood. Brief at 18. In the alternative, PersonalWeb proposes a construction taken verbatim from the specification. See ‘791 Patent, at 5:44–49. PersonalWeb believes Defendants’ construction is improper because it contains the word “includes.” Brief at 19. PersonalWeb argues Defendants’ construction would make “directory” and “collection of named files” two different things, even though a directory is a collection of named files. *Id.* Defendants describe this as a mischaracterization of their construction. Response at 33. Defendants believe a directory contains only named files, and that the parties’ constructions are very similar. *Id.*

“File system” is readily understood without construction. The specification states that “a file system is a collection of directories. A directory is a collection of named files.” ‘791 Patent, at 5:45–46. This is the commonly understood meaning of the term, and a jury will be able to understand the term without construction. Defendants contend their construction tracks the language of the specification, but they fail to justify their inclusion of “each” in their construction. “Each” does not appear in the specification, and it would only add ambiguity to the term. See *id.* at 5:45–54.

“File system” requires no construction.

**“substantially unique identifier,” “data identifier,” and “True Name”<sup>3</sup>**

PersonalWeb proposes “an identity for a data item generated by processing all of the data in the data item, and only the data in the data item, through an algorithm.” Defendants propose “an identifier for access to a data item generated by processing all of the data in the data item, and only the data in the data item, through an algorithm that makes it substantially unique to that data.”

There are two substantive differences between the parties’ constructions. First, Defendants include the phrase “for access to.” Defendants contend accessing a data item is the key function of a substantially unique identifier. Response at 6. They assert that the Patents do not teach any embodiment allowing access to a data item without using a substantially unique identifier. Id. at 7. Further, Defendants cite a portion of the specification stating “the present invention” is used to “reference and access” data items. ‘791 Patent, at 6:17–18. Thus, Defendants believe the construction should include “access.” Response at 8. PersonalWeb counters that “access” is improper because “nothing in the specification limits the use of the identifier once it is generated.” Brief at 6. Instead, PersonalWeb contends a substantially unique identifier can be used for other purposes, such as determining whether a data item exists at a specific location. Id.

The second difference between the constructions is Defendants’ inclusion of “through an algorithm that makes it substantially unique to that data.” PersonalWeb contends there is no need for language “purporting to state how “unique” the identifier must be.” Brief at 6. PersonalWeb argues a substantially unique identifier is defined by how it is calculated, not by its level of

---

<sup>3</sup> The parties agreed in their briefing the Court should construe “data identifier” the same way it construes “substantially unique identifier.” See Brief at 11–12; Response at 9. At the hearing, the parties also stipulated that the Court should construe “True Name” the same way it construes the other two terms.

uniqueness. *Id.* In support, *PersonalWeb* cites a passage teaching that two different data items can have the same identifier value. See ‘791 Patent, at 13:27–34.

Defendants argue the issue of how unique an identifier must be is an issue of law the Court must decide during claim construction.<sup>4</sup> Response at 8. Defendants contend their construction accurately reflects that an algorithm cannot be “simply any algorithm.” *Id.* Instead, the algorithm must generate an identifier that is substantially unique for each data item. *Id.* Further, Defendants cite an embodiment teaching that a substantially unique identifier is “virtually guaranteed” to represent a specific data block. ‘791 Patent, at 12:54–59. Accordingly, Defendants believe the construction of “substantially unique identifier” must include some measure of uniqueness. Response at 9.

It would be inappropriate to include Defendants’ “for access to” language in the construction because the specification teaches embodiments where a substantially unique identifier is used for things other than access. For example, a substantially unique identifier can be used to determine whether two files are identical. See ‘791 Patent, at 15:25–29. Substantially unique identifiers can also locate, move, delete, or copy files. See *id.* at 39:1–6. Defendants do not cite any disavowal that would mandate one particular use of a substantially unique identifier over another. Given the number of different uses disclosed in the specification, it would be improper to include “for access to” in the construction.

The claim language also supports this construction; different claims teach different uses of substantially unique identifiers. Claim 35 recites a method “for determining whether a particular data item is present in a data processing system.” ‘791 Patent, Claim 35, at 43:42–43.

---

<sup>4</sup> Defendants seemed to retreat from this position during the hearing, stating they believed the exact level of “substantial uniqueness” required by the Claims was a factual issue for the jury. However, Defendants still believe the construction should include the phrase “through an algorithm that makes it substantially unique to that data.”

The Claim teaches determining a substantially unique identifier, then determining whether that identifier is included within a set of data items. *Id.* at 43:47–54. This is a locate function—the Claim uses a substantially unique identifier to locate a file. On the contrary, Claim 30 specifically relates to access. Claim 30 uses a substantially unique identifier to “access[] a data item in the system.” ‘791 Patent, Claim 30, at 42:66–67. If a substantially unique identifier was necessarily required to access a data item, this claim language would be superfluous. See *Versa Corp. v. Ag-Bag Int’l Ltd.*, 392 F.3d 1325, 1330 (Fed. Cir. 2004) (“The difference in meaning and scope between claims is presumed to be significant to the extent that the absence of such difference in meaning and scope would make a claim superfluous.”) (internal quotations omitted).

Claim 4 is also illustrative. Claim 4 depends from Claim 2, which depends from Claim 1. Claim 4 contains the limitation “access means for accessing a particular data item using the identifier of the data item.” ‘791 Patent, Claim 4, at 39:40–41. Neither Claim 1 nor Claim 2 contain any limitation about “accessing,” indicating a substantially unique identifier is not required to access a data item. See *Curtiss-Wright Flow Control Corp. v. Velan, Inc.*, 438 F.3d 1374, 1380 (Fed. Cir. 2006).

The proper construction must also include a reference to substantial uniqueness. PersonalWeb seems to be advocating a construction that requires no “uniqueness” at all. However, the claim language specifically recites “substantially unique” identifier. PersonalWeb’s construction would essentially read these two words out of the construction. Therefore, the construction must include a reference to substantial uniqueness.

No further construction of “substantially unique” is required. During the hearing, the Court inquired whether it needed to separately construe “substantially unique.” Both sides said

no. Both sides agreed the level of uniqueness is a function of the data size, so whether a particular identifier is substantially unique is a factual issue to be resolved by the jury.<sup>5</sup> This position is supported by the specification, which discusses uniqueness as a function of data size. See ‘791 Patent, at 13:20–45 (discussing “collisions” between data items).

“Substantially unique identifier,” “data identifier,” and “True Name” are “an identity for a data item generated by processing all of the data in the data item, and only the data in the data item, through an algorithm that makes the identifier substantially unique.”

### **“identifier”**

PersonalWeb argues the term does not require construction. Defendants Apple, HP, and NECAM argue the term does not require construction. EMC and VMware take no position on the term. Google contends the term should be construed the same as “substantially unique identifier.” NetApp and Amazon propose “a name or a substitute for a name for a data item for use in a computer network.”

PersonalWeb contends the word “identifier” is readily understood to mean “something that identifies” or “a value that identifies.” Brief at 3. PersonalWeb notes that several Defendants have not asked the Court to construe this term, which illustrates that it can be readily understood without further construction. *Id.* at 4. PersonalWeb further argues “identifier” should not be given the same construction as “substantially unique identifier” because “identifier” is used differently in the Claims. *Id.* PersonalWeb cites Claim 6 of the ‘310 Patent, which recites “wherein the identifier of each licensed content item is based at least in part of the function of at least some of the data comprising the licensed content item.” ‘310 Patent, Claim 6, at 38:16–18. PersonalWeb asserts that the words “at least some of the data” indicate an “identifier” can be

---

<sup>5</sup> This also addresses any concerns that the Court’s construction is circular by defining “substantially unique identifier” as “substantially unique.”



based on less than all the data comprising the licensed content form. Motion at 4. Thus, “identifier” and “substantially unique identifier” are not interchangeable. Id. at 5.

Additionally, PersonalWeb argues the construction should not include the phrase “for use in a computer network.” Id. at 7. Instead, PersonalWeb contends an identifier can be used in a computer network or on a single computer. Id. In support, it cites a portion of the specification teaching an embodiment using “one or more processors (or computers).” See ‘791 Patent, at 4:58–61. Thus, PersonalWeb believes NetApp and Amazon’s construction improperly imports a limitation from an embodiment into the Claims. Id. at 8.

NetApp and Amazon argue “identifier” as used in “substantially unique identifier” in the ‘791 Patent should be construed as “a name or a substitute for a name for a data item for use in a computer network.” Response at 16. Defendants base this construction on statements made by PersonalWeb during the IPR of the ‘791 Patent. Defendants contend PersonalWeb distinguished the ‘791 Patent from prior art by arguing the result of an algorithm in the reference did not serve as a name or substitute for a name in a computer network. See id. at 17. Based on this statement, Defendants assert that “PersonalWeb deliberately and unambiguously disclaimed any interpretation of “identifier” that is not a name or a substitute for a name.” Id. at 18.

At the hearing, it became clear the parties’ primary dispute focuses on “identifier” as used in Claims 1, 5, and 6 of the ‘310 Patent.<sup>6</sup> Those claims use “identifier” with no preceding modifier (such as “substantially unique” or “data”). Google seeks a construction that would give “identifier” and “substantially unique identifier” the same meaning. The language of the Claims indicates why Google’s position is improper.

---

<sup>6</sup> Claims 5 and 6 depend from independent Claim 1.

Claim 1 of the '310 Patent does not contain the phrase “substantially unique identifier.”

However, it includes a similar concept—the “content-based name.” A content-based name is:

Based at least in part on a function of at least some of the data which comprise the contents of the particular data item, wherein the function comprises a message digest function or a hash function, and wherein two identical data items will have the same content-based name.

'310 Patent, Claim 1, at 37:50–54. Later, the claim recites “ascertaining whether or not the content-based name for the particular data item corresponds to an entry in a database comprising a plurality of identifiers.” *Id.* at 37:56–59. In the context of this claim language, the term “identifiers” is used in a generic manner; “content-based name” is used in a manner more closely related to the “substantially unique identifier” recited in other claims. Thus, the claim language weighs against Defendants’ construction. Additionally, when the specification addresses a “substantially unique identifier,” it focuses on the “substantially unique” aspects of the term. This implies a mere “identifier” is something different. *Cf. Phillips*, 415 F.3d at 1314 (“The claim in this case refers to ‘steel baffles,’ which strongly implies that the term ‘baffles’ does not inherently mean objects made of steel.”).

Defendants cite no disavowal in the specification of the limited term “identifier.” At the hearing, Defendants cited various file history statements made during the '310 Patent’s prosecution. See Defendants’ Hearing Slides at 38–39. However, all of the cited statements referred to the “content-based name,” not the identifier. Thus, those statements are not disavowals of “identifier.” Further, during prosecution of the original parent application, the patentee used “identifier” in a manner clearly distinct from “substantially unique identifier:”

Using the present invention, a substantially unique identifier is determined for a data item, regardless of any other names (identifiers) the data item may have. Further the substantially unique identifier is determined for the data item,

regardless of any names (identifiers) or the contents of any other data or data items.

See 6:11-CV-655, Docket No. 89, Ex. B, at 10 (August 29, 1997 Amendment).

“Identifier” requires no further construction.

**“digital identifier” and “data item identifier”**

PersonalWeb contends “digital identifier” and “data item identifier” require no construction. Defendants contend the terms are synonymous with “substantially unique identifier,” so the Court should adopt whatever construction for these terms it adopts for substantially unique identifier.

The dispute over these terms is whether a digital identifier and a data item identifier must be determined by processing all the data in a data item, as Defendants contend, or less than all the data in a data item, as PersonalWeb contends. PersonalWeb bases its position on the Claim language. Claim 86 of the ‘310 Patent recites that a digital identifier is based “at least in part, on a given function of at least some of the bits in a particular sequence of bits.” ‘310 Patent, Claim 86, at 46:28–30 (emphasis added). Similarly, Claim 81 of the ‘096 Patent recites that a data item identifier is “based at least in part on the data comprising the first data item.” ‘096 Patent, Claim 81, at 44:60–61 (emphasis added). Thus, both digital identifiers and data item identifiers may be calculated using less than all of the data in a data item. Brief at 8–11.

PersonalWeb also argues Defendants’ construction would give different claim terms identical meanings. *Id.* at 9. The Claims use several different modifiers of the word “identifier”—for example, “substantially unique,” “data,” “digital,” and “data item”—and Defendants construction would eliminate any distinction between them. *Id.* PersonalWeb

believes using different modifiers indicates the patentee intended each term to have a different meaning. Id.

Defendants present several arguments in support of their construction. First, they argue the specification does not disclose an identifier consistent with PersonalWeb's construction. Response at 11. Instead, Defendants represent that every embodiment disclosed in the specification describes an identifier created by processing all the underlying data. See id. ("PersonalWeb does not, and indeed cannot, point to any such identifier or algorithm."). Defendants contend that for "data item identifier" and "digital identifier" to be supported by the specification, they must be construed the same as "substantially unique identifier." Id. at 12. Next, Defendants contend PersonalWeb should be constrained by arguments it made during an IPR of the '096 Patent. Id. Defendants allege PersonalWeb distinguished the '096 Patent from a prior art reference because of the '096 Patent's "substantially unique identifier." See id. at 13. However, the Claims of the '096 Patent never use the term "substantially unique identifier." Thus, PersonalWeb used "substantially unique identify" interchangeably with "data item identifier" and "digital identifier" before the PTAB. Id. Lastly, Defendants argue the patentee's lexicography requires Defendants' proposed construction. Id. at 14. Defendants contend "data item identifier" and "digital identifier" must have the same meaning as "data identifier," which PersonalWeb agrees is synonymous with "substantially unique identifier." Id. Thus, all of the terms must have the same meaning. Id. at 15.

As to the "at least" claim language cited to by PersonalWeb, Defendants further assert that the claim language does not support PersonalWeb's construction. In particular, Defendants point to claims using "True Name" and "data identifier" (terms all parties agree equate to "substantially unique identifier") that contain the same "at least" language PersonalWeb cites in

support of its constructions for “data item identifier” and “digital identifier.” In particular, Defendants point to:

- ‘539 Patent, Claim 8: “said data identifier being based at least in part on a second function of said multiple content-based part identifiers;”
- ‘539 Patent, Claim 21: “said first data identifier is based, at least in part, on a second given function of data comprising the plurality of segment identifiers;” and
- ‘442 Patent, Claims 13 and 38: “the True Name being based at least in part on a given function of the data.”

Defendants contend this demonstrates PersonalWeb’s argument is not controlling. Further Defendants argue the specification teaches that other information related to a data item can add additional levels of identification in combination with a True Name. See Response at 15–16 (citing ‘791 Patent, at 13:55–14:12). Defendants assert that a True Name is always based on all and only the data item, but additional levels of identification may also be used. Defendants contend the “at least in part” language of the Claim reflects that some claims allow a later step. See *id.* at 16.

PersonalWeb’s arguments regarding the “at least” language are not persuasive. Other claims use similar “at least” language with regard to “data item identifier” and “True Name.” Both of these terms have the same meaning as “substantially unique identifier.” Further, the fact that multiple terms have the same construction does not control—the parties themselves agreed “substantially unique identifier,” “data identifier,” and “True Name” all have the same meaning.

Using the patentee’s lexicography and the transitive property, it is clear “data item identifier” should be given the same construction as “substantially unique identifier.” First, the specification states that “data identifier” and “substantially unique identifier” are the same:

In the following, the terms “True Name”, “data identity” and “data identifier” refers to the substantially unique data identifier for a particular data item.

‘791 Patent, at 6:6–8. Second, “data” and “data item” are the same—“the terms ‘data’ and ‘data item’ as used herein refer to a sequence of bits.” Id. at 1:54–60. Thus, “data item identifier” carries the same meaning as “data identifier.” Therefore, “data item identifier” will have the same construction as “data identifier,” “True Name,” and “substantially unique identifier.”

“Digital identifier” should likewise get the same construction. The specification defines “data” as a sequence of bits. ‘791 Patent, at 1:54–55. Based on this definition, “data” and “digital” carry a common meaning because digital information is commonly known to be expressed as a series of bits. The specification also makes clear that “data identifier” refers to a “substantially unique data identifier.” See id. at 6:6–8. Thus, “digital identifier” and “substantially unique identifier” share a common meaning.

This construction is supported by the claims of the ‘096 Patent. The ‘096 Patent emphasizes the relationship of “digital identifier” to the other specific identifier terms. A “digital identifier” is also referred to in the Claims as a “digital data item identifier.” See ‘096 Patent, Claim 1, at 38:50–52 (“A digital identifier for the data item, said digital data item identifier being....”). Thus, the claim itself a ties “digital identifier” to a “data item identifier.” As noted above, “data item identifier” and “data identifier” also carry the same meaning. Thus, the ‘096 Claims provide further guidance suggesting a common meaning between the two terms.

This can be contrasted with the “identifier” claims addressed above. The claims in which “identifier” is used alone do not use “identifier” in connection with the portion of the claim that references the results of the application of the function to two identical data items. Instead, that portion of those Claims reference the “content-based name.”

Thus, based on the specification as a whole and the claims themselves, “digital identifier” has the same construction as “data item identifier,” “data identifier,” “True Name,” and “substantially unique identifier.”

**“sufficient number of copies”**

PersonalWeb contends no construction is necessary. In the alternative, PersonalWeb proposes “application-specific criteria regarding the number of copies required.” Defendants propose “a preset number of copies greater than or equal to 2 of different locations in which a file is stored.”

PersonalWeb contends the phrase has no special meaning and requires no further construction. Brief at 19. If the phrase is construed, PersonalWeb believes its construction is supported by the specification, which teaches that the mechanism used to ensure files are available “depends on application-specific migration/archival criteria.” See ‘791 Patent, at 26:22–28. Defendants counter that their construction is consistent with the intrinsic record. Response at 34. Defendants contend both Claim 1 of the ‘662 Patent and the specification indicate copies must be stored at multiple locations. *Id.* Defendants also argue the specification passage cited by PersonalWeb supports their construction. *Id.* Defendants assert that migrating or archiving a data item “necessarily requires at least two copies of the data item.” *Id.*

The claim language itself is significantly clearer and more straightforward than Defendants’ proposed construction. Defendants’ proposed construction, intended to clarify the Claim, does the opposite. It is not readily apparent what “greater than or equal to 2 of different locations in which a file is stored” actually means. On the contrary, the claim language itself is comprehensible without further construction. A sufficient number of copies must be present “at one or more other storage locations in a file system,” and there must be enough copies to “satisfy

said predetermined degree of redundancy for said file system.”<sup>7</sup> See ‘662 Patent, Claim 1, at 39:15–19.

Another flaw in Defendants’ construction is it requires copies to be stored at multiple locations. Defendants do not cite any disavowal language from the specification stating that multiple copies at a single location are not redundant.<sup>8</sup> Further, Defendants’ proposal seems to contradict the Claim language. The Claim discusses copies “present at one or more other storage locations.” See *id.* at 39:16–17, 39:23–24. The “one or more” language indicates copies may be stored at a single location. Defendants also cite a portion of the specification to demonstrate that redundancy requires multiple locations, but even that language is permissive. See *id.* at 34:31–33 (“A DP system employing the present invention can be....”) (emphasis added). Thus, Defendants’ “different locations” language is improper.

“Sufficient number of copies” requires no construction.

#### **“licensed” and “unlicensed”**

PersonalWeb contends the terms do not require construction. In the alternative, PersonalWeb proposes “with license” and “without license.” Defendants propose “having a license to content stored within a requested file.”

PersonalWeb challenges two different aspects of Defendants’ construction. First, PersonalWeb argues Defendants’ construction is circular because it uses “license” to define “licensed.” Brief at 20. Second, PersonalWeb asserts that licensed content is not limited to content stored within a requested file. *Id.* Instead, licensed content could simply refer to stored content, regardless of whether there was any request for the content. *Id.* at 20–21. PersonalWeb

---

<sup>7</sup> Defendants’ proposed construction includes the phrase “a preset number of copies.” This phrase is unnecessary because the Claim already addresses a “predetermined degree of redundancy.”

<sup>8</sup> Defendants argue “copies at a single location are not redundant” without citation. See Response at 34.



also argues substantially unique identifiers can be used for passive enforcement of licensed content. *Id.* at 21. With passive enforcement, a user may be classified as licensed or unlicensed without ever making a request for a file. *Id.*

Defendants argue PersonalWeb’s construction rewrites the Claims to require only general authorization to access the entire system, rather than authorization to access specific files. Response at 35. Defendants contend all authorization is done on a content-specific basis. *Id.* For example, Claim 1 of the ‘442 Patent recites that “a copy of the requested file is only provided to licensed parties.” ‘442 Patent, Claim 1, at 39:52–53. Similarly, Claim 3 references an “unlicensed copy of the data file.” ‘442 Patent, Claim 3, at 39:60. Defendants argue these references to the data file indicate authorization is done on a file-by-file basis. Response at 36. Further, they assert that the Patents never refer to the “mere ability to access the system or data within it.” *Id.* Therefore, PersonalWeb’s contrary construction is improper. *Id.*

The parties have two distinct disputes regarding these terms. First, they debate whether the license must be to the content of a file or to the system as a whole. However, there is no need to resolve this dispute globally because all of the cited Claims reference a license to a file. See, e.g., ‘442 Patent, at 39:51–53 (Claim 1) (“Wherein a copy of the requested file is only provided to licensed parties.”); 39:60 (Claim 3); 40:19–20 (Claim 6); 40:34–36 (Claim 7); 40:58–59 (Claim 8); 41:22–24 (Claim 10). While there must be a license to a file, there is no further restriction on what type of license is necessary. For example, there is no disclaimer stating that a license must be to a specific file, or whether it could be to a series of files, or whether it could be even broader than that. Further, there is no language in the specification mandating a particular level of specificity of license. There is also no limiting language about the nature in which a

license must be issued. Accordingly, while a party must have a license to access a particular file, there is no restriction on precisely how that license grants access to the file.

Second, the parties debate whether the licensed file must be “requested.” This answer comes directly from the specification. The specification discloses an “audit” embodiment where license status is determined without a request for a specific file. ‘791 Patent, at 32:27–28. In the audit embodiment, a system administrator can determine employees’ license status for various files without actually requesting specific files. *Id.* at 28–48. This embodiment indicates a licensed file need not be “requested.”

“Licensed” and “unlicensed” require no further construction.

**“distributing a set of data files across a network of servers”**

PersonalWeb contends no construction is necessary. In the alternative, PersonalWeb proposes “storing a set of data files on various servers in a network.” Defendants propose “delivering a set of data files to two or more servers in a network of servers.”

The parties debate whether a “network of servers” includes a single server. See Response at 22. PersonalWeb takes the position that it does. PersonalWeb contends Defendants’ construction reads out an embodiment from the specification. Brief at 21. PersonalWeb argues the specification discloses two servers in a peer-to-peer relationship. See ‘791 Patent, at 5:5–16. Under this configuration, data is delivered from one server to a second server. Brief at 22. Because there is only one recipient server, data is not delivered to “two or more servers,” as required by Defendants’ construction. *Id.*

Defendants challenge two aspects of PersonalWeb’s non-construction. First, Defendants argue a “network of servers” must be two or more servers. Response at 37. Defendants contend that during the ‘280 Patent’s prosecution, the patentee used the Mirror True File mechanism to

overcome a written description rejection. *Id.* The mechanism is “used to ensure that files are available in alternate locations in mirror groups or achieved on archival servers.” ‘791 Patent, at 26:22–24. Defendants argue this redundancy cannot exist unless copies of files are stored on multiple servers. Response at 37. Second, Defendants believe PersonalWeb improperly substituted “storing” for “distributing.” *Id.* at 38. Defendants argue storing and distributing are distinct concepts, so “storing” is not a proper construction. *Id.* at 39.

Defendants’ construction in essence requires at least three servers—one delivering and at least two receiving. However, the specification explicitly discloses two-server embodiments. See ‘791 Patent, at 5:5–7. It even teaches three different two-server configurations. *Id.* at 5:5–16. It would be improper to require a three-server configuration when the specification unambiguously discloses three distinct two-server configurations. Further, there is no limiting language in the Claims. The Claims merely recite “distributing a set of data files across a network of servers,” with no further restrictive language. See ‘280 Patent, at 40:21 (Claim 10); 42:7 (Claim 25); 43:4 (Claim 31). Thus, there is nothing in the Patent indicating a two-server system cannot be a “network” of servers.

Defendants’ main argument seems to be that the patentee did not rely on the two-server embodiment to overcome a written description rejection, so it disclaimed a peer-to-peer server configuration. See Response at 37. However, this is not an express disavowal of claim scope. See *Omega Eng’g, Inc. v. Raytek Corp.*, 334 F.3d 1314, 1324 (Fed. Cir. 2003) (“We have, however, declined to apply the doctrine of prosecution disclaimer where the alleged disavowal of claim scope is ambiguous.”); *York Prods., Inc. v. Cent. Tractor Farm & Family Ctr.*, 99 F.3d 1568, 1575 (Fed. Cir. 1996) (“Unless altering claim language to escape an examiner rejection, a patent applicant only limits claims during prosecution by clearly disavowing claim coverage.”). Under

Defendants' argument, every embodiment not cited to overcome a prosecution rejection would be disclaimed. They cite no authority to support this proposition,<sup>9</sup> and their cited prosecution history statements do not rise to the level of "clear disavowal." See *id.*

There is no requirement that a network of servers deliver files to two or more servers. "Distributing a set of data files across a network of servers" requires no further construction.

**"access means for accessing a particular data item using the identifier of the data item"**

The parties agree the function is "accessing a particular data item using the identifier of the data item." PersonalWeb contends the corresponding structure is "at least one processor ('791 at 7:62–63), programmed in accordance with one or more of the mechanisms described at '791 33:51–34:32 (Make True File Local); 24:45–58; 17:10–45 (Request True File); 16:10–34 (Realize True File from Location); 21:52–66 (Open File); 25:25–38 (Acquire True File); and 35:38–41, 51–61 (Locate Remote File)." Defendants contend the corresponding structure is "the processor of Fig. 1(b), storing the True File Registry and programmed to execute the "Make True File Local" primitive mechanism depicted in Figures 17(a) and 17(b)."

PersonalWeb argues Defendants improperly include the True File Registry in their corresponding structure. Brief at 22. PersonalWeb argues the True File Registry does not perform the recited function. *Id.* at 23. Instead, it is merely an input to the mechanism that performs the function. *Id.* PersonalWeb also argues Defendants' construction is unnecessarily narrow. *Id.* PersonalWeb argues many mechanisms, "including primitive mechanisms, operating system mechanisms, remote mechanisms, background mechanisms, and extended mechanisms"

---

<sup>9</sup> Defendants cite Lucent for the proposition that "there is no rule of claim construction requiring that every disclosed embodiment be encompassed within the scope of a given claim." *Lucent Techs., Inc. v. Gateway, Inc.*, 525 F.3d 1200, 1215–16 (Fed. Cir. 2008); see Response at 38. This is a separate issue because Defendants cite no authority to support their prosecution disavowal argument.

can perform the recited function. *Id.* Therefore, Defendants’ construction to the contrary is improper. *Id.*

Defendants contend PersonalWeb did not sufficiently tie its disclosed structure to the underlying function. Response at 19. Defendants argue the proper test is not whether a structure is capable of performing the recited function but whether a structure is clearly associated with the recited function. *Id.* Defendants also argue their construction is supported by recent proceedings before the PTAB. *Id.* at 20. Defendants represent that the PTAB construed the term to be “a processor programmed according to steps S292 and S294 illustrated in Figure 17(a)” in the ‘791 Patent’s IPR. *Id.* Defendants contend this indicates the True File Registry is necessary to implement the claimed functionality. *Id.*

The parties have three distinct disputes. First, Defendants contend the processor should be limited to the processor of Figure 1(b). The specification discloses in Figure 1(a) a system having generic processors. Figure 1(b) provides a more detailed example of a processor. See ‘791 Patent, at 4:58–62. Defendants do not cite any disclosure suggesting all references to a “processor” are limited to the processor shown in Figure 1(b). Because the specification discloses general processor blocks and a more specific processor block, it would be inappropriate to limit “processor” to the more specific disclosure of Figure 1(b). Accordingly, the proper “processor” is the generic processor disclosed at 7:62–63.

Second, the parties debate whether the True File Registry must be included in the corresponding structure. The True File Registry is a look-up table containing a list of True Names. See ‘791 Patent, at 8:27–34, 12:28–32, 14:43–47. Each mechanism cited by PersonalWeb uses the True File Registry to locate a True Name, and the processor is programmed to go to the True File Registry to find True Names. Thus, the True File Registry is

central to the algorithm—it is the input to the algorithm needed to make the algorithm work. However, there is no restriction in the Patent requiring the processor of the access means to “store” the True File Registry. See ‘791 Patent, at 5:5–16 (describing multi-processor embodiments), 5:65–6:5, 7:60–8:9 (describing sharing across processors). Thus, the processor recited in the structure may satisfy the claim language by accessing a True File Registry stored in a different location.

Third, the parties debate which mechanisms should be included in the structure. Defendants contend the corresponding structure should be limited the Make True File Local mechanism depicted in Figures 17(a) and 17(b), while PersonalWeb recites a series of different mechanisms in its proposed construction.

Defendants’ construction is overly narrow—each of the mechanisms found in PersonalWeb’s construction provides a way to access a data item using an identifier. The Make True File Local mechanism (33:51–34:32) “provides transparent access to any data item by reference to its data identity.” The Request True File mechanism (24:45–58) allows a remote processor to request a copy of a True File from a local processor using its True Name. Both sides agree the Make True File Local mechanism (17:10–45) is part of the structure. The Realize True File from Location mechanism (16:10–34) allows a user to attempt to make a local copy of a True File, given its True Name. The structure disclosed at 21:52–66 allows a party to make a True File segment available locally. The Acquire True File mechanism (25:25–38) allows a remote processor to force a local processor to make a copy of a specified True File. Finally, the mechanism disclosed at 35:38–41, 51–61 allows a remote processor to search for True Names on other remote processors. All of these structures provide a way to access a data item using the identifier, so all should be included in the corresponding structure.

“Access means for accessing a particular data item using the identifier of the data item” has a function of “accessing a particular data item using the identifier of the data item.” It has a corresponding structure of “at least one processor (7:62–63), programmed to use a True File Registry and a True Name in accordance with one or more of the mechanisms described at ‘791 33:51–34:32; 24:45–58; 17:10–45; 16:10–34; 21:52–66; 25:25–38; 35:38–41, 51–61.’”

**“data associating means for making and maintaining for a data item in the system, an association between the data item and the identifier of the data item”**

The parties agree the function is “making and maintaining, for a data item in the system, an association between the data item and the identifier of the data item.” PersonalWeb contends the corresponding structure is “at least one processor (‘791 at 7:62–63), programmed in accordance with one or more of the mechanisms described at ‘791 8:27–34; 9:36–37; 14:40 to 15:4; 35:46–48; 38:54–65.” Defendants contend the corresponding structure is “the processor of Fig. 1(b), storing the True File Registry and programmed to execute the “Assimilate Data Item” primitive mechanism depicted in Figure 11.”

PersonalWeb makes similar arguments for this term as it did for the previous term. In particular, it contends the True File Registry is not part of the corresponding structure because it is merely an input to other mechanisms that perform the recited function. Brief at 24. Additionally, PersonalWeb argues Defendants’ construction limiting the corresponding structure to the Assimilate Data Item mechanism is unnecessarily narrow. *Id.* PersonalWeb contends the specification does not require the limited configuration Defendants propose. *Id.* Defendants counter that PersonalWeb is seeking a construction already rejected by the PTAB during the ‘791 Patent’s IPR. Response at 21. Defendants argue the PTAB’s construction includes looking for an

entry in the True File registry and determining whether a True Name exists within the True File registry. *Id.* Thus, Defendants’ reference to Figure 11 in their construction was proper. *Id.*

The parties’ arguments regarding the True File Registry and the appropriate processor are nearly identical to those made with regards to the “access means” limitation, and the same analysis applies to “data associating means” as well. Accordingly, that portion of the corresponding structure is identical to the “access means” structure.

The parties also debate which mechanisms should be included in the corresponding structure. Both sides agree the Assimilate Data Function mechanism (14:40–15:4) is relevant. The Assimilate Data Function discloses a mechanism for integrating a data item into a file system. ‘791 Patent, at 14:41–43. This is accomplished by creating a True Name and adding it to the True File Registry, which establishes a new association between a True File and its corresponding True Name. Thus, the Assimilate Data Function discloses corresponding structure. See *id.* at 14:51–67.

The remainder of PersonalWeb’s disclosures recite mechanisms that do not perform the claimed function of making and maintaining an association. 8:27–34 and 9:36–37 merely describe the True File Registry and True Names. The Assimilate Data Function uses these mechanisms, but they do not independently disclose corresponding structure. Neither 35:46–48 nor 38:54–65 provide sufficient independent structure because both reference back to the Assimilate Data Function. Both citations teach mechanisms where the first step is assimilating data items, so these mechanisms simply refer one skilled in the art back to the Assimilate Data Function mechanism.

“Data associating means for making and maintaining, for a data item in the system, an association between the data item and the identifier of the data item” has a function of “making



and maintaining, for a data item in the system, an association between the data item and the identifier of the data item.” The corresponding structure is “at least one processor (7:62–63), programmed to use a True File Registry and a True Name in accordance with one or more of the ‘Assimilate Data Item’ primitive mechanisms depicted in Figure 11 and described at 14:40–15:4.”

**“existence means for determining whether a particular data item is present in the system, by examining the identifiers of the plurality of data items”**

The parties agree the function is “determining whether a particular data item is present in the system, by examining the identifiers of the plurality of data items.” PersonalWeb contends the structure is “at least one processor (‘791 at 7:62–63), programmed in accordance with one or more of the mechanisms described at ‘791 Fig. 14 (S260); Fig. 11 (S232); 15:54–57; 14:53–56 (beginning with “Next”); 16:25–45; 33:6–7; co. 8:27–32; Fig. 28; 23:53 to 24:13.” Defendants contend the structure is “the processor of Fig. 1(b), storing the True File Registry and programmed to execute the mechanism depicted in step S232 of Figure 11 and the “Locate Remote File” primitive mechanism depicted in Figures 16(a) and 16(b).”

PersonalWeb argues Defendants’ construction is improper for the same reasons as their previous two constructions—Defendants improperly seek to import the True File Registry into the structure. Brief at 25. PersonalWeb also believes Defendants’ construction fails because it requires the Locate Remote File primitive mechanism depicted in Figures 16(a) and 16(b). *Id.* at 26. PersonalWeb contends the specification discloses many other structures beyond the ones disclosed in Figures 16(a) and 16(b). *Id.*

Defendants contend the Locate Remote File shown in Figures 16(a) and 16(b) is a required structure because it is the only thing in the specification that determines whether the

data item is anywhere in this system. Response at 23. Defendants also contend the True File Registry is a required structure. Id. The True File Registry stores location information about True Files. Id. Thus, it is necessary to examine “the identifiers of the plurality of data items.” Id. Defendants describe PersonalWeb’s arguments to the contrary as “nonsensical[.]” Id. Defendants argue a structure could not examine the identifiers without first knowing the identifiers to be examined. Id. Accordingly, the True File Registry is a required structure even if it is merely an input. Id.

The parties’ arguments regarding the True File Registry and the appropriate processor are nearly identical to those made with regards to the two previous means-plus-function limitations, and the same analysis applies to “existence means” as well. Accordingly, that portion of the corresponding structure is identical to the “access means” and “data associating means” structure.

The parties’ biggest dispute is the breadth of “the system.” The specification describes multiple methods to determine whether a file exists in the system. See ‘791 Patent, at 15:54–56 (search for True Name locally), 16:38–17:10 (Locate Remote File mechanism). In particular, there are mechanisms to search both locally (within a device’s own storage) and remotely (i.e. to outside storage or other processors). Defendants take the position that the function requires searching both locally and remotely, while PersonalWeb argues local and remote searching are not both required.<sup>10</sup>

Defendants do not cite any language from the Claims or the specification requiring both local and remote searching. The claim language only recites “in the system,” and there is no

---

<sup>10</sup> NetApp individually argues “the device must look not just remotely...but also locally within its own storage.” See Response at 24. While NetApp makes this argument on its own behalf, every Defendant’s proposed construction seems to require both local and remote searching.

further discussion addressing the scope of the “system.” See ‘791 Patent, Claim 1, at 39:21–23. With their construction, Defendants are attempting to add an additional limitation to the Claim that the existence means must determine every instance the data item is found anywhere within the system. However, the Claim is not that specific—it just generically refers to determining whether a data item is present in “the system.” See *id.* “The system” could refer to something less than the entire network. Additionally, Defendants’ inclusion of “and” would read out embodiments with a single processor. See *id.* at 4:58–61 (describing a one-processor embodiment). If there is only one processor, it would be impossible for the one processor to search remotely. Accordingly, the function is not required to search both locally and remotely.

Most of the passages cited by PersonalWeb disclose means for determining whether a data item is present in the system. Figure 14 (S260) teaches a mechanism for confirming “that the True Name exists locally by searching for it in the True Name registry or local directory extensions table.” See *id.* at 15:54–57. Figure 11 (S232) discloses a mechanism to search for a True Name within the True File Registry. *Id.* at 14:53–56. Figure 28 describes a mechanism allowing “a remote processor to determine whether the local processor contains a copy of a specific True File.” *Id.* at 23:53–55. This mechanism is described in further detail at 23:53–24:13.

PersonalWeb’s proposed construction contains three passages that do not describe the locate process: 16:25–37 merely describes the True File Registry; 33:6–7 does not describe the locate process; and 8:27–32 provides background information on the True File Registry. Therefore, none of these passages provide corresponding structure to “existence means.” Additionally, the Locate Remote File mechanism cited by Defendants (but not PersonalWeb) is corresponding structure. See *id.* at 16:38–17:18. The Locate Remote File mechanism, like its

name implies, teaches how to locate a file or data item from a remote source of True Files. Id. at 16:39–40. This mechanism is also disclosed in Figures 16(a) and 16(b). Id. at 16:43–45.

“Existence means for determining whether a particular data item is present in the system, by examining the identifiers of the plurality of data items” has a function of “determining whether a particular data item is present in the system, by examining the identifiers of the plurality of data items.” The corresponding structure is “at least one processor (7:62–63), programmed to use a True File Registry and a True Name in accordance with one or more of the mechanisms described at ‘791 Fig. 14 (S260); Fig. 11 (S232); 15:54–57; 14:53–56 (beginning with “Next”); Fig. 28; 23:53–24:13; Figures 16(a) and 16(b); 16:38–17:18.”

**“local existence means for determining whether an instance of a particular data item is present at a particular location in the system, based on the identifier of the data item”**

The parties agree the function is “determining whether an instance of a particular data item is present in the system, by examining the identifiers of the plurality of data items.” PersonalWeb contends the corresponding structure is “at least one processor (‘791 at 7:62–63), programmed in accordance with one or more of the mechanisms described at ‘791 Fig. 14 (S260); ‘791 at 15:54–57; 16:10–34; Fig. 28; 23:52 to 24:13.” Defendants contend the structure is “the processor of Fig. 1(b), storing the True File Registry and programmed to execute the “Locate True File” remote mechanism depicted in Figure 28.”

PersonalWeb argues Defendants’ construction is improper because it includes the True File Registry. Brief at 28. PersonalWeb contends the True File Registry is merely an input to the mechanisms that perform the recited function, and thus is not required structure. Id. PersonalWeb also asserts that Defendants’ construction improperly limits the structure to a particular configuration, even though many other configurations are disclosed in the Patents. Id. at 29.

Defendants contend PersonalWeb’s construction conflates independent Claim 1 of the ‘791 Patent with dependent Claim 2. Response at 25. The phrase in question appears in Claim 2, which depends from Claim 1.<sup>11</sup> See ‘791 Patent, Claim 2, at 39:25–28. Defendants argue PersonalWeb’s construction fails to account for the differences between Claim 1’s “existence means” and Claim 2’s “local existence means.” Response at 25 (emphasis added). Defendants argue the word “local” corresponds to the ability to respond to requests from other apparatuses in the system. *Id.* Thus, their inclusion of the Locate True File mechanism is proper. *Id.* Defendants also assert that their construction is proper because it was adopted by the PTAB, which expressly referenced the True File Registry in its discussion of the required structure. See *id.* at 26.

All of the discussion regarding the True File Registry and the appropriate processor from the previous means-plus-function terms is applicable here. Accordingly, that portion of the corresponding structure is identical to the corresponding structure for the previous terms.

Defendants contend the corresponding structure should be limited to the Locate True File mechanism depicted in Figure 28.<sup>12</sup> Defendants believe the local existence means functionality can occur when a processor searches remote sources to find a True File. However, they cite no disclaimer that a processor searching within its own files could not also satisfy the local existence means functionality. The Claim refers to “an instance of a particular data item” at a “particular location.” See ‘791 Patent, Claim 2, at 39:25–27. Nothing in this sentence requires the “particular location” to be a remote location. Thus, the corresponding structure is not limited to the Locate True File mechanism.

---

<sup>11</sup> Claim 3 depends from Claim 2, which depends from Claim 1.

<sup>12</sup> There is no dispute that the Locate True File mechanism should be part of the structure—it is also included in PersonalWeb’s proposed construction.

The majority of the other mechanisms cited by PersonalWeb also disclose “local existence means” functionality. Figure 14 (S260) discloses a mechanism to determine whether a True Name exists within a registry. See *id.* at 15:54–57. Figure 28, as discussed above, discloses the Locate True File mechanism. There is further discussion of the Locate True File at 23:53–24:13. However, 16:10–34, cited by PersonalWeb, is not part of the corresponding structure. The Realize True File from Location mechanism “is used to try to make a local copy of a True File,” not to determine whether a True File exists at a particular location. See *id.* at 16:11–12. Therefore, it is not appropriate corresponding structure.

“Local existence means for determining whether an instance of a particular data item is present at a particular location in the system, based on the identifier of the data item” has a function of “determining whether an instance of a particular data item is present in the system, by examining the identifiers of the plurality of data items.” The corresponding structure is “at least one processor (7:62–63), programmed to use a True File Registry and a True Name in accordance with one or more of the mechanisms described at ‘791 Fig. 14 (S260); 15:54–57; Fig. 28; 23:53–24:13.’”

**“identity means for determining for any of a plurality of data items present in the system, a substantially unique identifier, the identifier being determining using and depending on all of the data in the data item and only the data in the data item, whereby two identical data items in the system will have the same identifier”**

The parties debate both the function and structure of this term. For the function, PersonalWeb proposes “determining, for any of a plurality of data items present in the system, a substantially unique identifier.” Defendants propose “determining, for any of a plurality of data items present in the system, a substantially unique identifier, the identifier being determined using and depending on all of the data in the data item and only the data in the data item,

whereby two identical data items in the system will have the same identifier.” For the corresponding structure, PersonalWeb proposes “at least one processor (‘791 at 7:62–63), programmed to Calculate the True Name mechanism in accordance with ‘791 12:54 to 13:19 and 14:1–39.” Defendants propose “the processor of Fig. 1(b), programmed to execute the “Calculate True Name” primitive mechanism depicted in Figures 10(a) and 10(b), where the MD function is one of the MD4, MD5 and SHA functions.”

Regarding the function, PersonalWeb contends Defendants include limitations that are not a part of the determining process. Brief at 27. In particular, PersonalWeb argues the results of “determining” and “identifier” are not a part of the determining process itself, so they do not need to be included in the function. *Id.* Defendants argue their proposed function is identical to the function PersonalWeb proposed (and the PTAB adopted) in the ‘791 Patent’s IPR. Response at 27. Further, Defendants contend it is improper as a matter of law to abbreviate the function because it must match the claim language. *Id.*

Regarding the structure, PersonalWeb believes Defendants’ construction improperly limits the structure to a particular configuration. Brief at 27. PersonalWeb asserts that the Calculate True Name mechanism consists of a “family of functions,” so it should not be limited to the exemplary functions listed in the specification. See ‘791 Patent, at 13:10. Defendants believe the structure should be limited to the three examples disclosed in the specification. Response at 28. Defendants argue these are the only three examples given, so the MD4, MD5, and SHA algorithms must be part of the corresponding structure. *Id.* Additionally, Defendants contend PersonalWeb identified these three algorithms as the structure in arguments to the PTAB. *Id.* at 28–29. Therefore, PersonalWeb should be bound by its specific prosecution disclaimers. *Id.* at 29.

PersonalWeb’s proposed function is more appropriate than Defendants’ proposed function. PersonalWeb’s function is more succinct—it removes discussion about the requirements of a substantially unique identifier—and it would be easier for a jury to apply. See *Lockheed Martin Corp. v. Space Sys./Loral, Inc.*, 324 F.3d 1308, 1319 (Fed. Cir. 2003) (noting the phrase “means for” is typically followed by the recited function and claim limitations). Further, excluding Defendants’ proposed language has no impact on the Claim scope. Regardless of whether the phrase is included in the function, the identifier must still be determined using all of data in the data item and only the data in the data item, and two identical data items must have the same identifier. This limitation does not disappear if it is not included in the proposed functionality. It must still be met to satisfy the Claim. The function for “identity means” is “determining, for any of a plurality of data items present in the system, a substantially unique identifier.”

Regarding the corresponding structure, the parties first debate the processor. For the same reasons articulated above, it would be improper to limit “processor” to the one disclosed in Figure 1(b). Accordingly, the appropriate processor is the processor identified at 7:62–63.

There are two remaining flaws in Defendants’ proposed structure. First, it requires a system to use both the mechanism disclosed in Figure 10(a) and the mechanism disclosed in Figure 10(b). Figure 10(a) discloses a mechanism for calculating a True Name for a “simple data item.” See ‘791 Patent, at 14:4–6. A data item is simple if its size is smaller than the specified threshold. *Id.* Figure 10(b) discloses a mechanism for calculating a True Name for a “compound data item.” *Id.* at 14:12–13. A compound data item is larger than the specified threshold. *Id.* Figure 10(b) discloses first determining whether a data item is simple or compound, and if it is compound, performing a series of other steps. See *id.* at 14:13–31. However, an embodiment



containing only simple data items would not need or use the structure disclosed in Figure 10(b) relating to compound data structures. See Fig. 10(b). The structure of Figure 10(a) could exist independent of Figure 10(b). Accordingly, the Claim does not require using both Figure 10(a) and 10(b).

Second, Defendants’ construction limits the MD function to the MD4, MD5, and SHA functions. The ‘791 Patent describes the MD function by its characteristics, not its title. See *id.* at 12:61–13:9 (listing five characteristics of MD functions). It does not limit the MD function to the three exemplary functions disclosed in the Patent. Instead, the specification clarifies that those functions are non-exhaustive. The “family” of MD functions “include[s]” MD4, MD5, and SHA, but it is not limited to those functions. See *id.* at 13:10–11 (stating that functions “with the above properties” are MD functions). Thus, Defendants’ attempt to limit the corresponding structure to the MD4, MD5, and SHA functions is improper.

“Identity means for determining, for any of a plurality of data items present in the system, a substantially unique identifier, the identifier being determined using and depending on all of the data in the data item and only the data in the data item, whereby two identical data items in the system will have the same identifier” has a function of “determining, for any of a plurality of data items present in the system, a substantially unique identifier.” Its corresponding structure is “at least one processor (7:62–63), programmed to perform the Calculate the True Name mechanism in accordance with ‘791 12:54–13:19 and 14:1–39.”

## INDEFINITENESS

### “data item”

Defendant Facebook individually moved for summary judgment of indefiniteness of Claim 30<sup>13</sup> of the ‘791 Patent. See 6:12-CV-662, Docket No. 66 (“Facebook Motion”). Claim 30 recites:

A method of identifying a data item present in a data processing system for subsequent access to the data item, the method comprising:

determining a substantially unique identifier for the data item, the identifier depending on and being determined using all of the data in the data item and only the data in the data item, whereby two identical data items in the system will have the same identifier; and

accessing a data item in the system using the identifier of the data item.

‘791 Patent, Claim 30, at 42:58–67 (emphasis added). Facebook argues Claim 30 is insolubly ambiguous because “a data item” in the final line lacks antecedent basis. Facebook Motion at 5. The Claim recites determining a substantially unique identifier for “the data item,” then recites accessing “a data item” using that identifier. *Id.* Facebook asserts that these statements are inconsistent because “a data item” in the final line is broader than “the data item” described in the substantially unique identifier limitation. *Id.* Facebook argues the Claim is invalid unless the final limitation refers to the same data item identified earlier in the Claim. *Id.* at 6.

PersonalWeb argues the Claim is clear and unambiguous. 6:12-CV-662, Docket No. 90 (“Facebook Response”) at 4. The Claim specifically states that two identical data items in a system will have the same identifier. *Id.* Accordingly, a single substantially unique identifier can be used to access multiple data items. *Id.* PersonalWeb contends the Claim is logically written because it teaches determining an identifier, then using the identifier to identify a data item. *Id.*

---

<sup>13</sup> Facebook also argues dependent Claims 31, 32, and 40 are indefinite. All three depend from Claim 30.

PersonalWeb asserts that “a data item” is proper because a single identifier can identify multiple data items. *Id.* PersonalWeb also argues Facebook failed to present any evidence that one of ordinary skill would not understand the scope of the Claim. *Id.* at 5.

PersonalWeb’s interpretation of the Claim is reasonable. In the specification, there are numerous instances where multiple copies of the same data item are kept. See ‘791 Patent at 3:62–65, 4:6–8, 26:21–40, 34:51–35:11. Facebook argues the Claim language does not suggest multiple identical data items can be accessed through the same identifier, but this runs afoul of the plain language of the Claim. Claim 30 unambiguously states that “two identical data items in the system will have the same identifier.” ‘791 Patent, Claim 30, at 42:64–65. Thus, the “same identifier” must be able to access “two identical data items.”

The last step of the Claim requires the identifier used to access a data item to be the same identifier determined in the first step. There is no additional requirement that the exact same data item identified in the first step be the data item accessed in the final step. See *Microprocessor Enhancement Corp. v. Texas Instruments Inc.*, 520 F.3d 1367, 1375 (Fed. Cir. 2008) (stating two terms with the same antecedent need not have the same meaning). Defendants contend such a construction would leave one skilled in the art guessing as to what data item is actually accessed. However, it is a reasonable interpretation that the data item accessed is any data item having the same identifier (i.e. any of the multiple identical data items). See *id.* at 1376 (finding the scope of a claim term “readily apparent” from its context).

Facebook’s motion for summary judgment of indefiniteness (6:12-CV-662, Docket No. 66) is **DENIED**.

**“existence means” and “local existence means”**

Defendant HP individually moves for summary judgment of indefiniteness of Claims 1, 2, and 3 of the ‘791 Patent. See 6:11-CV-683, Docket No. 164 (“HP Motion”). All three claims are written in means-plus-function form. HP contends all three are indefinite for lack of sufficient structure. In particular, HP believes there is no structure to support “existence means” in Claim 1 and “local existence means” in Claims 2 and 3.

HP makes three distinct arguments in support of its motion. First, it argues the specification does not disclose step-by-step algorithms or procedures that are clearly linked or associated with the claimed functions. *Id.* at 8. HP argues there is no disclosure indicating how to determine whether a substantially unique identifier exists in the system. *Id.* Second, HP argues the specification does not disclose an algorithm. *Id.* at 9. HP asserts that the structures identified by PersonalWeb come from unrelated sections of the specification, and there is no logical way to correlate the structures. *Id.* Additionally, HP believes all recited structures require the True File Registry, but the TFR is not part of PersonalWeb’s disclosures. *Id.* at 10.

Third, HP contends PersonalWeb’s algorithms do not perform the claimed functions. *Id.* at 11. In particular, HP asserts that the mechanisms recited by PersonalWeb cannot determine whether a data item is present in the system. *Id.* HP argues none of the structures are capable of comprehensibly searching the entire system. *Id.* Instead, the specification only teaches searching a specific registry. *Id.* at 12. HP believes this leads to an “infinite loop” where the system continually passes search requests on to different registries without ever locating a data item. *Id.* at 14.

PersonalWeb responds that HP failed to present any actual evidence of indefiniteness. 6:11-CV-683, Docket No. 175 (“HP Response”) at 2. In particular, PersonalWeb contends all of

HP's "evidence" is attorney argument. *Id.* PersonalWeb also argues the specification discloses sufficient structure and procedures associated with the recited functions. *Id.* at 10. PersonalWeb argues the PTAB identified specific structure for both "existence means" and "local existence means" during a recent IPR, which indicates both must be sufficiently disclosed in the specification. *Id.* at 12–13. Further, PersonalWeb asserts that the procedures identified by the PTAB perform the recited structure. *Id.* at 14. PersonalWeb contends Defendants' "infinite loop" argument is based on a mischaracterization of the claim language. *Id.* PersonalWeb believes the claims do not require the tracking of data items across an infinite scope. *Id.* at 15. Instead, it is limited to searching with a "plurality of data items." *Id.* Therefore, PersonalWeb believes the Claims are not indefinite. *Id.*

HP's argument is premised on the assertion that the claimed function must determine every instance of a file across an entire system. See HP Motion at 11 ("But they are not capable of comprehensibly searching the entire system—that is, they cannot determine whether **or not** a data item is present in the system.") (emphasis in original). However, as articulated in the "existence means" section above, the function does not require determining every instance across the entire system. HP contends the Claims are indefinite because there is no disclosure regarding how to search across the entire system, but that is not required by the Claims. All that is required is searching "in the system." See '791 Patent, Claim 1, at 39:22.

The appropriate structure for both "existence means" and "local existence means" was determined and addressed above. See *supra*. Because there is sufficient disclosure to define the corresponding structure, the Claims are not indefinite. See *Biomedino, LLC v. Waters Techs. Corp.*, 490 F.3d 946, 950 (Fed. Cir. 2007) ("While the specification must contain structure linked to claimed means, this is not a high bar."); *Noah Sys., Inc. v. Intuit Inc.*, 675 F.3d 1302, 1312

(Fed. Cir. 2012) (articulating considerations for determining whether a computer algorithm is definite). Further, every other Defendant proposed a corresponding structure, and the PTAB determined a definite structure during the '791 Patent's IPR. See 6:11-CV-655, Docket No. 89, Ex. D at 22–23. This further supports a finding that the Claims are not indefinite.

HP's Motion for Summary Judgment of Indefiniteness (6:11-CV-683, Docket No. 164) is **DENIED**.

### CONCLUSION

For the foregoing reasons, the Court interprets the claim language in this case in the manner set forth above. For ease of reference, the Court's claim interpretations are set forth in a table in Appendix A. HP's Motion for Summary Judgment of Indefiniteness (6:11-CV-683, Docket No. 164) is **DENIED**. Facebook's Motion for Summary Judgment of Indefiniteness (6:12-CV-662, Docket No. 66) is **DENIED**.

**So ORDERED and SIGNED this 5th day of August, 2013.**

A handwritten signature in black ink, appearing to read 'Leonard Davis', written over a horizontal line.

**LEONARD DAVIS**  
**UNITED STATES DISTRICT JUDGE**

## APPENDIX A

Term	Page
Substantially unique identifier	An identity for a data item generated by processing all of the data in the data item, and only the data in the data item, through an algorithm that makes the identifier substantially unique
Identifier	No construction necessary
True Name	An identity for a data item generated by processing all of the data in the data item, and only the data in the data item, through an algorithm that makes the identifier substantially unique
Data identifier	An identity for a data item generated by processing all of the data in the data item, and only the data in the data item, through an algorithm that makes the identifier substantially unique
Data item identifier	An identity for a data item generated by processing all of the data in the data item, and only the data in the data item, through an algorithm that makes the identifier substantially unique
Digital identifier	An identity for a data item generated by processing all of the data in the data item, and only the data in the data item, through an algorithm that makes the identifier substantially unique
Data items	Sequence of bits
Data files	A named data item(s)
File system	No construction necessary
Sufficient number of copies	No construction necessary
Licensed/unlicensed	No construction necessary
Distributing a set of data files across a network of servers	No construction necessary
Access means for accessing a particular data item using the identifier of the data item	<p>Function: Accessing a particular data item using the identifier of the data item</p> <p>Structure: At least one processor (7:62–63), programmed to use a True File Registry and a True Name in accordance with one or more of the mechanisms described at ‘791 33:51–34:32; 24:45–58; 17:10–45; 16:10–34; 21:52–66; 25:25–38; 35:38–41, 51–61</p>

<p>Data associating means for making and maintaining, for a data item in the system, an association between the data item and the identifier of the data item</p>	<p>Function: Making and maintaining, for a data item in the system, an association between the data item and the identifier of the data item</p> <p>Structure: At least one processor (7:62–63), programmed to use a True File Registry and a True Name in accordance with one or more of the ‘Assimilate Data Item’ primitive mechanisms depicted in Figure 11 and described at 14:40–15:4</p>
<p>Existence means for determining whether a particular data item is present in the system, by examining the identifiers of the plurality of data items</p>	<p>Function: Determining whether a particular data item is present in the system, by examining the identifiers of the plurality of data items</p> <p>Structure: At least one processor (7:62–63), programmed to use a True File Registry and a True Name in accordance with one or more of the mechanisms described at ‘791 Fig. 14 (S260); Fig. 11 (S232); 15:54–57; 14:53–56 (beginning with “Next”); Fig. 28; 23:53–24:13; Figures 16(a) and 16(b); 16:38–17:18</p>
<p>Local existence means for determining whether an instance of a particular data item is present at a particular location in the system, based on the identifier of the data item</p>	<p>Function: Determining whether an instance of a particular data item is present at a particular location in the system, based on the identifier of the data</p> <p>Structure: At least one processor (7:62–63), programmed to use a True File Registry and a True Name in accordance with one or more of the mechanisms described at ‘791 Fig. 14 (S260); 15:54–57; Fig. 28; 23:53–24:13</p>
<p>Identity means for determining, for any of a plurality of data items present in the system, a substantially unique identifier, the identifier being determining using and depending on all of the data in the data item and only the data in the data item, whereby two identical data items in the system will have the same identifier</p>	<p>Function: Determining, for any of a plurality of data items present in the system, a substantially unique identifier</p> <p>Structure: At least one processor (7:62–63), programmed to perform the Calculate the True Name mechanism in accordance with ‘791 12:54–13:19 and 14:1–39</p>