January 12, 2011

The Honorable John D. Love
William M. Steger Federal Building and Unites States Courthouse
211 W. Ferguson, Room 210
Tyler, Texas 75702

Re: *Bedrock Computer Technologies, LLC v. SoftLayer Technologies, Inc.*, 6:09-CV-00269 – Defendants Letter Brief Requesting Permission to File Motion for Summary Judgment of Non-Infringement

Dear Judge Love:

Defendants respectfully request the Court's permission to file a joint motion for summary judgment of non-infringement of U.S. Patent Number 5,893,120 ("the '120 patent").[1] The accused instrumentalities[2]—servers using versions of the Linux kernel prior to 2.6.25[3] do not meet all elements of the '120 patent because: (1) removal of records does not occur "when the linked list is accessed" ('120 patent claims 1, 3, 4, 5, 7, and 8); (2) the removed records are not "expired" ('120 patent claims 1, 3, 5, and 7); (3) there is no "dynamically determining maximum number" of expired records to remove ('120 patent claims 2, 4, 6, and 8) (for all accused versions); (4) the accused code does not remove an expired record while using the record search means to search for a record to delete (for all accused versions); and (5) there is no evidence that the accused code has ever executed, as required by all asserted claims. Therefore, Defendants are entitled to summary judgment of noninfringement of the '120 patent.

A. The accused code does not remove expired records "when the linked list is accessed."

All independent claims of the '120 patent require that both identification and removal of expired records occur "when the linked list is accessed." The Court has construed this term to mean "both identification and removal of the automatically expired record(s) occurs during the same access of the linked list." Dkt. No. 369 at 21-22.

In the accused code, the identification and removal does not occur during the "same access" of the linked list. Specifically, Bedrock accuses code that accesses a linked list and identifies a "candidate" record (an IP route) that may (or may not) be later removed during a separate and distinct access. The accused code includes a while-loop that traverses an IP routing cache to determine whether a particular IP route (the target record) is already present. *See, e.g.*,

[1] "Defendants" refers to all defendants in the litigation, each of which joins in this letter brief.
[2] The Accused Instrumentalities are computer equipment configured with specified versions of the Linux operating system kernel. For convenience, this letter brief refers to "accused code" as a short-hand reference to the pertinent code identified by Bedrock in the specified versions of the Linux operating system kernel.
[3] Except for Google, Defendants are only accused of using versions of Linux prior to 2.6.25. Accordingly, this joint letter only addresses versions of Linux prior to 2.6.25, and all references to "accused versions" or "accused code" in this letter means versions of Linux prior to 2.6.25. Google is filing a separate letter requesting permission to file a motion for summary judgment of non-infringement for its modified 2.6.26 version of Linux.

Infringement Contentions for 2.6.18 ("Infring. Cont."), at 6-7 (accusing the function rt_intern_hash() in net/ipv4/route.c). The accused code includes conditional logic that identifies a candidate record (IP route) that may (or may not) be removed from a linked list under certain circumstances. *See id*. However, the accused code does not remove the candidate record during the same while-loop that accesses the linked list to search for a target record. Rather, the code performs any removal *only* after the while-loop has completed its access of the linked list.[4] *See, e.g.*, Linux 2.6.18.1, route.c, lines 915-1044. That is, the removal occurs during a separate and distinct access. Thus, there is no identification and removal of an expired record during the "same access" of a linked list, and there is no literal infringement of the independent claims in the '120 patent, including the means-plus-function claims 1 and 5. *Intellectual Sci. & Tech., Inc. v. Sony Elecs., Inc.*, 589 F.3d 1179, 1183 (Fed. Cir. 2009) ("For a means-plus-function claim term, the term literally covers an accused device if the relevant structure in the accused device performs the identical function recited in the claim and that structure is identical or equivalent to the corresponding structure in the specification.").

Bedrock has not alleged infringement under the doctrine of equivalents. Summary judgment of non-infringement based on the doctrine of equivalents is proper on this basis alone. Further, identifying the candidate record and removing the record in different accesses is substantially different than identifying and removing the record during the "same access." Put another way, removing a record after the while-loop has completed accessing the linked list and identifying the candidate record is substantially different from removing a record during the "same access," and cannot be held equivalent under the doctrine of equivalents. *See Planet Bingo v. GameTech, Int'l*, 472 F.3d 1338, 1344 (Fed. Cir. 2006) (holding that under the doctrine of equivalents, "before" can never be equivalent to "after"). The same goes for equivalence under 35 U.S.C. § 112(6). *Ishida Co. v. Taylor*, 221 F.3d 1310, 1316 (Fed. Cir. 2000) (doctrine of equivalents and 112(6) equivalence collapse into same analysis for structural equivalents). To expand the claim limitation to include removal "after" the access to the linked list that identifies the candidate record would entirely ignore the construction of this Court and the clear intent of the patentee. Thus, both identification and removal of the candidate record in the accused Linux versions do not occur when the linked list is accessed, and Defendants are entitled to summary judgment of non-infringement for those kernels.

B. In the accused code, the record that is removed is not "expired."

The independent claims of the '120 patent require the removal of an "expired" record, which this Court has construed to mean "obsolete and therefore no longer needed or desired in the storage system because of some condition, event, or period of time." Dkt. No. 369 at 7-9. In the accused code, Bedrock alleges that the "expired" record is the "candidate" for deletion

[4] Bedrock's proposed reexamination amendments to claims 3 and 7 support this argument. In its Nov. 23, 2010 Amendment in *Ex Parte* Reexamination of U.S. Patent No. 5,893,120 ("Amendment"), Bedrock amended the claims to read "accessing the linked list of records to search for a target record, identifying at least some of the automatically expired ones of the records while searching for the target record, and removing at least some of the automatically expired records from the linked list when the linked list is accessed." Bedrock states that this amendment is merely a clarification. *See* Amendment, at 3, 7**.** Nonetheless, once the while-loop ends, there is no further searching of the linked list for a target record; the target record has already been found or does not exist in the route cache. In either event, the removal of a candidate occurs only after the first access to the linked list has ended.

identified within the while-loop that accesses the linked list while searching for a record in the routing cache. *See, e.g.*, Infring. Cont. at 6-7.

But the "candidate" record for removal is not "expired." When the while-loop is accessing the linked list of IP routing cache records, each record is evaluated by the function rt_score(). *See id*. at 6; Linux 2.6.18.1, route.c at 966-973. Rt_score() weighs a number of criteria, including the time the record was last used and the type of IP route (e.g., unicast or multicast), to determine a score. *See, e.g.*, Linux 2.6.18.1 at lines 542-561. If the score for a record is lower than the lowest score of the records that have been previously evaluated during that access of the linked list, the record becomes the new "candidate" for deletion. *Id*. Thus, rt_score() does not determine whether a record is expired, or obsolete, and therefore no longer needed or desired—in fact, the candidate record may or may not be removed, and the routing information contained in that candidate record can later be used by the system as a valid IP route. The only determination made is that the candidate record is the lowest scoring of the records in the linked list at that time.

Because the record identified by Bedrock as the "expired" record is not, in fact, expired at all, no accused version can meet all the limitations of the independent claims of the '120 patent. Bedrock has not alleged infringement under the doctrine of equivalents, but there cannot be infringement under the doctrine of equivalents unless all limitations are met. *Warner-Jenkinson Co. v. Hilton Davis Chem. Co*. 520 U.S. 17, 29-30 (1997). Accordingly, Defendants are entitled to summary judgment of non-infringement of the '120 patent as to those versions of the Linux kernel.

C. No version of the Accused Linux Kernels contains code for "dynamically determining maximum number" of records to remove.

Dependent claims 2, 4, 6, and 8 of the '120 patent require that the accused code dynamically determine a maximum number of expired ones of the records to remove when the linked list is accessed. The Court construed "dynamically determining" to mean "making a decision based on factors internal and external to the information storage and retrieval system." Dkt. No. 369 at 15. The Court declined to construe "maximum number," but emphasized that "[t]he maximum number need only be an upper limit as to the records to be removed." *Id*. at 18. Bedrock accuses the binary decision of whether or not to remove a single candidate record as the dynamic determination of a maximum number. *See* Infring. Cont. at 8-9. This decision is merely and always whether to remove the single candidate record; nothing more, nothing less. Such a decision is plainly not "dynamically determining maximum number," as claimed.

Instead, the accused code performs a binary decision of whether or not to perform a removal of a single candidate record. It does not determine a number, much less a maximum number of records to remove. Furthermore, the accused code does not make any evaluation of internal or external conditions to determine a number of records to remove. A yes/no determination of whether to remove a single candidate record plainly does not qualify as "dynamically determining maximum number."

Nor does the accused Linux code choose between alternate algorithms as required by the "means for dynamically determining" limitations in claims 2 and 6. *See* Dkt. No. 369 at 40. The Court's construction requires "software instructions to dynamically determine a maximum number of records to remove by choosing a search strategy of removing all expired records from a linked list or removing some but not all of the expired records as described in col. 6 line 56 –

col. 7 line 15 and/or programmed with software instructions to dynamically determine a maximum number of records to remove by choosing between the pseudo-code of the Search Table Procedure (cols. 11 and 12) or Alternative Version of Search Table Procedure (cols. 11, 12, 13, and 14), and equivalents thereof." As can be seen, there must be a choice between two search strategies, or a choice between two search table procedures in the pseudocode in the '120 patent. There is no such choice in the accused code. In substantial contrast to claims 2 and 6 as construed by the Court, there is only a decision in the accused code of whether or not to remove a single candidate record.

Again, Bedrock has not alleged infringement under the doctrine of equivalents, and the structural differences prevent equivalence under §112(6). *Ishida*, 221 F.3d at 1316. Without meeting all the elements, there can be no infringement. *Warner-Jenkinson*, 520 U.S. at 29-30. Therefore, Defendants are entitled to summary judgment of non-infringement of claims 2, 4, 6, and 8 of the '120 patent.

D. The accused versions do not meet the delete limitation of Claim 5.

Claim 5 requires a "mea[n]s, utilizing the record search means, for inserting, retrieving, and deleting records from the system and, at the same time, removing at least some expired ones of the records in the accessed linked list of records." *See* '120 Patent, Claim 5. The Court construed the means for deleting records to include Figure 7 or the Delete pseudocode in Columns 11-12. Dkt. No. 369 at 42-43. Figure 7 and the pseudocode require the calling of "record search means" to remove expired records it identifies while searching for the record to delete. *See* '120 Patent, at Fig. 7, Col 11-12. When the record to delete is found, the procedure in Figure 7 and the pseudocode deletes the record and then terminates. *Id*.

Bedrock's infringement contentions acknowledge that the accused code does not utilize the record search means, portions of rt_intern_hash(), to delete a record. *See* Infring. Cont. at 35-36. Bedrock identifies rt_del() as the code that deletes a record from the routing cache. However, rt_del() simply traverses a linked list until it finds the record to delete and then deletes it. Rt_del() does not search for, identify or remove automatically expired records as is required by the record search means of Figure 3. The code identified by Bedrock then calls rt_intern_hash() after a record has been deleted by rt_del() to insert a new entry into a linked list in the routing cache. *See, e.g.*, Infring. Cont. at 34-36. Thus, the accused code does not remove an expired record while utilizing the record search means to search for a particular record to delete as required by Figure 7 and the pseudocode and cannot infringe Claim 5 either literally or by equivalents.

E. There is no evidence to show that the accused code has executed on Defendants' systems.

Bedrock has offered no evidence that the accused code has ever executed on any of the Defendants' actual systems. Bedrock's only argument to date is that the mere existence of the accused code in Defendants' systems is sufficient to prove infringement. For the method claims of the '120 patent, claims 3, 4, 7 and 8, such an argument is plainly insufficient, as method claims are only infringed when they are performed.

Even assuming that the accused code would infringe the method claims of the '120 patent if performed, there is no evidence that any of the Defendants' systems have ever executed it. Bedrock cannot meet its burden of proof regarding infringement of the method claims of the

’120 patent. Therefore, Defendants are entitled to summary judgment of non-infringement of claims 3, 4, 7, and 8.

Further, when an apparatus claim requires more than “mere capability,” a patent owner must show that all of the claimed elements are present in the accused device. *See Fantasy Sports Props. v. Sportsline.com, Inc.*, 287 F.3d 1108, 1118 (Fed. Cir. 2002) (affirming a judgment of non-infringement where the claims language “means for scoring . . . bonus points” required the software to actually award bonus points”). Here, the system claims of the ’120 patent require the removal of expired records while accessing the linked list, not just the capability of performing these limitations.

Defendants’ networks comprise multiple hardware and software layers that spread out network traffic loads. Defendants’ networks are architected and configured in such a way that the accused code is not useful, functional or operational. Bedrock can advance no evidence that the accused code does run or has ever run and so cannot meet its burden of proof of infringement. Therefore, Defendants are similarly entitled to summary judgment of non-infringement of claims 1, 2, 5, and 6.

F. Conclusion

As outlined above, there are no genuine issues of material fact and Defendants are entitled to summary judgment of non-infringement of the ’120 patent given Bedrock’s admissions. For the foregoing reasons the Court should permit the Defendants to file a joint motion.

Respectfully submitted,

/s/ Claude M. Stern
Claude M. Stern
Quinn Emanuel Urquhart & Sullivan
Attorneys For Defendant Google Inc. and Match.Com, LLC

/s/ Alan L. Whitehurst
Alan L. Whitehurst
Alston & Bird LLP
Attorneys For Defendants MySpace Inc. and AOL Inc.

/s/ E. Danielle T. Williams
E. Danielle T. Williams
Kilpatrick Townsend & Stockton LLP
Attorneys For Defendants SoftLayer Technologies, Inc. and Amazon.com, Inc.

/s/ Yar R. Chaikovsky
Yar R. Chaikovsky
McDermott Will & Emery
Attorneys For Defendant Yahoo! Inc.

DM_US 27421595-3.049256.0026