United States District Court
Northern District of California

UNITED STATES DISTRICT COURT

NORTHERN DISTRICT OF CALIFORNIA

THOUGHT, INC.,
Plaintiff,
v.
ORACLE CORPORATION, et al.,
Defendants.

Case No. 12-cv-05601-WHO

ORDER ON SUMMARY JUDGMENT

Re: Dkt. Nos. 202, 204, 205, 207, 208, 210, 212, 216, 221, 224, 225

Currently before me are the parties' motions for summary judgment regarding the invalidity and infringement of Thought's '197 Patent.[1] Oracle moves for summary judgment arguing that its accused product – TopLink – uses a different architecture than the '197 Patent's invention and that the earliest version of TopLink (which operated in a different manner than the current version of TopLink) renders the '197 Patent invalid under the doctrines of obviousness and anticipation. Oracle also moves to strike portions of Thought's expert's declaration on the ground that Thought failed to disclose three of its theories of infringement as required under my orders.

Thought moves for partial summary judgment on Oracle's obviousness/combination defenses. It argues that Oracle disregarded my order requiring Oracle to elect and disclose a limited number of prior art references that it intended rely on by having its expert rely on other undisclosed references in his expert report. As a consequence of Oracle's alleged circumvention of that order, Thought asks me to grant summary judgment in its favor on a number of Oracle's Section 103 obviousness affirmative defenses.

I heard argument on these motions on April 20, 2016. For the reasons described below, I GRANT Oracle's motion to strike and motion for summary judgment, and DENY Thought's

[1] The only patent still at issue in this suit is U.S. Patent No. 5,857,197 ('197 Patent).

motion as moot.

BACKGROUND

The ’197 Patent was filed March 20, 1997 and issued on January 5, 1999. It claims a “System and Method for Accessing Data Stores as Objects.” It was aimed at facilitating the transfer of information between two popular schemes for organizing information in a computer: the relational model and the object model. Generally speaking, the technology at issue – object relational mapping or ORM – teaches methods to allow users to make object (software) applications and relational databases work together, despite the different ways software applications and database systems organize data. Claim Construction Order (Dkt. No 116) at 1.

Both parties characterize the invention at issue and the accused products as “middleware” that “sit” or operate between the object applications and the relational databases. As Oracle’s expert – Dr. Antony L. Hosking – explains, the ’197 Patent allowed an object application to retrieve data from a relational database as follows: (1) the user defined the rules (metadata) that map an application’s object classes to a relational database’s tables; (2) the object application invoked the ORM system with a request for data stored in the relational database; (3) the ORM used the metadata to put the application’s request into a form that would work with the database; (4) the ORM system accessed the database and requested the data; (5) the ORM system used the metadata to put the relational database content into a form that would work with the object application; and (6) the ORM system sent the information back to the application. Declaration of Antony L. Hosking in Support of Oracle’s Motion for Summary Judgment (Dkt. No. 205-1, “Hosking MSJ Decl.”) ¶ 29.[2]

According to Thought, and as discussed at Claim Construction, the novelty of the ’197 Patent’s invention “is the fact that the two adapters or interfaces are used at runtime and one specializes in the object model and the other specializes in the relational database model.” Claim Construction Order at 1-2; Thought’s Opening Claim Construction Brief (Dkt. No. 70) at 3. One benefit of the ’197 Patent’s invention was that someone could “dynamically build the code” for

[2] The parties dispute whether prior art systems functioned in the same way as the ’197 Patent’s invention and whether “translation” – if any – is required for the data at steps 3 and 5.

accessing the relational database on behalf of the objection application "at runtime, as opposed to requiring that information to be "hard coded" and compiled at start up. *See, e.g.*, '197 Patent at 2:46-49.[3]

Thought alleges that Oracle's products – TopLink 10, TopLink Essentials, TopLink 11, TopLink 12, and EclipseLink – infringe Claims 1, 3, 5, 7 and 8 of the '197 Patent.[4] I construed seven terms from the '197 Patent in my October 22, 2014 Claim Construction Order: object, object schema, packing, unpacking, interface, adapter, and adapter abstraction layer.[5]

LEGAL STANDARD

Summary judgment on a claim or defense is appropriate "if the movant shows that there is no genuine dispute as to any material fact and the movant is entitled to judgment as a matter of law." Fed. R. Civ. P. 56(a). In order to prevail, a party moving for summary judgment must show the absence of a genuine issue of material fact with respect to an essential element of the non-moving party's claim, or to a defense on which the non-moving party will bear the burden of persuasion at trial. *See Celotex Corp. v. Catrett*, 477 U.S. 317, 323 (1986). Once the movant has made this showing, the burden then shifts to the party opposing summary judgment to identify "specific facts showing there is a genuine issue for trial." *Id.* The party opposing summary judgment must then present affirmative evidence from which a jury could return a verdict in that party's favor. *Anderson v. Liberty Lobby,* 477 U.S. 242, 257 (1986).

On summary judgment, the court draws all reasonable factual inferences in favor of the non-movant. *Id.* at 255. In deciding a motion for summary judgment, "[c]redibility determinations, the weighing of the evidence, and the drawing of legitimate inferences from the

[3] According to Thought, and not disputed by Oracle, much of the prior art was burdened by the need to embed "access code" in the object application to allow access to various database programs. That approach was costly and time-consuming. The '197 Patent removed the access code from the objection application and put it into the middleware layer. Rebuttal Declaration of Dr. Hosagrahar V. Jagadish (Dkt. No. 212-7) ¶¶ 10-11. That layer could also be altered at runtime, resulting in a much more flexible and scalable product. *Id.* ¶ 11.

[4] I will refer to the accused products generally as TopLink unless necessary to separately discuss EclipseLink or a specific version of TopLink.

[5] The parties stipulated to the construction of object name. Dkt. No. 184.

facts are jury functions, not those of a judge." *Id.* However, conclusory and speculative testimony does not raise genuine issues of fact and is insufficient to defeat summary judgment. *See Thornhill Publ'g Co., Inc. v. GTE Corp.,* 594 F.2d 730, 738 (9th Cir.1979).

DISCUSSION

I. ORACLE'S MOTION TO STRIKE

Oracle moves to strike portions of the reports of Thought's expert Dr. Jagadish that refer to three infringement theories identified by Jagadish as "Trace 2," "Trace2s,"[6] and "Trace 3."[7] Oracle argues that these three theories (which Oracle refers to as Theories 2, 3, and 4) as well as related references, should be stricken because they were either stricken by me in my October 7, 2015 Order or not adequately identified in Thought's infringement contentions. Thought opposes the motion, arguing that a stipulation between the parties allows Thought to rely on these theories and, in any event, the theories were disclosed in Thought's Original as well as the stipulated Supplemental Infringement Contentions.

A. Background

Pursuant to the requirements of the Patent Local Rules, Thought served its infringement contentions and accompanying document production on April 12, 2013, and Oracle served its Original and Supplemental Invalidity Contentions in August 2013, September 2013, and January 2014. October 7, 2015 Order at 2. After I issued my Claim Construction Order on October 22, 2014, the parties discussed and agreed to supplement their infringement and invalidity contentions. The parties, however, did not file any stipulation with the court regarding their purported agreed-to supplementations. *Id.*

Disputes between the parties over their proposed supplementations led to Oracle filing a

[6] Oracle explains that there are two Trace 2 theories, "Trace2" and "Trace2s," because the results of each Trace vary for different accused versions of TopLink. Oracle MTS at 5 n.9. A trace is a report of the execution of a software program. Hosking MTS Decl. ¶ 25.

[7] Oracle asks me to strike the following from the Jagadish Infringement Report and Rebuttal Report: (i) exhibits D2-H2 and D3-F3 of Dr. Jagadish's Infringement Report; (ii) ¶ 641 of that report; (iii) references to the "Trace 2" and "Trace 3" theories on pages 229-238 of that report; (iv) all files in Exhibit K to that report, except the "Trace1" files; (v) ¶¶ 674-678 of that report; (vi) section XV of that report; and (vii) the last sentence of each of ¶¶ 84-86 of the Rebuttal Report.

motion to strike "improper disclosures and contentions" by Thought and Thought filing a motion to strike Oracle's new Section 101 contentions. Dkt. Nos. 161, 163.[8] The parties could not agree on what their alleged agreement was. On October 7, 2015, I issued an Order determining that the parties' purported agreement to allow supplementation of their contentions was not effective because they could not agree on its terms and it had not been filed with or entered by the court. October 7, 2015 Order (Dkt. No. 176) at 12-13. Therefore, I analyzed whether the parties were entitled to supplement their contentions under Patent Local Rule 3-6, and concluded that both sides lacked justification to amend. Accordingly, I granted both motions to strike. *Id.* at 14.

Following that Order, the parties engaged in further negotiations and submitted a stipulation identifying "the portions of the text of the supplemental contentions that should be deleted based on the parties' respective motions to strike." Dkt. No. 193 at 1. As they explained, the parties "have agreed to the content of a revised set of supplemental contentions that are consistent with the parties' earlier agreement, but that exclude the portions that they agree were the subject of the parties' respective motions." *Id.* That agreed-set of revised contentions was exchanged on November 24, 2015. *Id.* Oracle stipulated and agreed "that Thought's revised contentions do not include the theories that were the subject of Oracle's motion to strike. (Dkt. 176)." *Id.* Finally, the parties agreed "that they will not move to strike any portion of an expert report, or any portion of an expert's testimony, on the grounds that it allegedly states a new theory of infringement or invalidity, provided that such alleged new theory of infringement or invalidity is set forth in the parties' original Patent Local Rule 3-1 or 3-3 contentions, an amendment to those contentions permitted by the Court, and/or the revised contentions exchanged on November 24, 2015." *Id.* at 2.[9] The parties' stipulation was entered by the Court on November 30, 2015.

[8] As relevant to this motion, in its prior motion to strike Oracle asked the Court to "hold" Thought to "its original JPA theory" and strike Thoughts "new 'TopLink sessions,' 'container-managed persistence,' and 'bean-managed persistence' infringement theories," as well as "functionality" "outside the context of JPA." Dkt. No. 163 at 18 & 19-22. That Order defined "Java Persistence API" as "JPA." *Id.* at 9. On this motion, Oracle argues that the "Trace2" theory – which corresponds to a proprietary TopLink API ("Session.readObject") was covered by the October 7, 2015 Order striking new "TopLink sessions" contentions.

[9] According to Oracle, the purpose of that stipulation was, in part, to allow Thought to supplement its existing contentions with a new level of detail following Thought's review of the Oracle source

Thought then served its opening and rebuttal expert reports on December 7, 2015 and January 25, 2016. Trial is set for October 31, 2016.

Oracle's current motion to strike argues that Theory 2 ("Trace2" from the Jagadish expert reports) was expressly stricken by my October 7, 2015 Order and Theories 3 and 4 ("Trace2s" and "Trace3") are wholly new, not "based on" the JPA-API, and not covered by the November 30th stipulation. Thought responds that through the parties' November 30th stipulation, Oracle expressly agreed to allow Thought to raise these theories and argues that, in any event, the theories are "JPA-compliant" and consistent with their Original Infringement Contentions. [10]

B. Legal Standard

Pursuant to the Patent Local Rules for the Northern District of California, a party claiming infringement must submit infringement contentions within fourteen days of the parties' initial Case Management Conference. Patent L.R. 3-1. These must include, among other things, "each accused apparatus, product, device, process, method, act, or other instrumentality ('Accused Instrumentality') of each opposing party of which the party is aware," and must "be as specific as possible." Patent L.R. 3-1(b).

"[A]ll courts agree that the degree of specificity under Local Rule 3–1 must be sufficient to provide reasonable notice to the defendant why the plaintiff believes it has a 'reasonable chance of proving infringement.'" *Shared Memory Graphics LLC v. Apple, Inc.*, 812 F.Supp.2d 1022, 1025 (N.D. Cal. 2010) (quoting *View Eng'g, Inc. v. Robotic Vision Sys., Inc.*, 208 F.3d 981, 986 (Fed. Cir. 2000)). While the patent rules do not "require the disclosure of specific evidence nor do they require a plaintiff to prove its infringement case, . . . a patentee must nevertheless disclose what in each accused instrumentality it contends practices each and every limitation of each asserted claim

code. Oracle MTS at 10.

[10] According to Oracle, and not disputed by Thought, "JPA" stands for Java Persistence API, and "API" stands for Application Programming Interface. "An API is the set of protocols that a software program provides so that other programs can use that first program. An API does not dictate how the first program implements those protocols, but merely sets out how they can be invoked by another program. In this way, the other programs that use the first program's API do not have to change even if the first program's implementation of that API changes." Oracle MTS at 2 n.3 [citing Hosking MTS Decl. at ¶¶ 5-6]).

to the extent appropriate information is reasonably available to it." *DCG Sys. v. Checkpoint Techs., LLC*, No. C 11–03792 PSG, 2012 WL 1309161, at *2 (N.D.Cal. Apr. 16, 2012); *see also Shared Memory Graphics*, 812 F.Supp.2d at 1025 (stating that patent holder "must map specific elements of Defendants' alleged infringing products onto the Plaintiff's claim construction").

C. The Impact of the October 15, 2015 and November 30, 2015 Orders and Stipulation

In my prior Order striking Thought's proposed amended contentions, I struck from the case any infringement "theories that are not related to [Thought's] original contentions based on Java Persistence API ('JPA')." October 7, 2015 Order at 9, 14. Any argument by Thought that the stipulation I entered on November 30, 2015 allowed Thought to "restore" stricken infringement theories is rejected. I entered the November 2015 stipulation based on representations by Oracle and Thought that the stipulation ***excluded*** infringement (and invalidity contentions) that the parties "agree were the subject of the parties' [prior] motions." November 30, 2015 Order at ¶ 6. The November stipulation and Order did not allow any restoration of infringement theories that Oracle sought to strike, *i.e*., any infringement theories not "based on Java Persistence API."

Thought's argument that Oracle's motion to strike is improper because Oracle agreed not to move to strike any contentions agreed-to in the November stipulation is without merit because, as Oracle argues, the contested theories were not agreed-to in that stipulation. Finally, Oracle's "silent-acquiescence" to the challenged Trace theories during December and January (when the Jagadish expert reports were produced) does not preclude this motion. Oracle objected to these theories after their assertion in discovery. Moving to strike at the summary judgment stage is appropriate given the timing of the expert reports and posture of the case.

D. Theory 2 Was Excluded by Court Order or Not Adequately Disclosed

Jagadish's Theory 2 (Trace2) of infringement is as follows: TopLink uses a "public Object" call which is found at publicinterface.Sessions.java to communicate a request including an object comprising object attributes and an object name from the object application. Jagadish MTS Decl. ¶ 35; Hosking MTS Decl. ¶¶ 26, 45; *see also* Jagadish Report Exs. G2, H2.

As noted above, in its prior motion to strike Oracle moved to strike and I struck any infringement theories not "based on" the JPA-API, including a newly asserted theory based on the use of TopLink's proprietary Session.readObject API. Dkt. No. 163 at 18 & 19-22. Oracle argues here that Jagadish's Trace2 infringement theory is barred by the Order because the theory requires TopLink to *directly* use a "readObject(…) method" found only in TopLink's proprietary Session API. Thought's attempt to rely (or re-rely) on the Session API is, according to Oracle, also prevented by the "redlined" amended contentions that expressly struck through any mention of "Session API." *See* Declaration of Steven D. Moore ISO Motion to Strike, Ex. 4 (Dkt. No. 208-3) at 3; *see also* Ex. 3 (final contention charts) at 195, 295. According to Oracle's expert, by removing reference to "Sessions API" in their final contentions Thought limited itself to functions "invoked directly" through JPA-API *because* Thought otherwise failed to identify any components of TopLink that could directly invoke those functions. Hosking MTS Decl. ¶¶ 16-18; *see also id*. ¶ 22 ("in Thought's final contentions, references to TopLink's Sessions are limited to the internal use of those Sessions by TopLink to implement the standard JPA API. The redline shows that Thought removed all references to an object application directly using Sessions."). In other words, Oracle asserts that because the methods relied on by Thought as infringing (like use of "readObject" method) are part of the Session API and can only be invoked through the Session API, their theory of infringement necessarily is excluded.

Thought's response, initially, is that unlike the "TopLink Sessions" theory struck by the Court's prior order for not being "based on JPA," Theory 2 *requires* the JPA features EntityManager and EntityManagerFactory, as well as the proprietary JpaEntity Manager and therefore is "based on JPA." Thought Oppo. at 10.

Thought also argues, as supported by Jagadish, that "the call" "sessions.readObject(object)" was included in the Supplemental Infringement Contentions (and not stricken in the redlines negotiated by the parties) and described as "communicating a request including an object comprising object attributed and an object name from the object application." Jagadish MTS Decl. ¶ 36 (citing Redlined Supplemental Infringement Contentions at 195, 295). Even though in the *same* reference the "use the Session API" was struck through, that did not matter according to Thought because the implementation of the EntityManager from the JPA-API requires a session functionality that "wraps" client-side and server-side sessions together and without that, the system "will not persist or retrieve object data from a database." Jagadish MTS Decl. ¶ 34. In other words, these "wrapped" sessions (unlike the "TopLink" Sessions) were not excluded from the contentions because they are inherently related to and necessary for the implementation of the JPA-API in order for any middleware to work.

I conclude that Thought's Theory 2 ("Trace2") should be stricken. To reiterate, Thought is precluded from relying on any functionality necessarily invoked by TopLink's Session API in light of my October 2015 Order. The parties dispute whether the only way to *get to* the "read_object" functionality relied on in Theory 2 is by first invoking TopLink's proprietary Session API. In its Motion to Strike reply, Oracle argues that the proprietary TopLink Session API is required for the Theory 2 infringement proof. Oracle MTS Reply at 3:23-4:2; *see also* Hosking MTS Decl. ¶¶ 19, 31, 39. At oral argument, Thought contested that assertion, and pointed to Theory 2's use of EntityManager as the gateway to the "wrapped sessions" Thought relies on in Theory 2.[11]

I need not resolve that dispute. Even if Thought is correct that it can avoid the Session API by relying on EntityManager and "wrapped sessions," this new theory *was not* adequately disclosed in either Thought's Original or the stipulated Supplemental Contentions, such that Oracle knew how TopLink was alleged to perform the step of communicating "a request including an object comprising object attributes and an object name from the object application." *See* Jagadish MTS Decl. ¶ 35. Thought's current argument that this theory was "implicitly" disclosed and that Oracle should have realized Thought intended to rely on the "internal" wrapped sessions when the Supplemental Infringement Contentions did not strike through "sessions.readObject(object)" and referred generally to the "functionality" of TopLink, is gamesmanship. The purpose of requiring parties to disclose the basis for their contentions is to make them *explicit* and streamline patent litigation. *See DCG Sys. v. Checkpoint Techs., LLC,* No. C 11-03792 PSG, 2012 WL 1309161, at *2 (N.D. Cal. Apr. 16, 2012) (a patentee must "disclose

[11] Thought's expert did not address this specific issue in his declarations.

what in each accused instrumentality it contends practices each and every limitation of each asserted claim to the extent appropriate information is reasonably available to it."); *see also Finjan, Inc. v. Proofpoint, Inc.*, No. 13-CV-05808-HSG, 2015 WL 1517920, at *7 (N.D. Cal. Apr. 2, 2015) ("if [plaintiff] believes that the first and second functions are contained within the obfuscated scripts, it was obligated to say so explicitly in its infringement contentions. Neither the Court nor the Defendants should be required to guess which aspects of the accused products allegedly infringe each claim element.").

Thought cannot avoid the impact of my October 2015 Order by now relying on "implicit" infringement arguments based on "indications" that TopLink accessed the unstruck-through "session.readObject(object)" method "via" *unspecified* "permitted paths." *See* Thought Summary Judgment slides at 263.

E. While Theory 3/Trace2s May be a "JPA-Based" Theory, It Was Not Adequately Disclosed in Original or Supplemental Contentions

Thought's Theory 3/Trace2s is that the EntityManager (created by the JPA-API) as well as a proprietary TopLink extension (JpaEntityManager) use "ReadObjectQuery" to implement JPA-API and perform functions covered by the claims. *See* Jagadish MTS Decl. ¶¶ 43-46; *see also* Jagadish Report, Exs. D2, E2, and F2. Jagadish explains that this infringement theory is "based on JPA" because this is how TopLink implements the functionality of EntityManager and EntityManagerImpl and "is the only known JPA-compliant implementation in TopLink/EclipseLink." Jagadish MTS Decl. ¶¶ 46, 51. To support its argument that Theory 3 was adequately disclosed, Jagadish points to mentions of "JpaEntityManager" and its extensions (createQuery and Query.getSingleResult) in the Original Infringement Contentions, and additional mentions of an extension (ObjectLevelReadQuery) in the Supplemental Contentions. *Id*. ¶¶ 47-49, 74.

Oracle responds, first, that this theory of infringement is not "based on JPA" because it depends solely on the direct invocation of TopLink's proprietary JpaEntityManager (as well as the proprietary EntityManagerImpl function) and is not reliant on a direct invocation by the JPA-

API.[12] *See* Hosking MTS Decl. ¶ 27. It also argues that this theory of infringement – where a Java application directly uses TopLink's proprietary ReadObjectQuery on its own – was not adequately disclosed in the Original or Supplemental Infringement Contentions and was waived when Thought redlined its attempted new reliance on "queries" from its final contentions on these elements.[13]

Assuming that Thought may – consistent with my October 2015 Order – rely on infringement theories which are "based on" JPA-API but not necessarily "invoked by" JPA-API, I still agree with Oracle that Trace3 was not adequately disclosed. In opposition and at oral argument, Thought relies on various instances in its Original and Supplemental Infringement Contentions where it mentioned TopLink's proprietary JpaEntityManager and its extensions (createQuery, Query.getSingleResult, ObjectLevelReadQuery). But it did not show that these names were disclosed with respect to claimed functions of "communicating a request" or "application bridge," as opposed to being mentioned with respect to other functions or generally as background. *See* Redlined Infringement Contentions (Ex. 3 to Moore Decl. ISO MTS) at pgs. 38-43, 195-03; *see also* Hosking MTS Decl. ¶ 32 ("Neither Thought's original nor final infringement contentions assert theories of infringement that involve an application invoking TopLink via the ReadObjectQuery API or the createQueryByExample API"); *cf*. Jagadish MTS Decl. ¶¶ 47-48 (citing pgs. 22-23 in Original Infringement Contentions discussing "first interface"). Tellingly, Thought does not point to any specific point in its Original or Supplemental Contentions where it disclosed that the "ReadObjectQuery" method performed the claimed functions of communicating a request or application bridge. As above, it is at best inconsistent for Thought to rely on the

[12] In Reply and at oral argument, Oracle also contended that Thought cannot rely on these arguments to demonstrate its theories are "based on JPA" because in a January 2016 meet and confer Thought admitted that its expert did not provide any traces of the portions of the code where these functions (EntityManagerFactory, EntityManager, JpaEntityManager) were invoked (at code lines 1-26) because the expert did not rely on those code lines 1-26 and admitted they were "irrelevant" for infringement. *See, e.g.*, Reply MTS at 10-11.

[13] Oracle points out that Thought's reliance on the fact that "query" was used hundreds of times in its contentions is irrelevant – especially when considering a query based ORM system – because Thought fails to identify any uses of query in connection with the claim functions purportedly performed in the Trace 3 analysis.

sporadic mentions of JpaEntityManager and its proprietary method extensions in other sections of its Original and Supplemental Contentions to argue Theory3 of infringement was "necessarily disclosed" or "disclosed by implication."

F. While Theory 4/Trace3 May be a JPA-Based Theory, It Was Not Adequately Disclosed by the Original or Supplemental Contentions

Thought's Theory 4/Trace 3 is based on a theory that the EntityManager (created by JPA-API) that is implemented in TopLink by EntityManagerImpl uses "createQueryByExample," an EntityManager extension in JpaEntityManager to perform functionality covered by the claims. *See* Jagadish Report Exs. D3, E3 and F3; *see also* Jagadish MTS Decl. ¶¶ 54-56. Jagadish asserts this theory was disclosed in the Original Contentions by reference to EclipseLink's defining and using the EntityManager interface "org.eclipse.persistence.jpa.package" to extend the EntityManager interface and implement "additional functionalities" "which are EclipseLink extensions to the entity manager API." Jagadish Decl. ¶ 57. Jagadish also relies on his assertion that EntityManagerImpl is the only known JPA-API compliant implementation in TopLink/EclipseLink, therefore "the functionality of EntityManagerImpl and JpaEntityManager are necessarily part of the TopLink/EclipseLink implementation of the JPA standard." *Id*. ¶ 59.

Oracle responds that this theory is not "based on JPA" because it is not specified/directly invoked by the JPA-API, but instead is invoked through Oracle's proprietary JpaEntityManager. Oracle also argues that invoking TopLink through the "createQueryByExample" query was not identified in Thought's contentions and that Thought's attempt to sweep this specific function into its disclosed assertion of code "the implements the entity manager interface" is deficient.

As above, assuming that Thought may – consistent with my October 2015 Order – rely on infringement theories that are "based on" JPA-API but not necessarily "invoked by" JPA-API, I agree that Theory 4 was not adequately disclosed. Thought admits that the specifics of how Theory 4 shows infringement is based on Oracle's proprietary extensions that go beyond the basics of JPA-API, but Thought failed to specifically identify those extensions in connection with the functionality of "communicating a request" in either its Original or Supplemental Contentions. Thought's arguments, that because it described the "functionality" of TopLink that infringed it need not have specifically identified the proprietary TopLink extensions that performed those steps and that it adequately disclosed its infringement theories because it "implicitly" disclosed its reliance on those proprietary extensions when it described that functionality, are not persuasive. As noted above, this Court's infringement contention disclosure requirements avoid implicit infringement theories by requiring express ones. Thought did not adequately disclose Theory 4.

For the foregoing reasons, Oracle's motion to strike is GRANTED. Thought cannot rely on Theories 2 – 4 to support its infringement arguments. The following are STRICKEN from the record: (i) exhibits D2-H2 and D3-F3 of Dr. Jagadish's Infringement Report; (ii) ¶ 641 of that report; (iii) references to the "Trace 2" and "Trace 3" theories on pages 229-238 of that report; (iv) all files in Exhibit K to that report, except the "Trace1" files; (v) ¶¶ 674-678 of that report; (vi) section XV of that report; and (vii) the last sentence of each of ¶¶ 84-86 of the Rebuttal Report.

II. INFRINGEMENT BY TOPLINK

Thought alleges that Oracle infringes Claims 1, 3, 5, 7, and 8 of the '197 Patent by sale and use of versions 10 – 12 of Oracle's TopLink ORM software product. Oracle argues that TopLink does not infringe because it uses a different architecture than the '197 Patent invention. According to Oracle, the '197 Patent invention requires a "split-interface" architecture to perform ORM where separate and distinct interfaces/adapters[14] are responsible for performing different functions, but the accused versions of TopLink do not have a "split-architecture" and the components identified by Thought as the first interfaces/adapters do not perform the functions the asserted claims require. Thought counters that Oracle's "split-architecture" argument was rejected by the Court on claim construction and the TopLink components it identifies perform the claims' functions.

[14] Claims 1 and 7 use the term "interfaces" which I construed to mean: "part of an adapter abstraction layer that facilitates interaction between software components." Claim Construction Order at 44. Claims 3 and 8 use the term "adapters" which I construed to mean: "runtime interchangeable part of an adapter abstraction layer that enables incompatible software components to work together." *Id.* Claim 5 (which depends from claims 1 and 3), uses both terms. Oracle asserts that under those constructions and the parties' contentions, there is no material difference in the terms for purposes of its non-infringement argument and refers to adapters/interfaces interchangeably. Oracle MSJ at 2-3 n.3. Thought does not dispute this, or otherwise point to any distinction between the terms for purposes of its Opposition.

A. Legal Standard

"Literal infringement requires the patentee to prove that the accused device contains each limitation of the asserted clai[m]." *Bayer AG v. Elan Pharm. Research Corp.*, 212 F.3d 1241, 1247 (Fed. Cir. 2000). "If any claim limitation is absent from the accused device, there is no literal infringement as a matter of law." *Id*.

B. TopLink's Alleged Infringement

1. Split Architecture

Oracle argues the asserted claims require a "split interface," because of the logical division inherent in the '197 Patent's claims. That logical division, according to the Oracle, is why the first set of adapters/interfaces has to pack data to communicate it across the division to the second set of adapters/interfaces. *See, e.g.*, Hosking MSJ Decl. ¶¶ 32 - 45. I agree that the claims require a "logical division," but to the extent that Oracle argues the division must be a physical division (*i.e.*, a system implemented across a network), that argument was rejected on claim construction. *See* Claim Construction Order at 8 ("There is nothing in the language of the claims, or in the specification, that requires packed data to be communicated across a network and I will not incorporate that limitation into the definition of packing."); *see also* Oracle MSJ Reply at 1 n.1 (recognizing a physical division is not required).

As to the "logical division," all that the claim language requires is that "software on one side of a logical divide can be programmed independent of the actions of the software on the other side." Claim Construction Order at 8. Oracle asserts that TopLink does not have that sort of logical division and the components Thought identifies as the first and second sets of adapters/interfaces "always run in the same logical space and are bound together in the same 'Java virtual machine' (the process in which a Java program executed, also referred to as a JVM)." Hosking MSJ Decl. ¶ 50; *see also* Oracle MSJ at 7 ("all relevant TopLink components run on the same computer or network and in the same process.").[15]

[15] Oracle argues this lack of logical division in TopLink is demonstrated – as discussed more below – by the fact that its software does not perform the extracting and packing that are necessary in the '197 Patent to communicate the data between the first and second sets of interfaces/adapters. Hosking MSJ Decl. ¶ 51.

However, the fact that there is a "logical division" between the first and second sets of interfaces/adapters does not mean that the components cannot operate in the same computer, network, or on the same virtual machine. Jagadish MSJ Decl. ¶ 30 ("The embodiments that were specific to a local, two-tier environment operated both the first and second interfaces/adapters in the same local address space, and therefore did not have any particular need to accommodate any physical separation."). As Thought explained at claim construction, the novelty of the '197 Patent's invention "is the fact that the two adapters or interfaces are used at runtime and one specializes in the object model and the other specializes in the relational database model." Claim Construction Order at 1-2; Thought's Opening Claim Construction Brief (Dkt. No. 70) at 3; *see also* Thought Oppo. to MSJ at 3 ("By *logically* dividing the middleware into an object-facing function and a database-facing function, CocoBase permitted runtime interchangeability without the need to alter the application source code.").[16] That functionality does not necessarily require the sort of "split architecture" that Oracle implies.

2. Functions of the First Set of Interfaces/Adapters

Related to the lack of split architecture argument above, Oracle argues that the claims require a specific allocation of the "extracts and instantiates" and "packs and unpacks" tasks between two *different* interfaces/adapters, and that the alleged first interface/adapter Thought identifies in TopLink does not perform the required functions of: (1) extracting content from an object; (2) packing that content as data; (3) communicating that packed data to the second adapter; (4) after receiving packed database content from the second adapter, unpacking the packed database content; and (5) instantiating that unpacked content into a new object.[17]

[16] Oracle's own expert does not opine that the language of the claims prevents the two sets of interfaces/adapters from operating in the same computer or virtual machine, but instead points to examples from the '197 Patent specification showing an implementation across a distributed network. Hosking MSJ Decl. ¶¶ 36, 38.

[17] Thought identifies the first interface/adapter as JPA's EntityManager, TopLink's JpaEntityManager and TopLink's EntityManagerImpl. Jagadish MSJ Decl. ¶ 78.

a. extract and instantiate by first interface/adapter[18]

Claim 1 requires the first interface to extract “the object attributes and object name from the object application” and then (upon the return from the second interface), unpack the data “to effect instantiating the object attributes and the object name into a new object.” ’197 Patent, 35:10-15. Claim 7 requires an adapter abstraction layer which “extracts the object attributed and the object name from the object” and then (upon return from the second interface), performs “instantiating the object attributes and the object name into at least one new object and/or status.” ’197 Patent 36:23-28.

Oracle argues that the EntityManager component identified by Thought as the first interface/adapter cannot infringe because EntityManager (nor any other component identified by Thought) does not instantiate a new object based on the retrieved data. Oracle argues that Thought’s reliance on a different TopLink component – DatabaseAccessor – as the instantiator does not work because that component is not the first interface/adapter. Instead, according to Oracle, the DatabaseAccessor passes the instantiated object back to the EntityManager. Hosking MSJ Decl. ¶ 57.

Thought responds that the claims do not require the first interface/adapter to instantiate itself, but only to “effect instantiation.” ’197 Patent 35:14-15 (“said first interface unpacking data to effect instantiating the object”); 35:37-38 (“said first adapter unpacking the data to effect instantiating”). According to Thought, as long as the first interface/adapter causes some other component to instantiate the new object, the claim is covered. Thought MSJ Oppo. at 5-7. On this point, Thought’s expert contends that unidentified components “work” with EntityManager (or other components identified as the first interface/adapter like JPAEntityManager and EntityManagerImpl) to effect and carry out the instantiation. Jagadish MSJ Decl. ¶ 78. In its Opposition brief, Thought identifies “ObjectBuilder” as the component that instantiates, but that component was not discussed by Jagadish in his summary judgment declaration. Thought MSJ Oppo. at 7.

Thought also asserts that the ’197 Patent disclosed the principle of having two components

[18] The terms extract and instantiate were not construed.

(in the patent specification identified as CocoPowder and CocoBeansImpl) perform the functions of the second adapter. Thought MSJ Oppo. at 6. Therefore, it contends that nothing in the claims or specification prevents the functions of the first or second interfaces/adapters from being performed by more than one component.

In response and at oral argument, Oracle pointed out that even if two components could perform these tasks,[19] Thought failed to identify the components that instantiate in conjunction with EntityManager, JpaEntityManager, or EntityManagerImpl. *See* Reply MSJ at 2; *see also* Hosking MSJ Decl. ¶ 57 ("But neither the TopLink component used to implement the EntityManager nor any of the other alleged first interfaces or adapters instantiates the requested object based on the retrieved database content. In TopLink, instantiation is done by a different TopLink component, namely, a "DatabaseAccessor." The DatabaseAccessor passes the instantiated object back to the implementation of the EntityManager and the other alleged first interfaces or adapters.").[20]

I conclude that even if Thought is correct that two components can perform the steps at issue – extract and then effect instantiation – under the teaching of the claims, Thought's expert does not identify the components that perform those functions sufficient to rebut Oracle's assertion that TopLink does not have a first interface/adapter that performs the claimed functions. Once Oracle submitted evidence that the components Thought identified as TopLink's first interface/adapter did not perform these functions, the burden shifted to Thought to present *evidence* that at least raised a genuine issue of fact that components it identifies did. *See,*

[19] At oral argument, Oracle abandoned its argument that two components could not perform these tasks as part of the first interface/adapter under the claim language.

[20] While Thought identified and relied on DatabaseAccessor in its infringement proofs, as Jagadish admits, Database Accessor was only identified in connection with the *second adapter*. Jagadish MSJ Decl. ¶ 78; *see also* Thought Summary Judgment slides at 110 (referring to Jagadish Infringement Report at 389/6705, disclosing that the DatabaseAccessor only "manipulates and passes data."). At claim construction, Thought relied heavily on the idea that the "two adapters or interfaces" played different functions where one specializes in the object model and the other specializes in the relational database model. Claim Construction Order at 1-2; Thought's Opening Claim Construction Brief (Dkt. No. 70) at 3. Thought cannot attempt to combine the functions performed by the second interface/adapter into the functions performed by the fist interface/adapter to attempt to avoid summary judgment.

e.g.,Optivus Tech., Inc. v. Ion Beam Applications S.A., 469 F.3d 978, 990 (Fed. Cir. 2006). There is no evidence from Jagadish that ObjectBuilder is part of or invoked by the first interface/adapter and as Jagadish admits, DatabaseAccessor is part of the second interface/adapter.

b. packs and unpacks by first interface/adapter

I construed packing as "placing data into a form suitable for communication between software components" and unpacking as "retrieving data from a packed data structure." Claim Construction Order at 8. Oracle argues that the component Thought identifies as the first interface/adapter does not perform both of these tasks. Further, as to unpacking, Oracle argues that the component Thought identifies as performing the unpacking step (valueFromRow) is entirely different from the EntityManager identified by Thought at the first interface/adapter and is never invoked by any component Thought has identified as the first interface/adapter. Hosking Decl. ¶¶ 55, 58.

In response Thought relies on its argument that the unpacking function can be performed by two components, and relies again on valueFromRow. Thought MSJ Oppo. at 7. However, as above, Thought presents no *evidence* from its expert to show that valueFromRow is *invoked* by the components identified as the first interface/adapter to unpack. Thought fails to rebut with evidence Oracle's evidence sufficient to defeat summary judgment.[21]

3. No packing/unpacking

Oracle argues that – even if Thought is correct that more than one component can combine to perform the packing/unpacking steps – no components identified by Thought actually perform the packing/unpacking contemplated by the claims. As noted above, "packing" was defined as placing the data in a form "suitable for communication between software components" and unpacking as "retrieving data from a packed data structure." Oracle asserts that packing is not

[21] I recognize that valueFromRow was identified and relied on by Jagadish in his Infringement Report. *See, e.g.*, Jagadish Infringement Report at pgs. 390, 2004-05; Thought Summary Judgment slide at 111. The problem, however, is that Thought does not cite evidence from Jagadish or elsewhere showing that this component is invoked by what it has identified as the first interface/adapter to unpack. The discussion at pages 2004-05 of the Infringement Report discuss valueFromObject as performing extracting with respect to the stricken Trace 2 theory of infringement.

United States District Court
Northern District of California

necessary to communicate data among components running on the same computer and in the same software process, as happens in TopLink. Oracle MSJ Reply at 8. Instead, in TopLink, Oracle contends that data is simply passed between components "as is" without performing the "extra step" of packing data in a suitable form for communication. *Id*.; Hosking MSJ Decl. ¶ 51. Oracle argues that, given the Court's Claim Construction, the form of the data must be manipulated somehow relevant to communication to infringe, and criticizes Thought's alleged attempt to equate packing with the function of setting the value of data to have the value of another object. Hosking MSJ Decl. ¶ 54c. Thought's equation of packing with setting values is not equivalent to "packing," according to Oracle, because there is no change to form *necessary* for communicating data between the components, as required by the claim language. Setting the value of data is simply not the same as "manipulating" the data for communications purposes as required by the claims.[22]

Thought responds that Oracle is over-simplifying its position; Thought asserts that "packing" is performed when the extracted object names and attributes are "set in a newly instantiated query object, in a form the query object requires for its internal structures." Thought Oppo. at 8 (relying on the "findInternal" method of EntityMangerImpl); *see also* Jagadish MSJ Decl. ¶ 72 (an example of packing includes "a query structure as the form in which the packed data is provided on the trip toward the database, and a result set as the form on the return trip; placing the necessary information where it is expected within those datastructures does not necessarily dictate a change in the data, just placing the data in the forms suitable for the respective communication events between software components "). In Reply, Oracle argues that there is still no dispute that TopLink does not change the "form" the data is passed to the query in and that the data is simply passed along in the same form as it was received.

Thought fails to raise an issue of material fact as to packing. It does not address the claim language's requirement that "packing" requires some manipulation in the "form" of the data *to*

[22] Oracle also relies on the description the '197 Patent's inventor, Ward Mullins, made when attempting to sell his software that "packing" means processing the object into more "primitive data" types. Artuz Decl., Ex. 6 at THOR00086541. That description, however, is just one example of packing encompassed by the claim language.

facilitate communication. Jagadish does not explain how "setting" data in a "query structure" in the trip towards the database and then "setting" the data as a "result set" on the return trip manipulates the form of the data for purpose of communication, as opposed to simply arranging the identical data in two different ways. The claim language requires something more than simply "arranging" existing data, it requires a *manipulation* of its form.

4. No Extracting . . . from Object

Oracle contends that "extracting" attributes "from Object" requires the component to take only part of an object's attributes out for packing and communicating to the second adapter/interface. TopLink does not infringe, according to Oracle, because its components pass the whole object along, and do not "extract" anything. Hosking MSJ Decl. ¶¶ 54a, 54b. Oracle notes that in Thought's Theory 1 (Trace 1) of infringement, that relies on the JPA standard as opposed to proprietary TopLink APIs, TopLink uses the "find()" method to extract, but that method receives *two* objects, as opposed to *one* object required by the claims. '197 Patent 35:64 – 36:2 ("an object comprising object attributes and an object name"); Hosking MSJ Decl. ¶ 54b.

Thought responds that "extracting" can mean obtaining, and – given the nature of how data functions in the digital world – a complete copy can be made of an object, and that should still be considered "extraction." According to Thought, the claims require no more "than the object is the source of the object attributes and name." Thought MSJ Oppo. at 10. Thought also disagrees that the "extraction" performed by the claims must be from only one Java object, as opposed to the two that are used by the find() method relied on by Thought in Jagadish's infringement Theory 1. Thought contends that find() method extracts object attributes from one object and that one object contains "references to two other objects." According to Thought, Oracle's disagreement on whether an object which contains references to two other objects is "an object" as used by the claims is a dispute of fact. *Compare* Jagadish MSJ Decl. ¶¶ 65-68 (the "object to which my infringement proofs point, in the find() theory, is in fact an instance of a class"); *with* Hosking MSJ Decl. ¶ 54b.

In Reply, Oracle argues that Thought is attempting to construe "extracting" at summary judgment as "obtaining," which is not only procedurally improper but not supported by the claim language which "plainly requires" a subset of information to be acquired from an object so that it could be packed. Oracle also contends that Thought is impermissibly attempting – by relying on its reference to an object that contains references to two other objects – to change the agreed-to construction of object from "instance of a class" to "instance of a template" which are not synonymous. Hosking MSJ Decl. ¶ 54b.

As to the first argument, I agree with Oracle; "extracting" as supported by the claim language refers to extracting a *subset* of information from an object.[23] While Thought's expert explains that in the data processing realm "extracting" can mean wholesale copying of all of an object's information (for example, a query that asks for production of all employee information from a database containing only employee information), that unremarkable concept does not help Thought here. That sort of wholesale copying of all information from an object, or passing along the whole object, is simply not contemplated by either the claim language or shown in the preferred embodiments, which require and show "extracting" a subset of information in order for it to be packed for communication.[24]

As to the second argument, Thought characterizes this as a clash of the experts, with Jagadish arguing that he has shown how two components effect extraction[25] and Hosking arguing those two components could never be considered "one object."[26] In its rebuttal Summary Judgment slides, Thought focuses on a slightly different theory; instead of an object containing

United States District Court
Northern District of California

[23] This is true regardless of whether "extract" is defined as "pulling from" (proposed by Oracle) or "obtained from" (as proposed by Thought).

[24] At oral argument, in an attempt to raise a material issue of fact, Thought relied on parts of its Trace 2 and Trace 3 theories. Thought Summary Judgment slides at 92-94; *see also* Hosking MSJ Decl. ¶ 70. Those theories, however, have been stricken and cannot be invoked to defeat summary judgment.

[25] Namely, that extraction as contemplated by the claims is achieved in TopLink where the object name in the form of "Class<T> entityClass" is extracted from the "object" that is a "set of parameters" and the object attributes in the form of "ObjectprimaryKey" are extracted from the "object" that is a "set of parameters." Thought Summary Judgment slides at 99 (citing Jagadish Infringement Report at pg. 378); *see also* Jagadish MSJ Decl. ¶¶ 61, 62.

[26] According to Hosking, the two objects identified by Thought would never be considered one object because they are not instances of the same class. Hosking MSJ Decl. ¶ 54b; Thought Summary Judgment slides at 99.

references to two other objects meeting the "extract . . . from the object" claim requirements, it is the "set of parameters" that forms the one object. *See* Thought Rebuttal Summary Judgment slides at 6. I conclude that Thought cannot defeat summary judgment relying on continuously-shifting theories. As noted above, "object" has been defined as an instance of a class. Thought provides no explanation to show that that Court-accepted definition encompasses *either* an object which necessarily (to read on the claims) contains a reference to another two objects or an object that is "a set of parameters."[27] Thought has not raised a material issue of fact to rebut Oracle's evidence that TopLink does not "extract" a subset of information – namely an object name and then object attributes – from a singular "instance of a class."

C. TopLink and Other Products

Oracle argues that because TopLink does not infringe, none of the other accused products infringe. Thought responds not by pointing out distinctions between TopLink and the other products, but instead by asserting that because TopLink infringes Oracle's twelve other accused middleware products necessarily infringe because they are packaged and sold with TopLink. Having concluded that Thought failed to raise a material issue of fact precluding summary judgment of non-infringement as to TopLink, none of the other Oracle products packaged and sold with TopLink can be accused of infringement.

For the foregoing reasons, Thought fails to raise material issues of fact as to a number of claim elements in order to avoid summary judgment of non-infringement. Summary judgment of non-infringement, therefore, is granted to Oracle.[28]

[27] Hosking explains that an object is an instance of a class and a class is "simply a way to group similar things." Hosking MSJ Decl. ¶¶ 5, 6. Jagadish agrees. Jagadish MSJ Decl. ¶ 63 ("the collections of objects that are instances of a particular template are said to form a class."). "Parameter" is defined as "the name given to the placeholder in the definition of a method for an input to a method." Hosking MSJ Decl. ¶ 14. Jagadish criticizes what he characterizes as Hosking's limitation of the definition of "class" and "object" to Java programming language (Jagadish MSJ Decl. ¶¶ 66-67), but does not explain why the use of an object containing two references to additional objects covers the claim language of "extracting the object attributes and the object name from the object."

[28] As such, I need not determine whether the sale of TopLink constitutes direct infringement if Claims 1, 3, and 5 of the '197 Patent contain "functional" language (as Thought contends) or whether Thought needs to show actual proof of TopLink being used in an infringing mode.

III. INVALIDITY

Oracle moves for summary judgment on its counterclaim of invalidity of the ’197 Patent. Oracle claims that a prior version of TopLink, TopLink for Smalltalk 2.0 or TLS2, developed in Canada in 1994 using the Smalltalk programming language is anticipating or obviating prior art. Thought also moves for partial summary judgment on Oracle’s “combination” of prior art obviousness theories, arguing that portions of Oracle’s expert’s reports should be struck as improper and after striking, Oracle is left without an obviousness defense.

Having granted Oracle’s summary judgment motion on non-infringement, I need not consider Oracle’s declaratory relief counterclaim on invalidity. *See Liquid Dynamics Corp. v. Vaughan Co., Inc*., 355 F.3d 1361, 1371 (Fed.Cir.2004) (“A district court judge faced with an invalidity counterclaim challenging a patent that it concludes was not infringed may either hear the claim or dismiss it without prejudice, subject to review only for abuse of discretion.”); *Augme Techs., Inc. v. Yahoo! Inc*., No. C-09-5386 JCS, 2012 WL 3627408, at *1 (N.D. Cal. Aug. 21, 2012). Here, the non-infringement judgment resolves this case and appears to extinguish any threat of future enforcement of the claims or litigation against Oracle.

Likewise, I need not reach Thought’s motion seeking to strike portions of Hosking’s Expert Report on obviousness as a result of combinations of prior art and “dismiss” on summary judgment Oracle’s obviousness defenses and counterclaim. That motion is DENIED as moot.

IV. EVIDENTIARY MOTIONS AND MOTIONS TO FILE UNDER SEAL

Oracle objects and moves to strike portions of the Rebuttal Declaration of Dr. Hosagrahar V. Jagadish on summary judgment (Jagadish MSJ Decl.). Dkt. No. 221. Oracle objects to statements in the Jagadish MSJ Declaration that Oracle alleges: (i) go beyond his expert reports; (ii) constitute improper claim construction; and (iii) recite infringement theories subject to Oracle’s motion to strike. I have considered each objection and DENY the motion as to statements that allegedly go beyond the Jagadish reports and constitute claim construction. I GRANT the motion as to the two discrete statements that go to infringement theories subject to the motion to strike; ¶¶ 70, 109 (2nd sentence).

Thought likewise objects to portions of the summary judgment Declaration of Dr. Antony

L. Hosking that disclose new opinions or claim constructions, as well as portions of the Declaration of Dennis Leung that are not based on personal knowledge or contain improper lay opinion testimony. Dkt. No. 225. I have considered each objection and DENY the motion as to the Hosking statements which allegedly disclose new opinions or claim constructions and DENY the objections to portions of the Leung Declaration as moot.[29]

Oracle's Motion to Seal Exhibits in Support of its Motion for Summary Judgment. Dkt. No. 204. Oracle moves to seal Exhibit 6 to the Declaration of Robert J. Artuz, which is an email chain between Ward Mullins and Rembrandt IP management that was designated as AEO by Thought. Thought has not submitted a declaration explaining why this exhibit merits sealing under the compelling justifications standard. The Motion to Seal Dkt. No. 204-4 is DENIED. The clerk shall unseal Dkt. No. 204-4.

Oracle moves to seal Exhibits 11, 12, 13, and 21 to the Artuz Declaration containing excerpts from the expert reports of Drs. Jagadish and Hosking, because substantial portions of those Exhibits disclose software source code. Relatedly, Oracle moves to seal Exhibits 14, 15, 16, 17, 18, 19, 20, and 27 to the Artuz Declaration containing excerpts of the exhibits to those expert reports, because those exhibits disclose source code. According to the Declaration of Matthew Ahn (Dkt. No. 204-2), substantial portions of Exhibits 11-21 and 27 comprise confidential, non-public source code and product design information that is highly confidential to Oracle. The Motion to Seal these Exhibits (Dkt. Nos. 204-5 through 204-16) is GRANTED.

Oracle's Motion to Seal Exhibit in Support of Motion to Strike. Dkt. No. 207. Oracle moves to seal Exhibit 6 to the Declaration of Steven D. Moore in Support of Oracle's Motion to Strike containing excerpts of the Jagadish Expert Report which disclose source code. According to the Declaration of Matthew Ahn (Dkt. No. 207-1), all or significant portions of the Jagadish Expert Report discuss or disclose highly confidential source code of the Oracle TopLink product, or non-public details about the operation and organization of that source code. The Motion to Seal Exhibit 6 (207-3) is GRANTED.

[29] The Leung Declaration goes only to Oracle's invalidity arguments, which I have declined to reach.

Oracle's Motion to Seal Exhibits in Support of Oracle's Opposition to Thought's Motion for Partial Summary Judgment. Dkt. No. 210. Oracle moves to seal portions of Exhibits 6 and 7 to the Declaration of Kevin J. O'Brien in Support of Oracle's Opposition to Thought's Motion for Partial Summary Judgment. Exhibit 6 is the Hosking Expert Report, and the portions which Oracle seek to file under seal (according to the Declaration of Matthew Ahn, Dkt. No. 210-1) contain or discuss portions of confidential source code for Oracle's TopLink and Smalltalk products and contain non-public product design information that are highly confidential. Oracle's Motion to Seal portions of Exhibit 6 (Dkt. Nos. 210-6, 210-7, and 210-8) is GRANTED.

Pages 21-26 of Exhibit 7 are portions of an exhibit to the Hosking Expert Report that according to Ahn contain highly confidential, non-public product design information concerning the TopLink and Smalltalk products. Oracle's Motion to Seal portions of Exhibit 7 (Dkt. No. 210-10) is GRANTED.

Thought's Motion to Seal Portions of Opposition and Exhibits in Support of Opposition to Oracle's Motion for Summary Judgment. Dkt. No 212. Thought seeks to seal portions of its Opposition brief, Exhibits 5 & 6 to the Declaration of Mark Carlson, and the Declaration of Dr. Jagadish in Opposition to the Motion for Summary Judgment, as these documents or portions of documents reference Oracle's source code. According to Oracle's declarant, Matthew Ahn, the only portions of these documents that should remain sealed because they disclose highly confidential source code and non-public product design information are: (i) pages 001590-002241 of Exhibit 5 (Dkt. No. 212-5; ECF Pgs. 11-112); and (ii) the graphic on page 9 of the Opposition (Dkt. No. 212-4). Therefore, the Motion to Seal is GRANTED in part: Docket Nos. 212-4 and 212-5 shall REMAIN under seal. The clerk shall UNSEAL Dkt. Nos. 212-6 and 212-7. Within 10 days of the date of this Order, Thought shall file a redacted version of Exhibit 5, redacting only pages 001590-002241 and a redacted version of the Opposition to the Motion for Summary Judgment, redacting only the graphic on page 9.

Thought's Motion to Seal Portions of Opposition and Exhibits in Support of Opposition to Oracle's Motion to Strike. Dkt. No. 216. Thought seeks to seal portions of its Opposition brief, Exhibits 10-12 to the Declaration of Mark Carlson, and the Declaration of Dr. Jagadish in

United States District Court
Northern District of California

Opposition to the Motion to Strike, as these documents or portions of these documents reference Oracle's source code. According to Oracle's declarant, Matthew Ahn, the only portions of these documents that should remain sealed because they disclose highly confidential source code and non-public product design information are: (i) the graphics on pages 17 and 19 of the Opposition, (ii) Exhibit 10 to the Carlson Declaration; and (iii) Exhibit 11 to the Carlson Declaration. Therefore, the Motion to Seal is GRANTED in part: Docket Nos. 216-3, 216-6, and 216-7 shall REMAIN under seal. The clerk shall UNSEAL Dkt. Nos. 216-5 and 216-8. Within 10 days of the date of this Order, Thought shall file a redacted version of its Opposition to the Motion to Strike, redacting only the graphics on pages 17 and 19.

Thought's Motion to Seal Portions of its Objections to Evidence. Dkt. No. 224-2. Thought seeks to seal Exhibits 1 and 2 to its Objections to Evidence (tables discussing excerpts from the Jagadish expert reports) and Exhibit 3 (the Hosking MSJ Declaration). The MOTION is GRANTED in part: For the reasons described by Matthew Ahn, all or significant portions of the Jagadish expert reports discuss or disclose highly confidential source code of the Oracle TopLink product, or non-public details about the operation and organization of that source code. Dkt. Nos. 224-3 and 224-4 shall REMAIN under seal. However, Oracle has not objected to the public filing of the Hosking MSJ Declaration, therefore, the clerk shall UNSEAL Dkt. No. 224-5.

CONCLUSION

For the foregoing reasons, summary judgment of non-infringement is GRANTED to Oracle. Oracle's counterclaim seeking a declaratory judgment that the '197 patent is invalid is dismissed without prejudice.

IT IS SO ORDERED.

Dated: June 13, 2016

WILLIAM H. ORRICK
United States District Judge