Tharan Gregory Lanier, Bar No. 138784 (Lead Counsel)
tglanier@jonesday.com
Evan M. McLean, Bar No. 280660
emclean@jonesday.com
JONES DAY
Silicon Valley Office
1755 Embarcadero Road
Palo Alto, California 94303
Telephone: +1.650.739.3939
Facsimile: +1.650.739.3900

Michael A. Lavine, Bar No. 321727
mlavine@jonesday.com
JONES DAY
555 California Street, 26th Floor
San Francisco, CA 94104
Telephone: +1.415.626.3939
Facsimile: +1.415.875.5700

Sachin M. Patel (*pro hac vice* forthcoming)
smpatel@jonesday.com
JONES DAY
110 North Wacker Drive, Suite 4800
Chicago, IL 60606
Telephone: +1.312.782.3939
Facsimile: +1.312.782.8585

Peter Young Kim (*pro hac vice* forthcoming)
pkim@jonesday.com
JONES DAY
51 Louisiana Ave, N.W.
Washington DC, 20001
Telephone: +1.202.879.3433
Facsimile: +1.202.626.1700

Attorneys for Plaintiff Google LLC

UNITED STATES DISTRICT COURT

NORTHERN DISTRICT OF CALIFORNIA

GOOGLE LLC, **Plaintiff,** **v.** **K.MIZRA LLC,** **Defendant.**	**Case No.** **COMPLAINT FOR DECLARATORY JUDGMENT OF NON-INFRINGEMENT OF U.S. PATENT NO. 8,144,717 AND U.S. PATENT NO. 8,438,120** **JURY TRIAL DEMANDED**

Plaintiff Google LLC ("Google") files this Complaint for Declaratory Judgment of Non-Infringement against Defendant K.Mizra LLC ("K.Mizra") and in support of its Complaint alleges as follows:

NATURE OF THE ACTION

1. This action against K.Mizra for declaratory judgment of non-infringement of U.S. Patent Nos. 8,144,717 ("the '717 Patent") (Ex. 1) and 8,438,120 ("the '120 Patent") (Ex. 2) (collectively, "Patents-in-Suit") is pursuant to the Declaratory Judgment Act, 28 U.S.C. §§ 2201–2202, and arises under the patent laws of the United States, Title 35 of the United States Code.

2. Google seeks a declaration that it and its products do not infringe any claim of the Patents-in-Suit under the patent laws of the United States, 35 U.S.C. § 1 *et seq.* Google also seeks any other remedy or relief that this Court deems just and proper.

3. K.Mizra is a non-practicing entity. K.Mizra does not innovate, make, or sell any products. Instead, K.Mizra buys patents, identifies companies as potential targets for licensing, and threatens potential targets with the option to become K.Mizra's licensees or otherwise face litigation. *See Who Are We?*, K.Mizra, *available at* https://kmizra.com/who-we-are (last visited September 24, 2025). K.Mizra sues potential targets who do not submit readily to K.Mizra's licensing demands. *See id.* As K.Mizra's website states: "K.Mizra is very proactive. We are currently engaged in patent litigation in the US and Europe. We also provide technical discussions surrounding our portfolio and a potential licensees' [sic] products or services prior to enforcement. We offer licenses to our patents. . . ." *Id.* K.Mizra's solicitation and word-of-mouth marketing of its licenses are directly tied to threats of litigation. *See id.*

4. K.Mizra has two primary endeavors: (1) patent licenses; and (2) patent litigation suits. Last year, K.Mizra's CEO, Chuck Hausman, stated that "companies fight the idea of taking a licence until we litigate[.]" *Meet K.Mizra, The NPE With a Seasoned CEO at the Helm*, IAM (Dec. 11, 2024), *available at* https://www.iam-media.com/article/meet-kmizra-the-npe-seasoned-ceo-the-helm. The threat of a lawsuit is K.Mizra's tactic to obtain revenue.

5. K.Mizra describes its litigation strategy on its website. For example, the website's menu bar includes a tab titled "Litigation Current" that lists K.Mizra's currently pending patent

litigation against various companies.



See Home, K.Mizra, *available at* https://kmizra.com/. Under the tab "Archives," K.Mizra identifies its past patent litigations.

6. K.Mizra sued Google in the United States District Court for the Western District of Texas. In that case, K.Mizra sued Google for alleged infringement of U.S. Patent Nos. 8,234,705 ("the '705 Patent") and 9,516,048 ("the '048 Patent") (collectively "Texas Asserted Patents"). *See K.Mizra LLC v. Google LLC*, No. 1:25-cv-00236 (W.D. Tex. Feb. 18, 2025) ("the Texas action"). K.Mizra lists the Texas action on its website. *See Litigation Current: K.Mizra LLC v. Google LLC*, K.Mizra, *available at* https://kmizra.com/k-mizra-v-google-1 (last visited September 24, 2025).

7. The Texas Asserted Patents are not the Patents-in-Suit. This declaratory judgment action relates to the '717 and '120 Patents, which are listed under K.Mizra's North American patent portfolio ("Patent Portfolio"). *See North America*, K.Mizra, *available at* https://kmizra.com/north-america (last visited September 24, 2025) (listing the Patents-in-Suit among others).

FACTUAL ALLEGATIONS

8. K.Mizra directed communications to a first Google employee located in Northern California three years ago. On or about June 6, 2022, K.Mizra's CEO directed communications to

that Google employee, soliciting the employee's availability to schedule a call to discuss an issue potentially relevant to Google.

9. The next day, on or about June 7, 2022, a second Google employee located in Northern California confirmed receipt of K.Mizra's CEO's correspondence inviting Google to discuss a K.Mizra matter on a call. The second Google employee also provided K.Mizra's CEO with a proposed non-disclosure agreement to help facilitate the discussions.

10. From June 2022 to February 2023, K.Mizra continued directing communications to Google employees located in California. On or about March 7, 2023, K.Mizra's CEO directly communicated to a third Google employee that K.Mizra sought Google to take a license to K.Mizra's Patent Portfolio. During this time, K.Mizra's Patent Portfolio already included the Patents-in-Suit.

11. Nearly two years later, on or about March 6, 2025, K.Mizra's CEO again sent correspondence to the first Google employee located in Northern California, stating that K.Mizra filed a lawsuit against Google a few weeks before, and that K.Mizra was open to a discussion with Google. K.Mizra's CEO was expressly referring to the Texas action, which was filed in February 2025.

12. From April 2025 until present, K.Mizra's CEO and a fourth Google employee, who is located in Northern California, have discussed K.Mizra's license demand for its Patent Portfolio several times.

13. On May 23, 2025, in the Texas action, K.Mizra's counsel sent an email to Google's outside counsel requesting leave to amend K.Mizra's complaint. K.Mizra's outside counsel stated that K.Mizra "recently completed its factual investigation concerning Google's infringement of" the '717 and '120 Patents and "[f]or the sake of efficiency, would like to assert those claims in [the Texas] action." Ex. 3.

14. On May 28, 2025, K.Mizra's counsel provided a draft amended complaint setting forth K.Mizra's infringement allegations accusing Google's Vertex AI Studio of infringing the '120 Patent and various Google Nest products of infringing the '717 Patent. *See* Ex. 4 (draft amended complaint); Ex. 5 (complaint redline provided by K.Mizra). K.Mizra expressed its plan

to assert the Patents-in-Suit (which are different from and unrelated to the Texas Asserted Patents) against Google products that are entirely different from the accused products in the Texas action.

15. On Monday, June 9, 2025, K.Mizra's counsel emailed Google's counsel, explaining that K.Mizra "decided against amending their complaint at this time." Ex. 6.

16. During the same time, K.Mizra's CEO continued directing correspondence to the fourth Google employee located in Northern California, seeking to discuss a license for K.Mizra's Patent Portfolio, including the Patents-in-Suit.

17. On June 16, 2025, K.Mizra's CEO emailed Google's fourth employee located in Northern California, offering a license that included the Patents-in-Suit.

18. Given K.Mizra's current lawsuit against Google (the Texas action), K.Mizra's recent threats of further pursuing an infringement action against Google for allegedly infringing the Patents-in-Suit, and K.Mizra's business practice of suing companies, including companies located in California, that do not agree to K.Mizra's terms for a patent license, there exists a substantial controversy between K.Mizra and Google that involves adverse legal interests and is of sufficient immediacy and reality.

19. K.Mizra's actions place a cloud of uncertainty on the commercial use of Google's products; threaten Google's business and relationships with its customers and partners; and create a justiciable controversy between Google and K.Mizra. Google thus brings this action to obtain a declaratory judgment that Google does not infringe any claim of the Patents-in-Suit, directly or indirectly, literally, or under the doctrine of equivalents.

PARTIES

20. Google is a limited liability company organized and existing under the laws of the state of Delaware, with its principal place of business at 1600 Amphitheatre Parkway, Mountain View, California 94043.

21. On information and belief, K.Mizra is a Delaware limited liability company with its principal place of business at 777 Brickell Avenue, #500-96031, Miami, Florida 33131.

JURISDICTION AND VENUE

22. This action arises under the Federal Declaratory Judgment Act, 28 U.S.C. § 2201 *et*

seq., and under the patent laws of the United States, Title 35 of the United States Code.

23. The Court has subject matter jurisdiction over the claims alleged in this action under 28 U.S.C. §§ 1331, 1338(a), and 2201(a), because this action involves claims arising under the patent laws of the United States, 35 U.S.C. § 1 *et seq.* For the reasons stated herein, K.Mizra has created a real, live, immediate, and justiciable case or controversy between K.Mizra and Google with regard to alleged infringement of the claims in the Patents-in-Suit.

24. This Court has personal jurisdiction over K.Mizra, pursuant to due process and/or the California Long Arm Statute because K.Mizra has had continuous and systematic business contacts in this District, has purposely directed its business activities at residents of this District, and has conducted business activities in this District relating to the Patents-in-Suit. On information and belief, K.Mizra has purposefully directed its patent licensing and enforcement activities with respect to the Patent Portfolio, which includes the Patents-in-Suit, to companies having principal places of business and/or operations in this judicial district, including at least Google, Broadcom, Cisco, Epson America, Forescout, Fortinet, HP Inc., and SonicWall, including alleging the infringement of one or more patents in K.Mizra's Patent Portfolio across the United States including in this District. The claims asserted herein arise out of or relate to activity by K.Mizra within and directed at this forum. K.Mizra's minimum contacts with this forum allow this Court to exercise personal jurisdiction over K.Mizra and do not offend any traditional notions of fair play and substantial justice.

25. On information and belief, K.Mizra has conducted business in this District through its commercial, licensing, and enforcement activities relating to its Patent Portfolio, which includes the Patents-in-Suit. For example, since at least 2022, K.Mizra's CEO has directed numerous communications to Google, which resides in this District, as well as specifically to its employees located in this District. *See*, *e.g.*, *supra*, ¶¶ 8–18. K.Mizra was and remains aware that Google maintains its headquarters in Mountain View, California, and averred as such in the Texas action. Ex. 4 ¶ 14.

26. On information and belief, K.Mizra has intentionally directed activities and licensing communications relating to its Patent Portfolio, which includes the Patents-in-Suit, to the

State of California. In the draft amended complaint it provided to Google's counsel, K.Mizra alleged that each of the Patents-in-Suit have "previously been licensed to various industry participants." Ex 4 ¶¶ 50, 70. On information and belief, K.Mizra's portfolio licenses (including the Patents-in-Suit) have been realized for numerous California residents, including Broadcom,[1] Cisco,[2] Epson America,[3] Forescout,[4] Fortinet,[5] HP,[6] and SonicWall.[7] *See, e.g.*, *Meet K.Mizra, The NPE With a Seasoned CEO at the Helm*, IAM (Dec. 11, 2024), *available at* https://www.iam-media.com/article/meet-kmizra-the-npe-seasoned-ceo-the-helm ("Hausman learned at Philips that

[1] *See generally* Complaint, *K.Mizra LLC v. Broadcom Inc.*, No. 2:21-cv-00247 (E.D. Tex. July 8, 2021), *available at* https://kmizra.com/k-mizra-v-broadcom-inc (last visited September 24, 2025); *see also Contact Us*, Broadcom, *available at* https://www.broadcom.com/company/contact/locations ("California, Palo Alto (Corporate Headquarters)") (last visited September 24, 2025).

[2] Complaint, ¶ 6, *K.Mizra LLC v. Cisco Sys., Inc.*, No. 6:20-cv-01031 (W.D. Tex. Nov. 6, 2020), *available at* https://kmizra.com/k-mizra-v-cisco-systems ("Defendant Cisco is a California Corporation. . . .") (last visited September 24, 2025); *see also Facts and information*, Cisco, *available at* https://www.cisco.com/c/en/us/about.html (Legal mailing address[:] "Cisco Systems, Inc.[,] 170 West Tasman Dr.[,] San Jose, CA 95134 USA") (last visited September 24, 2025).

[3] Complaint for Patent Infringement, ¶ 9, *K.Mizra LLC v. Epson Am., Inc.*, No. 8:25-cv-00027 (C.D. Cal. Jan. 7, 2025), *available at* https://kmizra.com/k-mizra-v-epson ("Defendant Epson America, Inc. is a company organized and existing under the laws of the state of California. . . .") (last visited September 24, 2025).

[4] Complaint, ¶ 5, *K.Mizra LLC v. Forescout Technologies, Inc.*, No. 2:21-cv-00248 (E.D. Tex. July 8, 2021), *available at* https://kmizra.com/k-mizra-v-forescout-tech ("Defendant Forescout is a California Corporation. . . .") (last visited September 24, 2025); *see also Global Presence*, Forescout, *available at* https://www.forescout.com/company/contact-us/ (indicating with a blue star on a world map the Northern California address of "300 Santana Row[,] Suite 400[,] San Jose, CA 95128") (last visited September 24, 2025).

[5] *See generally* Complaint, *K.Mizra LLC v. Fortinet, Inc.*, No. 2:21-cv-00249 (E.D. Tex. July 8, 2021), *available at* https://kmizra.com/k-mizra-v-fortinet (last visited September 24, 2025); *see also Fortinet Global Offices*, Fortinet, *available at* https://www.fortinet.com/corporate/about-us/global-offices ("US Headquarters[,] 909 Kifer Road[,] Sunnyvale, CA 94086") (last visited September 24, 2025).

[6] *See generally* Complaint for Patent Infringement, *K.Mizra LLC v. HP Inc.*, No. 2:21-cv-00226 (E.D. Tex. June 22, 2021), *available at* https://kmizra.com/k-mizra-v-hp-inc-3 (last visited September 24, 2025); *see also The company that literally started Silicon Valley is moving to Texas*, CNN (Dec. 2, 2020, 11:45 AM), *available at* https://www.cnn.com/2020/12/02/tech/silicon-valley-hpe-to-texas ("In 2015, the company split into HPE and *hardware maker HP Inc. (HPQ), which is not heading to Texas*." (emphasis added)) (last visited September 24, 2025); Press Release, *HP Inc. Reports Fiscal 2025 Second Quarter Results*, HP.com (May 28, 2025), *available at* https://www.hp.com/us-en/newsroom/press-releases/2025/hp-inc-reports-fiscal-2025-second-quarter-results.html ("*PALO ALTO, Calif.*, May. 28, 2025 – (GlobeNewswire) – HP (NYSE: HPQ)" (emphasis added)) (last visited September 24, 2025).

[7] Complaint for Patent Infringement, ¶ 9, *K.Mizra LLC v. SonicWall Inc.*, No. 1:25-cv-00047 (D. Del. Jan. 10, 2025), *available at* https://kmizra.com/k-mizra-v-sonicwall ("SonicWall is a corporation organized and existing under the laws of the state of Delaware with a principal place of business at 1033 McCarthy Boulevard, Milpitas, California, 95035.") (last visited September 24, 2025).

defendants want certainty—***that if they settle with a company, they are getting a licence to the entire portfolio instead of just one set of patents. He wanted to incorporate that at K.Mizra and started purchasing a wide range of patents.***" (emphasis added)).

27. Upon information and belief, K.Mizra has directed numerous communications to California including but not limited, to the companies identified in paragraph 26 for purposes of licensing K.Mizra's Patent Portfolio, which includes the Patents-in-Suit.

28. K.Mizra's business involves suing companies located in this District, including Google, and leaving them either the choice to expend significant resources defending themselves in court or submit to a licensing agreement. If K.Mizra is successful in its threat, a California resident, such as Google, transforms from a potential licensee to K.Mizra's customer. K.Mizra seeks Google's business, in one fashion or another: carrot or stick. During subsequent communications, K.Mizra's CEO reiterated that K.Mizra had a draft amended complaint against Google ready for filing.

29. K.Mizra has also voluntarily commenced actions in California courts as recently as January of this year, availing itself of the benefits of the California federal courts and demonstrating that California is not an inconvenient venue in which K.Mizra would litigate. *See*, *e.g.*, *K.Mizra LLC v. Epson Am., Inc.*, No. 8:25-cv-00027 (C.D. Cal. Jan. 7, 2025), *available at* https://kmizra.com/k-mizra-v-epson.

30. Because of its continuous and systematic contacts within the State of California and this District, K.Mizra has purposefully directed its commercial, licensing, and enforcement activities surrounding its Patent Portfolio, including the Patents-in-Suit, at California, and has voluntarily availed itself of courts in this District thereby subjecting itself to personal jurisdiction in this District. K.Mizra does business in the State of California.

31. Venue in this District is proper under 28 U.S.C. §§ 1391(b) and (c) at least because K.Mizra is subject to personal jurisdiction in this District, a substantial part of the events or omissions giving rise to the claims occurred in this District, and K.Mizra purposefully directed activities to this District.

32. An actual and justiciable controversy exists under 28 U.S.C. §§ 2201–2202 between

Google and K.Mizra as to whether the claims in the Patents-in-Suit are infringed by Google's products.

INTRADISTRICT ASSIGNMENT

33. Pursuant to Civil L.R. 3-2(c) and 3-5(b), this is an Intellectual Property Rights Action subject to assignment on a district-wide basis.

COUNT I:

DECLARATORY JUDGMENT OF NON-INFRINGEMENT OF THE '717 Patent

34. Google repeats and realleges Paragraphs 1 through 33 of this Complaint as if fully set forth herein.

35. K.Mizra claims to own all right, title, and interest in the '717 Patent by assignment.

36. During communications with Google's counsel in the Texas action, K.Mizra asserted that Google infringes at least claim 11 of the '717 Patent and provided infringement allegations thereto.

37. Claim 11 of the '717 Patent recites:

> A station for use in a wireless communication network, the station comprising a transmitter arrangement, a receiver arrangement and a processor circuit coupled to the transmitter arrangement and the receiver arrangement, the processor circuit being configured to start up in a not-associated state and to cause the transmitter arrangement to transmit association request messages in the not-associated state, the processor circuit being configured to switch to an associated state upon reception at the receiver arrangement of an association grant in response to one of the association request messages, the processor circuit being configured to establish an associated route for communication with an association unit during subsequent communication of operating messages to an association unit, the associated route including a source of the association grant, and, if the source of the association grant is not the association unit, the operating route between the source of the association grant and the association unit previously established for the source of the association grant in response to association request messages and/or the association grants for intermediate stations, the processor circuit being configured to cause the transmitter arrangement to initiate and transmit association grants in response to association request messages, but only after switching to the associated state.

38. In its draft amended complaint, K.Mizra accused Google's Nest line of products including Nest Hub, Hub Max, Wifi Pro, and Google TV Streamer, among others, that support the Thread networking protocol ("Accused Thread Products") of infringing claim 11 of the '717 Patent.

Ex. 4 ¶¶ 81–94, 134–62. K.Mizra's draft amended complaint treats claim 11 as a representative claim. *See*, *e.g.*, *id.* ¶¶ 79, 93, 135, 138. Google's Accused Thread Products do not infringe any claim of the '717 Patent because they do not include or practice multiple limitations required by claim 11 and any other independent claim, including, but not limited to "the processor circuit being configured to establish an associated route for communication with an association unit during subsequent communication of operating messages to an association unit, the associated route including a source of the association grant, and, if the source of the association grant is not the association unit, the operating route between the source of the association grant and the association unit."

39. Google, including its Accused Thread Products, has not, and does not, infringe any claims of the '717 Patent either directly, contributorily, or by inducement, literally or under the doctrine of equivalents. Google does not induce infringement of any claims of the '717 Patent for at least the reasons stated above with respect to no underlying direct infringement of the claims of the '717 Patent, and because Google has not acted with specific intent necessary for induced infringement. Google does not contributorily infringe any claims of the '717 Patent for at least the reasons stated above with respect to no underlying direct infringement of the claims of the '717 Patent, and because Google has not acted with specific knowledge of infringement.

40. As set forth above, there exists an actual controversy between Google and K.Mizra with respect to whether Google allegedly infringes any claims of the '717 Patent of sufficient immediacy and reality to warrant the issuance of a declaratory judgment. A judicial declaration is necessary and appropriate so that the parties may ascertain their respective rights regarding the claims of the '717 Patent.

41. Google seeks a judicial determination that Google has not directly infringed, induced others to infringe, or contributed to the infringement of the claims of the '717 Patent.

COUNT II:

DECLARATORY JUDGMENT OF NON-INFRINGEMENT OF THE '120 Patent

42. Google repeats and realleges Paragraphs 1 through 41 of this Complaint as if fully set forth herein.

43. K.Mizra claims to own all right, title, and interest in the '120 Patent by assignment.

44. During communications with Google's counsel in the Texas action, K.Mizra asserted that Google infringes at least claim 1 of the '120 Patent and provided detailed infringement allegations thereto.

45. Claim 1 of the '120 Patent recites:

> A method of determining hyperparameters of a classifier in a machine learning system by iteratively producing an estimate of a target hyperparameter vector, each iteration comprising the steps of:
>
> drawing a random sample of hyperparameter vectors from a set of possible hyperparameter vectors,
>
> updating the estimate of the target hyperparameter vector by using the random sample, and
>
> selecting, from the random sample of hyperparameter vectors, a hyperparameter vector producing a best result in the present and any previous iterations, and wherein the step of updating the estimate of the target hyperparameter vector uses said hyperparameter vector producing the best result.

46. In its draft amended complaint, K.Mizra accuses Google's Vertex AI Studio including its Vizier offering ("Vertex AI") of infringing claim 1 of the '120 Patent. Ex. 4 ¶¶ 65–66, 115–31. K.Mizra draft amended complaint treats claim 1 as a representative claim. *See*, *e.g.*, *id.* ¶¶ 116, 117. Google Vertex AI does not infringe any claim of the '120 Patent because it does not include or practice multiple limitations required by claim 1 and any other independent claim, including, but not limited to: "updating the estimate of the target hyperparameter vector by using the random sample, and [s]electing, from the random sample of hyperparameter vectors, a hyperparameter vector producing a best result in the present and any previous iterations, and wherein the step of updating the estimate of the target hyperparameter vector uses said hyperparameter vector producing the best result."

47. Google, including its Vertex AI, has not, and does not, infringe any claims of

the '120 Patent either directly, contributorily, or by inducement, literally or under the doctrine of equivalents. Google does not induce infringement of any claims of the '120 Patent for at least the reasons stated above with respect to no underlying direct infringement of the claims of the '120 Patent, and because Google has not acted with specific intent necessary for induced infringement. Google does not contributorily infringe any claims of the '120 Patent for at least the reasons stated above with respect to no underlying direct infringement of the claims of the '120 Patent, and because Google has not acted with specific knowledge of infringement.

48. As set forth above, there exists an actual controversy between Google and K.Mizra with respect to whether Google allegedly infringes any claims of the '120 Patent of sufficient immediacy and reality to warrant the issuance of a declaratory judgment. A judicial declaration is necessary and appropriate so that the parties may ascertain their respective rights regarding the claims of the '120 Patent.

49. Google seeks a judicial determination that Google has not directly infringed, induced others to infringe, or contributed to the infringement of the claims of the '120 Patent.

PRAYER FOR RELIEF

WHEREFORE, Google respectfully requests that judgment be entered:

A. Declaring that Google does not infringe, either directly, contributorily, or by inducement, any claim of the Patents-in-Suit;

B. Declaring that judgment be entered in favor of Google and against K.Mizra on each of Google's claims;

C. Finding this to be an exceptional case under 35 U.S.C. § 285;

D. Awarding Google its attorneys' fees and costs incurred in connection with this action; and

E. Awarding Google any other remedy or relief that the Court deems just and proper.

JURY DEMAND

Google hereby demands a trial by jury on all claims and issues so triable.

Dated: September 24, 2025

JONES DAY

By: */s/ Tharan Gregory Lanier*
Tharan Gregory Lanier (Lead Counsel)
Evan M. McLean
Michael A. Lavine
Sachin M. Patel
Peter Young Kim

Attorneys for Plaintiff Google LLC

EXHIBIT 1

US008144717B2

(12) **United States Patent**
Blange et al.

(10) **Patent No.: US 8,144,717 B2**
(45) **Date of Patent: Mar. 27, 2012**

(54) **INITIALIZATION OF A WIRELESS COMMUNICATION NETWORK**

(75) Inventors: **Marinus Johannes Blange**, Nootdorp (NL); **Miodrag Djurica**, Rotterdam (NL)

(73) Assignee: **Nederlandse Organisatie Voor Toegepast-natuurwetenschappelijk Onderzoek TNO**, Delft (NL)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 784 days.

(21) Appl. No.: **12/159,753**

(22) PCT Filed: **Jan. 2, 2007**

(86) PCT No.: **PCT/NL2007/000001**

§ 371 (c)(1),
(2), (4) Date: **Dec. 2, 2008**

(87) PCT Pub. No.: **WO2007/078193**

PCT Pub. Date: **Jul. 12, 2007**

(65) **Prior Publication Data**

US 2009/0215482 A1 Aug. 27, 2009

(30) **Foreign Application Priority Data**

Dec. 30, 2005 (EP) 05078044

(51) **Int. Cl.**
H04L 12/28 (2006.01)

(52) **U.S. Cl.** **370/402**; 370/395.2; 370/400; 370/401; 455/41.2; 455/445

(58) **Field of Classification Search** 455/41.2, 455/435.1, 445; 370/338, 328, 329, 341, 370/395.2, 395.21, 400–402, 406
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

7,548,523 B2 * 6/2009 Babbar et al. 370/331

(Continued)

FOREIGN PATENT DOCUMENTS

EP 1 471 693 A2 10/2004

OTHER PUBLICATIONS

Forno et al., "Design and implementation of a Bluetooth ad hoc network for indoor positioning" IEE Proceedings: Software, IEE, vol. 152, No. 5, Oct. 7, 2005 pp. 223-228.

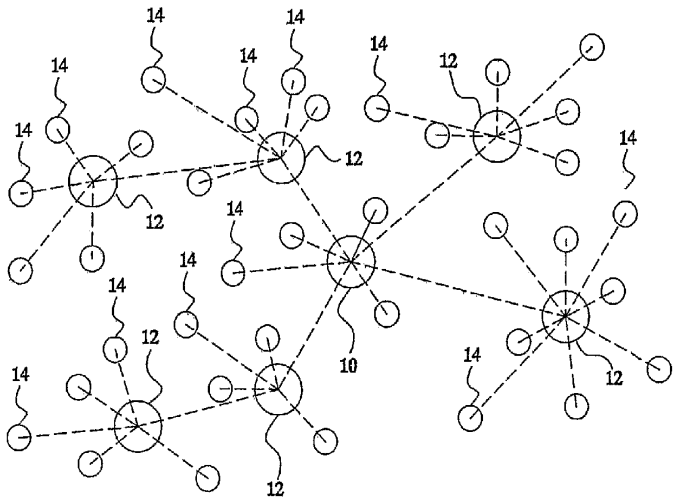
(Continued)

Primary Examiner — Dominic E Rego
(74) *Attorney, Agent, or Firm* — Leydig, Voit & Mayer Ltd.

(57) **ABSTRACT**

A wireless communication network comprises an association unit and a plurality of stations. At least part of the stations store association information indicative of an associated route for transmission of messages from the station to the association unit. The association information is generated by installing the stations in a not-associated state; transmitting association request messages from stations in the not-associated state; and transmitting association grant messages in response to the association request messages. Each association grant message is transmitted only from the association unit or, at least when the station that has transmitted association request message is outside a reception range of the association unit, from one of the stations that is in an associated state. Each station that has received the association grant message in response to its association request message is switched to the associated state. Association information is stored that indicates the station that has sent the association grant message as part of the associated route to the association unit.

20 Claims, 3 Drawing Sheets



U.S. PATENT DOCUMENTS

7,823,199	B1 *	10/2010	Rathi et al.	726/22
8,045,531	B2 *	10/2011	Cheng et al.	370/338
2001/0014868	A1 *	8/2001	Herz et al.	705/14
2003/0151513	A1	8/2003	Herrmann et al.	
2003/0174682	A1 *	9/2003	Barker	370/338
2004/0047307	A1 *	3/2004	Yoon et al.	370/324
2004/0223476	A1 *	11/2004	Jose et al.	370/338
2005/0088980	A1 *	4/2005	Olkkonen et al.	370/255
2005/0122927	A1	6/2005	Wentink	
2005/0195842	A1 *	9/2005	Dowling	370/401
2005/0221803	A1 *	10/2005	Hayashi	455/413
2006/0133266	A1 *	6/2006	Kim et al.	370/228
2006/0133328	A1 *	6/2006	Levendel et al.	370/338
2006/0166683	A1 *	7/2006	Sharma et al.	455/456.5
2006/0268756	A1 *	11/2006	Wang et al.	370/310
2007/0038346	A1 *	2/2007	Ehrlich et al.	701/33

OTHER PUBLICATIONS

Rubin et al., "TBONE: A Mobile-Backbone Protocol for Ad Hoc Wireless Networks" Aerospace Conference Proceedings, IEEE Mar 9-16, 2002. vol. 6, pp. 2727-2740.

International Search Report for PCT/NL/2007/0000001, dated Jun. 6, 2007.

* cited by examiner

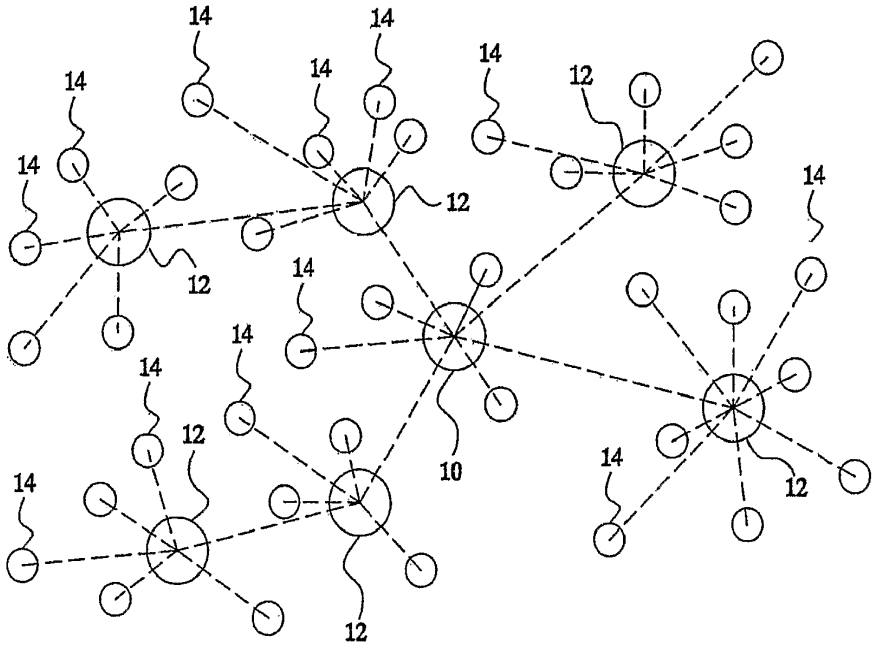


Fig.1

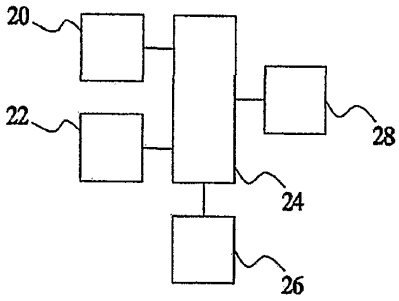


Fig.2

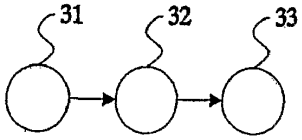


Fig.3

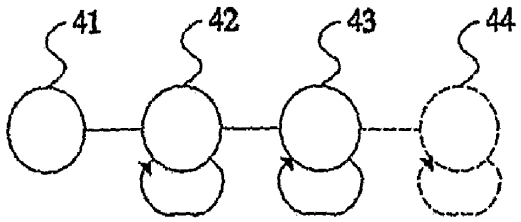


Fig4

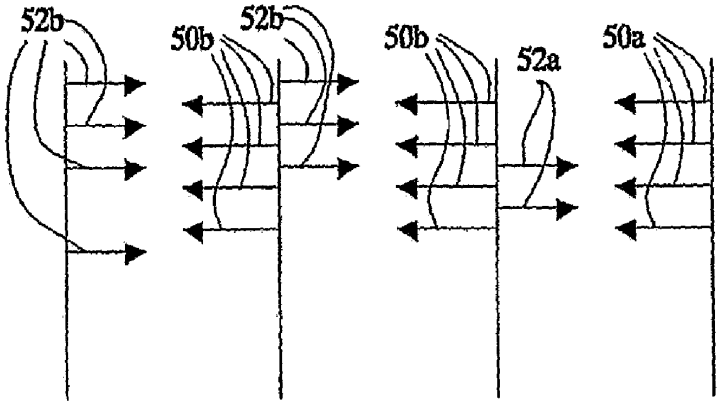


Fig.5a

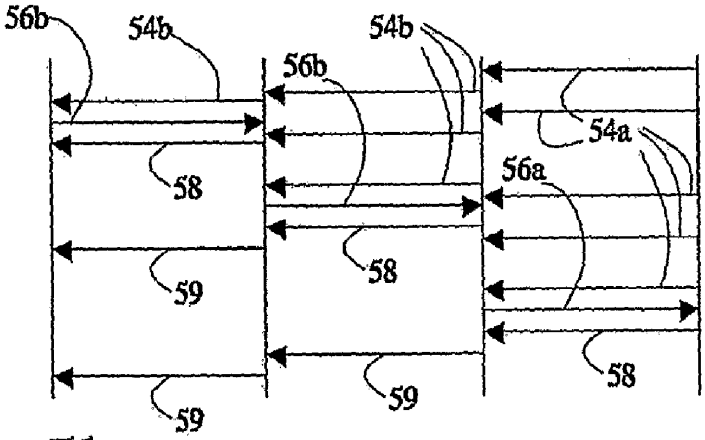


Fig.5b

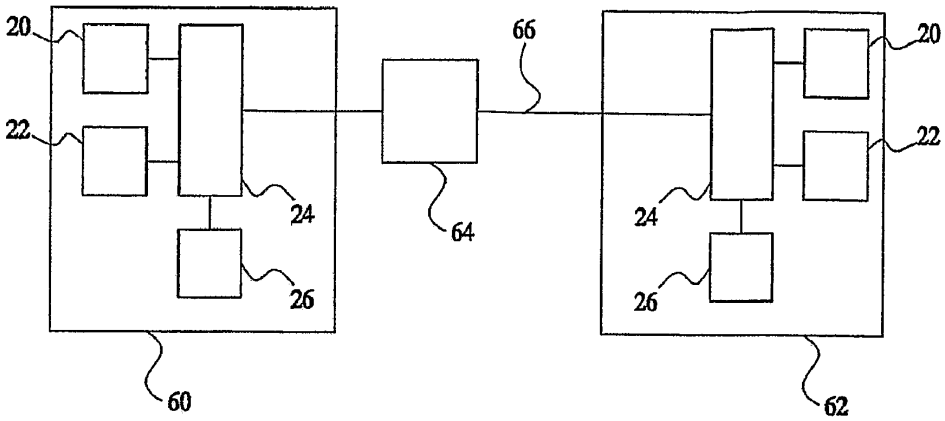


Fig.6

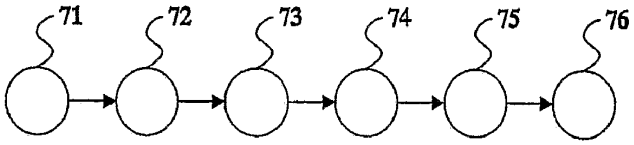


Fig.7

INITIALIZATION OF A WIRELESS COMMUNICATION NETWORK

FIELD OF THE INVENTION

The invention relates to a method of operating a wireless communication network, to a wireless communication system that uses such a method and to stations for use in such a network.

BACKGROUND

US patent application No. 2003/0151513 describes a wireless sensor station network. In operation data from the sensor stations is transmitted in multiple hops to a central panel and vice versa. This is because the sensor stations are not able to transmit directly to the central panel. Each sensor station is given its own, fixed route for transmitting and receiving data to and from the central panel. Typically, the data is first sent to another sensor station and from there to another sensor station and so on until a cluster head is reached. Then the data is sent through a network of cluster heads to the central panel.

After installation of the sensor stations at various locations in a sensing area a network topology is dynamically formed. That is, the routes for each of the sensor stations are automatically selected and programmed into the sensor stations. In US patent application No. 2003/0151513 describes a number of methods of dynamically forming the routes from the sensor stations to cluster head units that channel communication from groups of sensor stations. The initiative is taken by the cluster head unit, which transmits link discovery messages, which alert nearby sensor stations about the availability of the cluster head. These sensor stations forward the link discovery messages. Further sensor stations may receive and forward the forwarded link discovery messages and so on. The link discovery packets function to synchronize communication clocks of the cluster head and the sensor stations.

Next the routes for transmitting messages from the sensor stations to the association unit are defined. In one embodiment this is done by transmitting and forwarding route discovery packets from the association unit. The forwarded route discovery packets indicate the number of hops over which they have been forwarded. In this embodiment each sensor stations picks a route by selecting the association unit or sensor station from which it received a route discovery packet over least hops. The sensor stations record an indication of these routes for later use in normal operation. Subsequently, the sensor stations send route registration packets to the association unit to indicate the route that they have selected.

In another embodiment the sensor stations send link registration messages without waiting for route discovery messages. These link registration messages may reach the association unit via each available route. In this embodiment the association unit selects routes for the sensor stations from which it has received link registration messages. After making the selection the association unit sends route definition messages to the sensor stations for recording an indication of these routes for later use in normal operation.

A problem with these techniques is that they require transmission of information via sensor stations before the routing topology has been defined, i.e. when plural routes are still possible. This complicates the (sensor) stations, because they have to support different modes of communication while operating in low-power mode. If forwarding of information is only done using the definite routing topology, a central approach is slow (each station has to wait until its upward station has centrally been assigned a route) and a decentral approach is inefficient (a routing topology with an unnecessary amount of branches has been created)

Another problem that may occur is that the wireless links between nodes are hindered by barriers and that using the same frequency for upward and downward link routes requires synchronization. This complicates the (sensor) station, because they have to support synchronization and powerful RF signals.

A mobile communications protocol is known from an article titled "TBONE: A Mobile-Backbone Protocol for Ad Hoc Wireless Networks" by Izhak Rubin et al. and published in the IEEE Aerospace Conference proceedings 2002 (Piscataway N.J., Mar. 9-16, 2002) page 2727-2740. The mobile protocol of this article uses two types of network: a high power network termed the BNET, or backbone, and a plurality of low power networks called ANETs. In each low power network (ANET) low power nodes communicate under control of a central node of that ANET. Each of the central nodes also functions as backbone node, i.e. as a node in the high power backbone network.

Devices are provided that are capable of operating as backbone nodes. Initially, each of these devices decides whether it will switch to operation as backbone node, dependent on whether it has a higher qualifications to do so than neighboring devices that are capable of operating as backbone nodes. Subsequently, those devices that have switched to operation as backbone nodes receive join requests from remaining nodes that want to become part of a low power network (ANET) centred on a particular backbone node. The low power networks (ANETs) are formed by granting these join requests. Join requests can come from both remaining backbone capable devices that have not decided to operate as backbone devices, and from low power nodes that are not capable of operating as backbone devices at all.

The article also describes the possibility that remaining backbone capable devices (that have not initially decided to operate as backbone devices) can be commanded to become backbone nodes by existing backbone nodes. This happens when a remaining backbone capable device that is part of a low power network (ANET) receives and relays a join request from a requesting node that cannot be accepted by any existing backbone node. When this happens the central backbone node of the low power network (ANET) commands the remaining backbone capable device to switch to operation as a backbone node and to grant the join request of the requesting node. In this case the remaining backbone capable device joins the high power network (BNET), sets up a low power network (ANET) of its own and ceases to use the paths in the low power network (ANET) of the commanding backbone node. The remaining backbone capable device uses the high power network to relay further messages to or from the requesting node. No provision is made for routing through successive previously established low power network (ANET) nodes and association for this form of routing.

This system addresses the use of two types of network links: links in the low power networks and links in the high power networks. Direct communication links (not via other associated nodes) are used in the low power networks. The selection of communication paths in the high power network is not described, but as high power communication is used longer range paths can be used in this network than in the low power networks. As the system is intended to provide communication in a dynamic environment rather than control of devices in the system there is no single central backbone

node. The use of two types of network without central unit makes this system complex and inefficient in bandwidth use.

SUMMARY OF THE INVENTION

Among others, it is desirable to provide for a wireless network wherein routes for communication can be determined without the need to forward information via stations for which the routes have not yet been defined.

Among others, it is desirable to provide for a method of operating a wireless network wherein routes for communication can be determined without the need to forward information via stations for which routes have not yet been defined.

Among others it is desirable to provide for stations for flexible use in wireless networks.

The invention provides for a wireless communication network according to claim **1**. This network allows routes between stations and an association station to be set up for further use, using routes that have already been defined in response to earlier associations. Each route is established when an association request from a station is granted. Establishment of a route is effected by the association unit for example by storing information that defines the route to the associated station in a memory that the association unit uses to select the route each time when a message has to be sent to the associated station. Routes are established upon an association grant from the association unit or from a station that has switched to the associated state directly or indirectly based on an association grant from the association unit, in a wave of association that radiates from nodes to nodes starting from the association unit. Only the association unit and stations that have an established route to the association unit, which has been established in this way in response to association requests and corresponding association grants by nodes along the route, are allowed to grant association requests. Therefore their routes can immediately be used for the station that is associated.

In an embodiment a station is configured to transmit association requests first for association via an active frequency channel and subsequently, if the first transmitted association request is not granted, for association via an as yet unused frequency channel. The latter request may be transmitted for example directly to the association unit. Because the route that is to be used to the station is established only upon association it is prevented that inconsistent multiple routes are assigned when such different association requests are transmitted.

More generally, in an embodiment the station that issues the association grant to a specific station is configured to make issue of the association grant dependent on whether the specific station indicates that a function that is active in that specific station or not. This means that a specific station may associate with different other stations, dependent on whether the function is active in the specific station or not. As a result the specific station has the opportunity to issue a plurality of association requests: without activating the function and after the function is activated, for example with a request directly to the association unit. Thus, for example, if the function consumes a scarce resource (such as function that uses a frequency channel for communication between the stations, thereby excluding use of the frequency channel by other stations), it can be prevented that the resource is consumed needlessly by granting the association request before activating the function. Because the routes to stations are established only upon association it is prevented that inconsistent station addresses are assigned.

In an embodiment association is preceded by a discovery stage, wherein no routes are yet established but station discover other stations, that can function to relay messages. Thus, stations may be enabled first to discover addresses of available other stations to which association requests can be sent, or stations that may send association requests before a route is established. In a further embodiment discovery advertisement messages are used that identify stations that are enabled to grant discovery requests. Thus, it is made possible to avoid waste of energy or bandwidth on association requests that cannot be granted.

In another embodiment part of the stations are dual function stations, which start to send association grant messages only after receiving approval from the association unit to do so. Thus, the association unit has control over the selection of stations that act as relay station, even after the dual function station has obtained a route for its "own" messages (e.g. sensor messages from a built in sensor station). In an embodiment the dual function station comprises units that function as a beacon frame transmitter and as a station with sensor station-like communication abilities respectively. The association unit may follow a strategy of enabling such dual function stations selectively to avoid enabling many stations that act as relay for only a few stations each, or to avoid selecting stations when these will become a relay for messages for too many stations.

In another embodiment dual function stations are used that comprises a mechanically flexible connection between separate units that each have an antenna or antennas for transmission and reception. Thus such stations can be used to connect different spaces "around the corner".

In another embodiment dual function stations are used that operate concurrently in different frequency channels that are periodically active in time slots that repeat with different asynchronous periods. This simplifies coordination between different network parts.

BRIEF DESCRIPTION OF THE DRAWINGS

These and other objects and advantageous aspects will be described for exemplary examples, using the following figures.

FIG. **1** shows a wireless network
FIG. **2** shows a station
FIG. **3** shows a state diagram of a sensor station
FIG. **4** shows a state diagram of a beacon frame transmitter
FIG. **5***a* shows messages during a discovery phase
FIG. **5***b* shows messages during an association phase
FIG. **6** shows an embodiment of a beacon frame transmitter
FIG. **7** shows a state diagram of a beacon frame transmitter

DETAILED DESCRIPTION OF THE DRAWINGS

FIG. **1** shows a wireless network. The network comprises an association unit **10**, a plurality of beacon frame transmitters **12** and a plurality of sensor stations **14** (only some numbered explicitly). Fixed communication routes are defined between association unit **10** and sensor stations **14**, using wireless communication links (shown by dashed lines) between association unit **10**, beacon frame transmitters **12** and sensor stations **14**. The communication routes form a tree structure with association unit **10** at its root and beacon frame transmitters **12** at its nodes. Each beacon frame transmitter **12** is coupled in a star topology to a number of the sensor stations **14**.

FIG. **2** shows a station. At the level of detail shown, the figure applies to beacon frame transmitters **12** as well as

sensor stations **14**. The station contains a receiver circuit **20**, a transmitter circuit **22**, a processor circuit **24**, a memory **26** and, at least in case of sensor stations, a sensing circuit **28**. Processor circuit **24** is coupled to receiver circuit **20**, transmitter circuit **22**, memory **26** and optionally to sensing circuit **28**. Each of receiver circuit **20** and transmitter circuit **22** include an antenna (not shown). Preferably, the antennas of receiver circuit **20** and transmitter circuit **22** are formed by a shared antenna of receiver circuit **20** and transmitter circuit **22**.

Preferably the stations are battery operated, with on board batteries (not shown) or supplied by other on board power supplies such as solar cells. This means that it is desirable to reduce power consumption as much as possible.

In operation, data is exchanged between sensor stations **14** and association unit **10** via the links that are shown in FIG. **1**. Because the links form a tree structure, a single route to and from association unit **10** is defined for each sensor station **12**.

It should be appreciated that the communication routes are a result of selection from a greater number of possible routes. In principle, each sensor station **14** could communicate directly with association unit **10** and each beacon frame transmitter **12**. In practice, a number of these direct communication routes are not possible, because of insufficient signal strength. But direct wireless communication from a sensor station **14** to still more than one beacon frame transmitters **12** and/or to association unit **10** may be possible. Similarly, each beacon frame transmitter **12** may be able to communicate directly with association unit **10** and/or a number of beacon frame transmitters **12**.

The wireless connections shown in FIG. **1** are the result of a selection from these possible routes. A number of operable wireless communication links between sensor stations **14**, beacon frame transmitters **12** and association unit **10** is not used. These links are not shown in FIG. **1**. Sensor stations **14** transmit messages to association unit **10** via fixed routes and conversely association unit **10** transmits messages to sensor stations via the fixed routes. The routes are defined by information in memory **26**.

Each sensor station **14** stores at least information that is selective for the beacon frame transmitter **12** at the start of its route to association unit **10**; this address information is used to form messages that the sensor station **14** sends to association unit **10**. Typically, each beacon frame transmitter **12** stores at least information that is selective for the next beacon frame transmitter **12** in the route or for association unit **10**. This information is used to forward messages from the sensor station **14** sends to association unit **10**. Alternatively, each sensor station **14** may store information that is selective for the entire route to association unit **10** and the sensor station **14** may use this information to form messages, from which beacon frame transmitters **12** can determine how to forward the message.

Conversely, association unit **10** preferably stores information for each sensor station **14** that is selective for the beacon frame transmitter **12** at the start of the route to that sensor station **14**. Association unit **10** uses this information to form messages that association unit **10** transmits to the sensor station **14**. Typically, association unit **10** stores information that identifies the entire route to each sensor station **14**. The association unit **10** may use this information to form messages, from which beacon frame transmitters **12** can determine how to forward the message to the sensor station **14**. Alternatively, each beacon frame transmitter **12** stores, for each sensor station **14** for which it can receive messages, at least information that is selective for the next beacon frame transmitter **12** in the route to that sensor station **14**. This information is used to forward messages for the sensor station **14** from association unit **10** to the sensor station **14**.

An embodiment of the part of operation wherein the routes are established will now be described. Basically this embodiment includes two stages:

- a discovery stage wherein beacon frame transmitters **12** and sensor stations **14** become "aware" of stations (association unit **10** or beacon frame transmitters **12** in the case of sensor stations and association unit **10**, other beacon frame transmitters **12** and sensor stations **14** in the case of beacon frame transmitters **14**) that are within reception range and
- an association stage wherein sensor stations **14** and beacon frame transmitters **14** (or sensor stations that are fixedly coupled to these beacon frame transmitters) send requests to associate to association unit **10** or beacon frame transmitters **12**, i.e. to establish a route through the association unit **10** or beacon frame transmitter **12** for subsequent use.

The association stage for a sensor station **14** (or beacon frame transmitter **12**) is completed when an association request is granted by the association unit **10** or a beacon frame transmitter **12**. The route via the granting station is thereby established for subsequent use. In one embodiment each beacon frame transmitter **12** is enabled to grant association requests immediately after it has received a grant of an association request itself, but preferably the beacon frame transmitter **12** is enabled to grant association requests only after receiving secondary grant message, this time from the association unit **14** that allows the beacon frame transmitter **12** to grant association requests. Association unit **10** may for example make issuance of the secondary grant message dependent on information about the number of as yet unassociated stations that the beacon frame transmitter **12** has discovered. Thus, association unit **10** can control network topology.

Preferably, the association requests are addressed specifically to beacon frame transmitters **12** or the association unit **10**, using addresses (or more generally any identifier of these stations) that have been discovered in the discovery stage. Thus, it is excluded that conflicting association grants can be received in response to an association requests. However, addressing may be made unnecessary, for example when stations address all subsequent messages to one station that has sent a grant, or if some conflict resolution scheme is used.

In the embodiment the discovery stage comprises use of discovery messages from sensor stations **14** and beacon frame transmitters **12** (or sensor stations that are fixedly coupled to these beacon frame transmitters **12**) and responses in reaction to these discovery messages. Alternatively or in addition, stations may be discovered from unsolicited messages. The discovery stage and the association stage may overlap, the stations continuing to collect identifiers of discovered stations between transmission of association requests during the association stage. Alternatively non-overlapping consecutive stages may be used. In another embodiment the discovery stage may even be eliminated, for example if association requests are used that are not specifically addressed, or if association requests are used to discover stations.

When the discovery stage and the association stage may overlap, it is preferred that beacon frame transmitters **12** include information in their messages to indicate whether they have been enabled to grant association requests (preferably in general, i.e. not limited to requests from specific sensor stations **14**, but in a further embodiment specific sensor stations **14** or beacon frame transmitters **12** from which association requests may be accepted may be indicated).

Thus, sensor stations **14** and/or beacon frame transmitters **12** can avoid transmission of association requests to discovered beacon frame transmitters **12** that are not enabled to grant association requests.

FIG. **3** shows a state diagram of operation of a sensor station **14** before a route is established. FIGS. **5***a,b* shows messages exchanged with sensor station **14** during discovery and association respectively. Initially the sensor station **14** is in a first state **31**. In this first state **31** processor circuit **24** causes transmitter **22** to transmit discovery messages **50***a*, for example once every minute, every two minutes or every hour. When processor circuit **24** detects that receiver **20** has received a beacon frame **52***a* from a beacon frame transmitter **12** or association unit **10**, processor circuit **24** switches sensor station **14** to a second state **32**. In this way, the beacon frame **52***a* acts as a discovery confirm message.

Preferably, sensor station **14** switches to second state **32** in response indiscriminately to both beacon frames **52***a* that are transmitted in response to the discovery message or to any other beacon frame that is received spontaneously. In an embodiment sensor station **14** switches to second state **32** in response only to beacon frames **52***a* wherein it is indicated that the transmitting station is prepared to grant association requests and not to optional other beacon frames **52***a* that do not contain such an indication.

In second state **32** processor circuit **24** causes transmitter **22** to transmit association request messages **54***a* for example once every minute, every two minutes or every hour. Preferably each association request message **54***a* includes an identification of the sensor station **14** that allows messages specifically directed at the sensor station **14** to be sent in response. Also preferably, processor **24** reads an identification of the beacon frame transmitter **12** or association unit **10** from a previously received discovery confirm message and includes that identification in the association request message. This may be done for example when it is desirable to avoid collisions between association grant messages in response to the association request message.

When processor circuit **24** detects that receiver **20** has received an association grant message **56***a* from a beacon frame transmitter **12** or association unit **10** directed at the sensor station **14** of the processor circuit **24** in response to an association request message **54***a*, processor circuit **24** switches sensor station **14** to a third state **33** and records information for reaching this beacon frame transmitter **12** or the association unit **10** in memory **26** to define route to association unit **10** for use during normal operation. Optionally sensor station sends an acknowledgment **58** of the association grant message back to the beacon frame transmitter **12** or association unit **10**.

In third state **33** sensor station **14** stops transmitting association request messages **54***a*. In one embodiment sensor station **14** stops transmitting discovery messages **50***a* already when it switches to second state **32**. Alternatively sensor station **14** may be arranged to continue transmitting discovery messages **50***a* in second state **32**, only to stop after switching to third state **33** or after a predetermined time interval has elapsed after switching to second state **32**, if this occurs earlier than the switch to third state **33**.

In an embodiment a station is configured to transmit association requests first for association via an active frequency channel and subsequently, if the first transmitted association request is not granted, for association via an as yet unused frequency channel. The latter request may be transmitted for example directly to the association unit. Because the route that is to be used to the station is established only upon association it is prevented that inconsistent multiple routes are assigned when such different association requests are transmitted.

More generally, in an embodiment the station that issues the association grant to a specific station is configured to make issue of the association grant dependent on whether the specific station indicates that a function that is active in that specific station or not. This means that a specific station may associate with different other stations, dependent on whether the function is active in the specific station or not. As a result the specific station has the opportunity to issue a plurality of association requests: without activating the function and after the function is activated, for example with a request directly to the association unit. Thus, for example, if the function consumes a scarce resource (such as function that uses a frequency channel for communication between the stations, thereby excluding use of the frequency channel by other stations), it can be prevented that the resource is consumed needlessly by granting the association request before activating the function. Because the routes to stations are established only upon association it is prevented that inconsistent station addresses are assigned.

FIG. **4** shows a state diagram of operation of a beacon frame transmitter **12** before a route is established. Initially the beacon frame transmitter **12** is in a first state **41**. In this first state **41** processor circuit **24** causes transmitter **22** to transmit discovery messages **50***b*, for example once every minute, every two minutes or every hour. When processor circuit **24** detects that receiver **20** has received a beacon frame **52***b* from another beacon frame transmitter **12** or association unit **10**, processor circuit **24** switches beacon frame transmitter **12** to a second state **42**. In this way, the beacon frame **52***b* acts as a discovery confirm message. In an embodiment discovery confirm messages may also be transmitted unsolicited, i.e. not in response to a discovery message, in this sense discovery messages could also be called "hello" messages. The term "discovery advertisement message" is used herein both for messages in response to discovery requests and unsolicited (hello) messages. Examples of use of discovery advertisement messages will be given in terms of discovery confirm messages, but unless indicated otherwise these examples apply generally for discovery advertisement messages. In a further embodiment stations need not transmit discovery messages **50***a,b* at all. In this case the stations listen only to unsolicited messages. The discovery confirm message does not need to be addressed to any station in particular: any station may react to any discovery confirm message.

Preferably, beacon frame transmitter **12** switches to second state **42** in response indiscriminately to both a beacon frame **52***b* that is transmitted in response to the discovery message or to any other beacon frame that is received spontaneously. In an embodiment sensor station **14** switches to second state **42** in response only to beacon frames **52***b* wherein it is indicated that the transmitting station is prepared to grant association requests and not to optional other beacon frames **52***b* that do not contain such an indication.

In second state **42** processor circuit **24** causes transmitter **22** to transmit association request messages **54***b*, for example once every minute, every two minutes or every hour. Preferably each association request message **54***b* includes an identification of the beacon frame transmitter **12** that allows messages specifically directed at the beacon frame transmitter to be sent in response. Also preferably, processor **24** reads an identification of the beacon frame transmitter **12** or association unit **10** from a previously received discovery confirm message and includes that identification in the association request message. This may be done for example when it is

desirable to avoid collisions between association grant messages in response to the association request message.

When processor circuit **24** detects that receiver **20** has received an association grant message **56***b* from another beacon frame transmitter **12** or association unit **10** in response to an association request message **54***b*, processor circuit **24** switches beacon frame transmitter **12** to a third state **43** and records information for reaching this other beacon frame transmitter **12** or the association unit **10** in memory **26** to define a route to association unit **10** for use during normal operation. Optionally frame transmitter **12** sends an acknowledgement **59** of the association grant message back to the beacon frame transmitter **12** or association unit **10**.

In third state **43** frame transmitter **12** stops transmitting association request messages **54***b*. In one embodiment frame transmitter **12** stops transmitting discovery messages **50***b* when it switches to second state **42**. Alternatively frame transmitter **12** may stop after switching to third state **43**, or after a predetermined time interval has elapsed after switching to second state **42**, if this occurs earlier than the switch to third state **42**.

Preferably beacon frame transmitter **12** transmits discovery confirm messages in at least when in second and third state **42**, **43**. More preferably beacon frame transmitter **12** includes an indication in the discovery confirm messages indicating whether the beacon frame transmitter **12** is ready to respond to association grant messages with association requests. This allows stations to avoid using bandwidth for transmission of useless association requests. In principle beacon frame transmitter **12** may also be arranged to transmit discovery confirm messages in the first state **41**. Preferably, when beacon frame transmitter **12** receives a discovery request message from a sensor station **14** or another beacon frame transmitter **12** beacon frame transmitter **12** records an identification of the sensor station **14** or other beacon frame transmitter **12** from which the discovery request was received. Optionally beacon frame transmitter **12** also records identifications of stations that have transmitted association requests that the beacon frame transmitter **12** did not respond to, if any.

In first state **41** and second state **42** beacon frame transmitter **12** does not respond to association request messages **54***a*. However, in third state **43** beacon frame transmitter **12** does respond to association request messages **54***a*. In this state processor circuit **24** causes transmitter **22** to transmit an association grant message **56***a* in response to each received association request message **54***a*, directed at the station that has sent association request message **54***a*, using the station identification in the association request message **54***a*. Processor circuit **24** includes information for reaching the beacon frame transmitter **12** in the association grant message **56***a* and optionally also further information for reaching association unit **10**.

Furthermore, beacon frame transmitter **12** transmits an update message **59** with information that the sensor station **14** has been associated to association unit **10** after sending the association grant message **56***a* (and after receiving an optional acknowledgment **58** of that message).

This response of beacon frame transmitter **12** to association request messages **54***a* is not limited to such messages from sensor stations **14**, but applies also to association request messages **54***b* from other beacon frame transmitters **12**.

It should be noted that beacon frame transmitter **12** and sensor station **14** become stations of the network with well defined routes only when reaching their respective third states **33**, **43**. Furthermore it should be noted that beacon frame transmitters **12** respond to association request messages **54***a,b* only once they have entered their third state **43**, that is, only when the beacon frame transmitter **12** has a well defined route to association unit **10**. Thus, the information about association can be transmitted along the route that will be used during normal operation. This route may also be used as part of the route to the newly associated sensor station **14**. Association unit **10** may use information from its memory, about the route to the beacon frame transmitter **12** to define the route to the sensor station **14**. In another embodiment, the beacon frame transmitters **12** along the route may add their identity to inform association unit **10** of the route, or in yet another embodiment the beacon frame transmitter **12** that associates to the sensor station **14** may insert this information.

It may be noted that not all beacon frame transmitters **12** need to become part of the network as beacon frame transmitters. If a beacon frame transmitter **12** receives no association grant signal **56***b* or grants no association grant requests from other beacon frame transmitters **12** or sensor stations **14** the beacon frame transmitter **12** does not become a functional beacon frame transmitter in the network.

In an embodiment association unit **10** decides on whether a beacon frame transmitter **12** becomes a functional beacon frame transmitter. In this embodiment functional beacon frame transmitters **12** in third state **43** forward association requests **54***a* from other beacon frame transmitters **12** (and optionally also from sensor stations **14**) to association unit **10** before sending an association grant message **56***b*. (Alternatively a further message derived from this association request is forwarded).

In this embodiment association unit **10** is arranged to grant or deny the forwarded association request in response to forwarding (or to the further message). When the association request is granted association unit sends back a grant message, which is used to generate the association grant signal **56***b* to the original association request **54***b*. Thus association unit **10** (and/or a beacon frame transmitter **12**) is enabled to control generation of the network topology, for example by first granting association to selected preferred beacon frame transmitters and granting association requests to others later only if this is still necessary to complete connections to all sensor stations **14**.

In a further embodiment beacon frame transmitters **12** are arranged to supply information about the number and/or identity of the sensor stations **14** and or beacon frame transmitters **12** from which it has received beacon frame request messages and/or association request messages. The identity may be provided by including identity information in beacon frame request messages and/or association request messages. The beacon frame transmitters **12** may supply this information for example in their own beacon frame request messages and/or association request messages. In this further embodiment the association unit **10** (and/or a beacon frame transmitter) is provided with information which it may use to select beacon frame transmitters **12** for granting association. This may be used for example to distribute links over several beacon frame transmitters **12**, or on the contrary to avoid distribution by concentrating links on one beacon frame transmitter **12**.

In a further embodiment beacon frame transmitters **12** return grants of in response to association requests only after receiving a grant of a secondary association request, distinguished, at least in function, if not in format, from the previously described association request. For this purpose optionally an additional fourth state **44** may be added to the states of FIG. **4**. When in third state **43** (after receiving the previously described grant to one of the previously described association requests, which will be termed primary association requests beacon frame transmitter **12** transmits a secondary associa-

tion request to association unit **10**. Upon receiving a grant of this secondary association request beacon frame transmitter **12** switches to the fourth state **44**. In this embodiment the beacon frame transmitter **12** is enabled to begin to return association grant messages to requesting stations only in fourth state **44**. Optionally the beacon frame transmitter indicates this in discovery confirm messages.

Preferably beacon frame transmitter **12** is configured to include a list of stations from which beacon frame transmitter **12** has received discovery request messages (and/or not-granted association request) in the secondary association request. As an alternative beacon frame transmitter **12** may optionally be configured to transmit one or more accompanying messages to provide such a list to association unit **10**. This enables association unit **10** to select to which of the beacon frame transmitters **12** it will grant association requests of the second type. In this way association unit **10** can control the network topology, after admitting beacon frame transmitters **12** to the network by granting a primary association request.

In a preferred embodiment each functional beacon frame transmitter **12** (and association unit **10**, which acts as a beacon frame transmitter in this respect) operates in a time slot scheme during normal operation, periodically transmitting a beacon signal to which the sensor stations **14** that are associated with that beacon frame transmitter may synchronize. After the beacon signal the beacon frame transmitter **12** and its associated stations exchange messages for a limited period.

This allows the beacon frame transmitters **12** and the sensor stations **14** to switch between a low power sleep mode and a normal operating mode. After exchanging messages the beacon frame transmitters **12** and the sensor stations **14** switch to the sleep mode wherein power supply and/or clock signals to at least part of the beacon frame transmitter **12** and the sensor station **14** is switched off. After counting off a predetermined time-interval the beacon frame transmitters **12** and the sensor stations **14** switch back to the normal operating mode. With a predetermined delay after switching to the normal operating mode the beacon frame transmitter **12** transmits the beacon frame and the sensor stations **14** receive this beacon frame.

In this embodiment the association unit **10** and/or the beacon frame transmitters preferably select the timing of the first of the periodic beacon frames so that the time slots of different beacon frame transmitters **12** that are within range to receive each others messages do not overlap. In this embodiment the information in the sensor stations **14** after association defines the time slot in which the sensor station **14** should switch to the normal operating mode.

In one embodiment each beacon frame transmitter **12** also switches to a forwarding mode in the time slot defined by the further beacon frame transmitter **12** through which the original beacon frame transmitter **12** transmits messages to the association unit **10** (or in the time slot of the association unit **10**). When in the forwarding mode the original beacon frame transmitter **12** exchanges messages with the further beacon frame transmitter **12**.

This embodiment has the disadvantage that overlap of the time slots of neighboring beacon frame transmitters **12** must be avoided. This means that beacon frame transmitters **12** must monitor time slots of other beacon frame transmitter **12**. In turn this means that the beacon frame transmitters **12** cannot operate in the sleep mode as much as desirable.

In an alternative embodiment the beacon frame transmitter **12** are designed to transmit and receive in a selectable one of a plurality of frequency channels. In this embodiment the frequency channels for different functional beacon frame transmitters **12** are selected during initialization, for example by the beacon frame transmitters themselves after detecting that no messages are received in the frequency channel during a predetermined time period, or by the association unit **10** so that beacon frame transmitters **12** that have been detected to receive each other get different frequency channels to avoid collisions. In an embodiment the beacon frame transmitters **12** and their associated sensor stations **14** switch to the selected frequency channel when they start normal operation. In this embodiment the information in the sensor stations **14** after association defines the frequency channel for communicating with the beacon frame transmitter **12**. Use of different frequency channels has the advantage that no extensive synchronization between time slots of beacon frame transmitters **12** is needed. Therefore beacon frame transmitters **12** can switch to the sleep mode more often.

In this further embodiment one frequency channel may be used during initialization. Alternatively a plurality of the frequency channels may be used in parallel. As an alternative the beacon frame transmitters **12** themselves may select a frequency channel during initialization after detecting that no messages are received in the frequency channel during a predetermined time period. Subsequently each beacon frame transmitter **12** performs all of its operations towards sensor stations **14** using the selected frequency channel, and uses another frequency channel to make upstream connections toward association unit **10**.

FIG. **6** shows a further embodiment of a beacon frame transmitter. The beacon frame transmitter contains two unit **60**, **62** as shown in FIG. **2**, plus a message exchange memory **64** coupled between the processor circuits **24** of the two units. In one embodiment mutually different units **60**, **62** are used, a first unit **60** arranged to perform the functions of a beacon frame transmitter (having a higher power transmitter **22** and/or a differently programmed processor circuit for example than the other unit) and a second unit **62** of a type used in sensor stations **14**. A mechanically flexible connection **66** is provided between the units **60**, **62** (e.g. between message exchange memory **64**, which may be part of the first unit **60** and second unit **62**). Flexible connection **66** is for example at least 5 centimetre long.

In this embodiment a first unit **60** acts as beacon frame transmitter, except for the transmission of discovery requests **50***b*, association requests **54***b* and any communication towards association unit (e.g. update message **59**). The second unit **62** performs the remaining functions of the beacon frame transmitter. It may be noted that this second unit performs much like a sensor station **14**, except that it does not generate or receive its own messages. Instead it forwards messages to or from the other unit. Messages to and from the association unit **10** or other beacon frame transmitters **12** (or data from such messages) are placed in message exchange memory **64**.

In normal operation units **60**, **62** switch between the sleep mode and normal active mode in time slots defined by respective beacon frames. The unit that acts as beacon frame transmitter transmits its own beacon signal, preferably in the select frequency channel for the beacon frame transmitter. The unit that acts as sensor station switches to the normal operation mode in the time slot defined by the beacon frames of the further beacon frame transmitter to which the beacon frame transmitter **60**, **62**, **64** is associated. Thus, no power expensive synchronization is needed.

FIG. **7** shows a state diagram of operation of the paired units of FIG. **6** during discovery and association. The paired units **60**, **62** start in a first state **71** wherein a second unit **62** of

the pair **60**, **62** that acts like a sensor station transmits discovery messages. When the second unit **62** detects that a beacon frame from a beacon frame transmitter **12** or association unit **10** has been received, the paired units switch to second state **72**. In second state **72** the first unit **60** of the pair that acts as a beacon frame transmitter starts transmitting beacon frames in response to received discovery messages by transmitting a discovery confirm message. In these beacon frames the first unit **60** includes information to indicate that the first unit **60** is not yet associated.

Also in second state **72** the second unit **62** starts monitoring whether it receives beacon frames with information that indicates that the station that transmits the beacon frame has been associated (or is association unit).

When the second unit **62** detects a beacon frame with information that indicates that the unit that transmits the beacon frame has been associated, the paired units **60**, **62** switch from second state **72** to a third state **73**. Herein the second unit **62** starts transmitting association requests. When the second unit **62** receives an association confirmation in response, the paired units **60**, **62** switch to a fourth state **74**. In second, third and fourth state **72**, **73** and **74** first unit **60** starts and restarts counting a time out interval. The time out interval is restarted each time when first unit **60** receives a discovery message from a new source (sensor station or other beacon frame transmitter). The time out interval expires after a predetermined duration (e.g. a minute or more) wherein first unit **60** has received no discovery message from any new source. After expiration of the time out interval unit **60** stops transmitting beacon frames in response to received discovery messages.

From fourth state **74**, after expiry of the time out interval the paired units **60**, **62** switch to a fifth state **75** wherein second unit **60** transmits a beacon frame transmitter association request for the first unit **62**, listing the sources from which the first unit has received a discovery message. This beacon frame transmitter association request is forwarded to association unit **10**, which grants or denies the request, dependent on the list (granting the request for example if more than a threshold number of sources are listed that have not yet been associated). When the beacon frame transmitter association request is denied, new beacon frame transmitter association requests may be sent. Association unit **10** may be configured to lower the threshold with time so that the same request may be granted later if a source that is mentioned in the request cannot be reached in any other way even after an extended period of time.

Association unit **10** sends a grant message with a grant signal to the first unit **60** via the second unit **62**. When the grant message is received the paired units **60**, **62** switch to a sixth state **76**. Wherein first unit **62** starts transmitting beacon frames indicating that it has been associated, after which it can cause other units to switch from the first state **72** to the third state **73**.

Although completely separate units **60**, **62** are shown, it should be appreciated that part or all of these units may use shared circuits. Use of separate circuits has the advantage that a standard sensor station unit may be used. The use of a flexible connection has the advantage that the different units may be located "around a corner" from each other, so that one unit has contact with a space to which the other has no contact. Thus, a system in a building can be realized very easily with this type of beacon frame transmitter.

Although the network has been described with beacon frame transmitters **12**, it should be appreciated that instead of beacon signal synchronized transmission other forms of message exchange may be used. In this case beacon frame transmitters **12** act simply as stations. In that case beacon frame requests are only discovery requests and beacon frame transmissions responses to discovery requests.

Furthermore, although an example has been discussed wherein the beacon frame transmitters **12** merely act to relay sensor data, it should be appreciated that the beacon frame transmitters **12** themselves may also be equipped with sensors, so that the beacon frame transmitters **12** can also transmit their own sensor data.

Although a network of sensor stations has been described (each provided with a temperature sensor, and/or a smoke sensor and/or a light intensity sensor and/or a moisture sensor and/or a gas sensor etc.) has been described it should be appreciated that the described techniques can also be applied to non-sensor or part-sensor networks wherein sensor stations **14** are simply information handling stations.

Although an association process with a discovery phase (FIG. **5***a*) and an association phase has been described, it should be appreciated that the two phases can be integrated partly or wholly. In one embodiment beacon frame requests **50***a* double as association requests **54***a*. In another embodiment association grant messages **56***a* double as beacon transmissions **52***a*. Both embodiments may be combined. However, the use of separate phases has the advantage that the whole process runs more quickly.

The invention claimed is:

1. A wireless communication network that comprises an association unit and a plurality of stations, each station being configured to start up in a not-associated state and to transmit association request messages in the not-associated state, each station being configured to switch to an associated state upon reception of an association grant in response to one of its association request messages, the association request message and/or the association grant in response thereto establishing an operating route associated with the station for exchanging operating messages with the association unit during subsequent operation, the operating route associated with the station running through a source of the association grant in response to said one of the association request messages and, if the source of the association grant is not the association unit, the operating route between the source of the association grant and the association unit previously established for the source of the association grant in response to association request messages and/or the association grants for intermediate stations, in which network

the association unit is configured to transmit association grants as responses to received ones of the association request messages;

at least part of the stations is configured to become active to transmit association grants as responses to received ones of the association request messages, but only after switching to the associated state upon an association grant from the association unit or one of the at least part of the stations switched to the associated state directly or indirectly based on an association grant from the association unit.

2. A wireless communication network according to claim **1**, wherein said part of the stations comprises at least one dual function station, said dual function station being configured to transmit association grants only after receiving a secondary association grant, the dual function station being configured to transmit one or more secondary association request messages to the association unit along its operating route when in the associated state, only the association unit being configured to originate the secondary association grant in response to the secondary association request message.

3. A wireless communication network according to claim 2, wherein the dual function station is configured to transmit information to the association unit indicating from which of the stations said dual function station has received transmissions, the association unit being configured to select whether to grant the secondary association request dependent on said information.

4. A wireless communication network according to claim 3, wherein the stations are configured to transmit discovery messages before switching to the associated state, said transmissions comprising at least the discovery messages.

5. A wireless communication network according to claim 2, wherein the dual function station comprises a first and second unit, each with at least a respective antenna, coupled via a mechanically flexible connection, the first unit being configured to transmit first ones of the association request messages and to receive first ones of the association grants, the second unit being configured to receive second ones of the association request messages and to the transmit second ones of the association grants in response to the first ones of the association request messages.

6. A wireless communication network according to claim 2, wherein the dual function station is configured to operate using respective, mutually distinct first and second frequency channels, using the first frequency channel for transmitting association request messages and receiving association grants, and using the second frequency channel for receiving association request messages and transmitting association grants.

7. A wireless communication network according to claim 1, wherein each station from said at least part of the stations is configured to transmit discovery advertisement messages including an identification of the station from said at least part of the stations, the stations that transmit association request messages being configured to address the association request messages using identifications from the discovery advertisement messages.

8. A wireless communication network according to claim 7, wherein each station in said at least part of the stations is configured to indicate, in the discovery advertisement messages, whether the station is enabled to transmit association grants, the stations that transmit association requests being configured to transmit association requests addressed to stations from said at least part of the stations only if the addressed station has indicated in at least one of the discovery advertisement messages that it is enabled to transmit association grants.

9. A wireless communication network according to claim 1, wherein said at least part of the stations are configured to transmit route information toward the association unit after transmitting an association grant, the route information defining the associated route for the station to which the association grant has been sent.

10. A wireless communication network according to claim 9, wherein said at least part of the stations is configured to forward at least part of the route information towards the association unit, to store at least a further part of the route information and to route subsequent messages from the association unit to the station for which the route information was sent according to said stored at least further part.

11. A station for use in a wireless communication network, the station comprising a transmitter arrangement, a receiver arrangement and a processor circuit coupled to the transmitter arrangement and the receiver arrangement, the processor circuit being configured to start up in a not-associated state and to cause the transmitter arrangement to transmit association request messages in the not-associated state, the processor circuit being configured to switch to an associated state upon reception at the receiver arrangement of an association grant in response to one of the association request messages, the processor circuit being configured to establish an associated route for communication with an association unit during subsequent communication of operating messages to an association unit, the associated route including a source of the association grant, and, if the source of the association grant is not the association unit, the operating route between the source of the association grant and the association unit previously established for the source of the association grant in response to association request messages and/or the association grants for intermediate stations, the processor circuit being configured to cause the transmitter arrangement to initiate and transmit association grants in response to association request messages, but only after switching to the associated state.

12. A station according to claim 11, wherein the processor circuit is configured, when switched to the associated state, to cause the transmitter arrangement to transmit a secondary association request message to the association unit, and to cause the transmitter arrangement to transmit the association grants only after receiving a secondary association grant in response to said secondary association request message.

13. A station according to claim 11, wherein the processor circuit is configured to send information to the association unit indicating from which one of the stations said dual function station has received messages prior to receiving the secondary association grant.

14. A method of operating a wireless communication network that comprises an association unit and a plurality of stations, the method comprising

installing the stations in a not-associated state;

transmitting association request messages from stations in the not-associated state;

transmitting the association grants in response to the association request messages,

switching each station to an associated state when the station has received an association grant in response to one of its association request messages;

storing route information defining a route between the station that has switched to the associated state and the association unit, the route including the station that has sent the associated grant and, if the source of the association grant is not the association unit, the operating route between the source of the association grant and the association unit previously established for the source of the association grant in response to association request messages and/or the association grants for intermediate stations; wherein

each association grant is transmitted only from the association unit or from an intermediate station, the latter only after said intermediate station has switched to the associated state upon an association grant from the association unit or one of the at least part of the stations switched to the associated state directly or indirectly based on an association grant from the association unit.

15. A method according to claim 14, wherein the intermediate station is a dual function station, the method comprising

transmitting a secondary association request message from the dual function station when the dual function is in the associated state;

transmitting a secondary association grant originating from the association unit in response to the secondary association request message;

enabling the dual function station to transmit the association grants only upon receiving the secondary association grant.

16. A method according to claim 15, comprising

transmitting information from the intermediate station to the association unit about from which one of the stations the intermediated station has received transmissions;

selecting, in the association unit, whether to grant the secondary association request dependent on said information.

17. A method according to claim 16, further comprising transmitting discovery messages from the stations before switching to the associated state, said transmissions comprising the discovery messages.

18. A method according to claim 15, further comprising the steps of:

transmitting discovery advertisement messages from at least part of the stations, including an identification of the station that has transmitted the discovery advertisement message;

addressing the association request messages using identifications from the discovery advertisement messages.

19. A method according to claim 18, further comprising the steps of:

including an indication in the discovery advertisement messages, whether the station that transmits the discovery advertisement message is enabled to transmit association grants,

transmitting association requests addressed to stations only if the addressed station has indicated that it is enabled to transmit association grants.

20. A method according to claim 15, further comprising switching the intermediate station to operate at a first and second, mutually different transmission frequency after switching to the associated state, for directly or indirectly communicating with the association unit and for communicating with its associated stations respectively.

* * * * *

EXHIBIT 2

US008438120B2

(12) **United States Patent**
Raaijmakers

(10) **Patent No.: US 8,438,120 B2**
(45) **Date of Patent: May 7, 2013**

(54) **MACHINE LEARNING HYPERPARAMETER ESTIMATION**

(75) Inventor: **Stephan Alexander Raaijmakers**, Amsterdam (NL)

(73) Assignee: **Nederlandse Organisatie voor toegepast-natuurwetenschappelijk Onderzoek TNO**, Delft (NL)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 607 days.

(21) Appl. No.: **12/597,257**

(22) PCT Filed: **Apr. 25, 2008**

(86) PCT No.: **PCT/NL2008/050247**

§ 371 (c)(1),
(2), (4) Date: **Jul. 8, 2010**

(87) PCT Pub. No.: **WO2008/133509**

PCT Pub. Date: **Nov. 6, 2008**

(65) **Prior Publication Data**

US 2010/0280979 A1 Nov. 4, 2010

(30) **Foreign Application Priority Data**

Apr. 25, 2007 (EP) 07106963
Jul. 9, 2007 (EP) 07112037

(51) **Int. Cl.**
G06N 5/00 (2006.01)

(52) **U.S. Cl.**
USPC **706/12**; 706/45

(58) **Field of Classification Search** 706/12, 706/45

See application file for complete search history.

(56) **References Cited**

PUBLICATIONS

Duckstein, et al., Bayesian forecasting of hydrologie variables under changing climatology, the Influence of Climate Change and Climatic Variability on the Hydrologie Regime and Water Resources (Proceedings of the Vancouver Symposium, Aug. 1987). IAHSPubl. No. 168, 1987, pp. 301-311.*

Raaijmakers, "Sentiment classification with interpolated information diffusion kernels," Proceedings of the 1st International Workshop on Data Mining and Audience Intelligence for Advertising, pp. 34-39, (Aug. 12, 2007).

AMIDA Project: "Chapter 3: subjectivity recognition" Deliverable 5.2: Multimodal Content Abstraction, 2007, pp. 28-47.

De Boer et al., "A Tutorial on the Cross-Entropy Method," Annals of Operating Research, vol. 134, No. 1, (Feb. 2005), pp. 19-67.

Mannor et al., "The cross entropy method for classification," Proceedings of the 22nd International Conference on Machine Learning (Aug. 9, 2005) (the whole document).

International Search Report dated Oct. 9, 2008 for PCT/NL2008/050247.

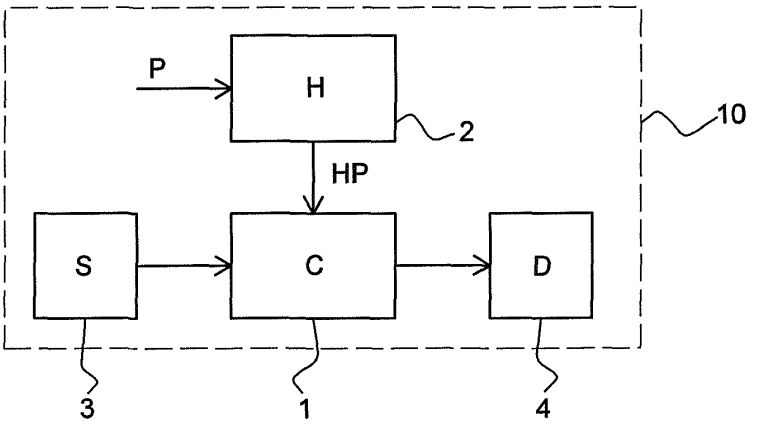
* cited by examiner

Primary Examiner — Wilbert L Starks
(74) *Attorney, Agent, or Firm* — Leydig, Voit & Mayer, Ltd.

(57) **ABSTRACT**

A method of determining hyperparameters (HP) of a classifier (**1**) in a machine learning system (**10**) iteratively produces an estimate of a target hyperparameter vector. The method comprises the steps of selecting from the random sample the hyperparameter vector producing the best result in the present and any previous iterations, and updating the estimate of the target hyperparameter vector by using said selected hyperparameter vector. The random sample may be restricted by using the hyperparameter vector producing the best result in the present and any previous iterations.

16 Claims, 1 Drawing Sheet



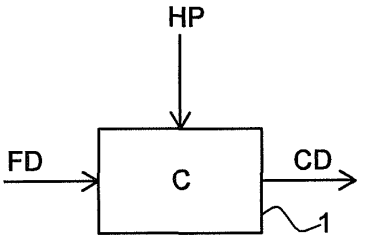


FIG. 1

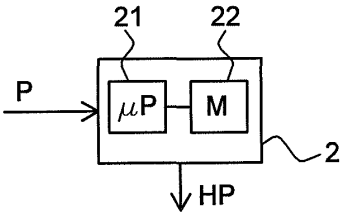


FIG. 2

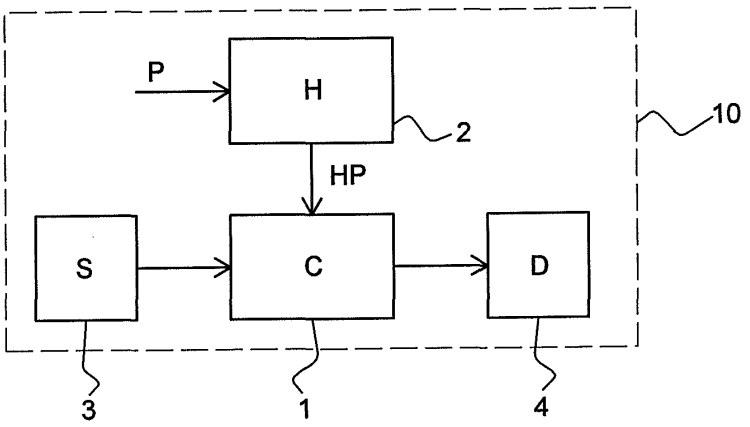


FIG. 3

MACHINE LEARNING HYPERPARAMETER ESTIMATION

FIELD OF THE INVENTION

The present invention relates to hyperparameter estimation. More in particular, the present invention relates to a method and device for determining hyperparameters of classifiers in machine learning systems and applications.

BACKGROUND

Classifiers are used in machine learning systems to classify physical objects and/or their (typically digital) representations. Machine learning systems may, for example, be used to assist fruit picking robots. The classifier of the machine learning system is trained to distinguish ripe fruit (e.g. tomatoes) from unripe fruit. For each fruit item, the classifier determines a set of parameters which are compared with stored parameters in order to classify the items as "ripe" or "unripe". The classification process is, in turn, controlled by hyperparameters which determine, for example, decision criteria such as threshold values.

Classifiers are typically trained using a training set prepared on the basis of human input: an operator indicates the correct classification for the items of the training set. On the basis of these training data, the correct hyperparameters of the classifier can be estimated.

Hyperparameters may be determined using various methods, for example heuristic or statistical optimisation methods. However, many heuristic methods are ad hoc methods, while not all statistical optimisation methods are suitable for determining hyperparameters.

A particularly efficient yet relatively simple method for estimating parameters in general is the cross-entropy (CE) method. This iterative method comprises the repeated steps of drawing, in a parameterised way, a random sample of candidate solutions, and updating the parameters on the basis of the random sample. At the time of writing, the paper "A Tutorial on the Cross-Entropy Method" by P. T. de Boer et al. could be found at http://iew3.technion.ac.il/CE/tutor.php, said paper is herewith incorporated in this document in its entirety.

The paper "The cross-entropy method for classification" by S. Mannor, D. Peleg and R. Rubinstein, Proceedings of the 22nd International Conference on Machine Learning, 2005, discloses an application of the CE algorithm for searching the space of support vectors. Hyperparameter values are determined by using a simple grid search, not by using the CE algorithm, for the reasons discussed below.

The cross-entropy method as described in the above-mentioned papers is unfortunately not suitable for determining hyperparameters. Determining optimal sets of hyperparameter values is a difficult problem due to the extremely large size of the search space: the optimal solution typically is a rare event. While the cross-entropy method is geared towards sampling rare event spaces, it is not a priori clear how the process of drawing hyperparameter values can be parameterised, as the hyperparameter samples are not classical parameterised probability density functions.

SUMMARY OF THE INVENTION

It is an object of the present invention to overcome these and other problems of the Prior Art and to provide a method and device for determining hyperparameters which allow efficient methods similar to the cross-entropy method to be utilized.

Accordingly, the present invention provides a method of determining hyperparameters of a classifier in a machine learning system by iteratively producing an estimate of a target hyperparameter vector, each iteration comprising the steps of:

- drawing a random sample of hyperparameter vectors from a set of possible hyperparameter vectors, and
- updating the estimate of the target hyperparameter vector by using the random sample,

the method being characterised in that each iteration comprises the step of:

- selecting from the random sample a hyperparameter vector producing the best result in the present and any previous iterations,

and in that the step of updating the estimate of the target hyperparameter vector involves said selected hyperparameter vector.

By selecting from the random sample a or the hyperparameter vector producing the best result in the present and any previous iterations, and using said selected hyperparameter vector to update the estimate of the target hyperparameter vector, the properties of the best performing hyperparameters are used to guide the next iteration. In this way, a method similar to the cross-entropy method can be used effectively for determining hyperparameters, even when the hyperparameters are not continuous and their effects are not transparent. By using the method according to the present invention, an improved classification accuracy is achieved.

In the method of the present invention, hyperparameter vectors obtained through randomly sampling a set of hyperparameter vectors are used to estimate a desired (that is, target) hyperparameter vector to be used in the classifier. Each hyperparameter vector preferably comprises two or more elements, that is hyperparameters, but it is also possible for a hyperparameter vector to contain only a single hyperparameter. Although the samples are preferably completely random, pseudo-random samples may be used instead.

In a preferred embodiment, the method according to the present invention further comprises the steps of:

- selecting from the random sample a further hyperparameter vector producing the best result in any previous iterations, and
- restricting the random sample of hyperparameter vectors by using the further selected hyperparameter vector.

The step of restricting the random sample preferably involves the use of an interval surrounding the further selected hyperparameter vector which produces the best result in any previous iteration.

It is preferred that the step of restricting the random sample is carried out prior to the step of determining the hyperparameter vector producing the best result in the present and any previous iterations. However, in alternative embodiments, the step of restricting the random sample is carried out during the step of updating the estimate of the target hyperparameter vector.

In a preferred embodiment, E^t is the hyperparameter vector X^t_i at iteration t producing the best result $S(X^t_i)$ in the present iteration t and all previous iterations (if any). The random sample X^t_i has elements X^t_{ij}, while the step of updating the

hyperparameters comprises the step of determining the (target) hyperparameter v^t_j, where

$$v^t_j = \frac{\sum_{i=1}^{n} I\{S(X^t_i) \geq \gamma^t\} W(X^t_i; E^t) X^t_{ij}}{\sum_{i=1}^{n} I\{S(X^t_i) \geq \gamma^t\} W(X^t_i; E^t)},$$

wherein γ^t is a threshold value, W is a weighting function and $I\{S(X_i^t) \geqq \gamma^t\}$ is an indicator function which is equal to 1 if $S(X_i^t)$ greater than or equal to the threshold value γ^t, and else is equal to 0. The weighting function W is preferably given by

$$W(X^t_i; E^t) = 1 - \frac{\sqrt{\sum_{j=1}^{m} (X^t_{ij} - E^t_j)^2}}{\sqrt{\sum_{j=1}^{m} (X^t_{ij})^2} \sqrt{\sum_{j=1}^{m} (E^t_j)^2}}.$$

Accordingly, the step of updating the hyperparameters may involve a weighting function based upon a distance function, preferably a Euclidean distance function. It is noted that when the Euclidean distance between E^t and X^t_{ij} is zero, the weighting function W is equal to one.

The method according to the present invention is typically carried out by computer apparatus, for example a general purpose computer system comprising a processor and an associated memory, one part of the memory storing a software program for instructing the processor to carry out the method steps of the present invention, and another part of the memory storing data, said data comprising the hyperparameter values referred to above.

The present invention also provides a computer program product for carrying out the method as defined above. A computer program product may comprise a set of computer executable instructions stored on a data carrier, such as a CD or a DVD. The set of computer executable instructions, which allows a programmable computer to carry out the method as defined above, may also be available for downloading from a remote server, for example via the Internet.

The present invention additionally provides a classifier for use in a machine learning system, the classifier being arranged for using hyperparameters as control parameters, wherein the hyperparameters have been determined by the method according to any of the preceding claims. The classifier is preferably embodied in software, but may also be embodied in hardware, or in a combination of hardware and software.

The present invention still further provides a device for determining hyperparameters of a classifier in a machine learning system, the device comprising a processor arranged for iteratively producing estimates of a target hyperparameter vector, each iteration comprising the steps of:

drawing a random sample of hyperparameter vectors from a set of possible hyperparameter vectors, and

updating the estimate of the target hyperparameter vector by using the random sample,

the device being characterised in that the processor is arranged such that each iteration comprises the step of:

selecting from the random sample a hyperparameter vector producing the best result in the present iteration and any previous iterations, and

in that the step of updating the estimate of the target hyperparameter vector involves said selected hyperparameter vector.

The processor may be arranged for carrying out the steps mentioned above by providing a suitable software program in the memory associated with the processor. Alternatively, or additionally, the processor may be designed specifically to carry out the steps mentioned above.

A machine learning system comprising a device as defined above and/or arranged for carrying out the method as defined above is also provided by the present invention. The machine learning system may comprise hardware and/or software components.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention will further be explained below with reference to exemplary embodiments illustrated in the accompanying drawings, in which:

FIG. **1** schematically shows a classifier having hyperparameters as input control variables.

FIG. **2** schematically shows a device for determining hyperparameters according to the present invention.

FIG. **3** schematically shows a machine learning system according to the present invention.

DETAILED DESCRIPTION OF THE DRAWINGS

The classifier (C) **1** shown schematically in FIG. **1** receives feature data FD and hyperparameters HP, and produces classification data CD. The feature data FD may include image data produced by a CCD (Charge Coupled Device) array or other light-sensitive device. Additionally, the feature data FD may include sound data or data concerning database items, such as library data (book numbers and titles). The classifier may be constituted by an apparatus (hardware embodiment), but is preferably constituted by a software program running on a computer apparatus, for example a general purpose computer ("personal computer"), a dedicated computer, or other suitable apparatus.

The classifier **1** is capable of classifying the items represented by the input feature data FD. The feature data FD typically comprise parameters of the items, for example their size and/or colour in the case of fruit or machine parts. Under control of the hyperparameters HP the items are each assigned to a class, for example "ripe" in the case of fruit. At least two different classes are used, and typically an item is assigned to one class only.

The hyperparameters HP determine the decision function of the classifier that classifies items. For example, when this function is based upon thresholding, items producing an aggregate value (derived from the feature values) equal to or greater than the threshold are assigned to class A, while items producing a value smaller than the threshold are assigned to class B. It is noted that other classification techniques may be used as well.

Applications of the classifier **1** include picking fruit, sorting parts (e.g. machine parts), classifying books, selecting songs, recognising adult video material, detecting damage, etc.

The present invention allows a method similar to the cross-entropy (CE) method to be used for estimating and determining hyperparameters. To this end, the present invention proposes to include a memory facility into the algorithm by preserving samples which produce a good result (the result being defined by a suitable function, such as a loss function). Additionally, or alternatively, the sampling of candidate

hyperparameter settings is made dependent on the (hyper) parameter vector updated by the algorithm, and/or a suitably weighting function (also known as "change of measure" or "likelihood ratio" in the CE algorithm) guiding the search process should be provided.

The method of the present invention uses parameters N, ρ and μ which are typically predetermined. It is noted that these parameters may be thought of as hyperparameters of the method but are distinct from the hyperparameters to be determined by the method. The parameter N represents the number of random draws; the parameter N may for example be equal to 10, 100 or 1000.

In the iterations a parameter n is used which represents the size of each data vector, that is, the number of elements X_{ij} of each hyperparameter vector X_i of the set of possible hyperparameter vectors. The parameter n may for example be equal to 5, 10 or 100, although other numbers are also possible. The parameter ρ represents a fraction and may have a value ranging from 0.2 to 0.01, although other values may also be used. The width parameter μ, which will be explained in more detail below, indicates a preferred sampling interval.

The initial set of target hyperparameters, the hyperparameter vector u, can typically be chosen arbitrarily. Each estimated (target) hyperparameter vector v^t at point in time (or iteration number) t consists of hyperparameters v^t_j.

In the present invention, the "best" hyperparameter vector v of iterations 1 . . . t is kept for use in the next iteration and is denoted E^t. This "best" (or "elitist") hyperparameter vector is found using a function $S(X^t)$ which may be a loss function or any other suitable function producing a result value based upon X^t. The value of $S(X^t)$ may also be viewed as an importance measure. The value of X^t corresponding with the maximum value of $S(X^t)$ is denoted argmax $S(X^t)$. In addition, the values of $S(X^t)$ are ranked and are used to determine a threshold γ^t, involving ρ and N to determine the $((1-\rho)N)^{th}$ value of the ranked values of $S(X^t)$. It is noted that the number $((1-\rho)N)$ is not necessarily an integer and is therefore rounded up, expressed mathematically as $\lceil((1-\rho)N\rceil$.

Subsequently, the (target) hyperparameters are determined using an expression involving indicator functions $I\{S(X^t_i)\geqq\gamma^t\}$ and weighting functions $W(X_i; E^t)$. The indication functions are equal to 1 when $S(X^t_i)\geqq\gamma^t$ and equal to 0 when $S(X^t_i)<\gamma^t$. The weighting functions $W(X_i; E^t)$, also known as "change of measure" functions, have the data vector elements X_i (actually X^t_i) and E^t as input variables. This implies that the best data vector E^t of the present iteration is used to determine the (target) hyperparameters.

The weighting function $W(X^t_i; E^t)$ is preferably defined as:

$$W(X^t_i; E^t) = 1 - \frac{\sqrt{\sum_{j=1}^{m}(X^t_{ij} - E^t_j)^2}}{\sqrt{\sum_{j=1}^{m}(X^t_{ij})^2}\sqrt{\sum_{j=1}^{m}(E^t_j)^2}} \quad (1)$$

This function produces a value of 1 when the Euclidean distance between a sample and the "best" sample is 0.

Accordingly, a preferred embodiment of the method according to the present invention comprises the following steps:

1. Choose initial parameters N, ρ and μ and generate an initial (target) hyperparameter vector u.
2. Let $v^0=E^0=u$ and set iteration number t=1.
3. Generate a random sample $X^t_1, \ldots, X^t_n$ of hyperparameters drawn from a (given) set of possible hyperparameters. The random sample preferably is restricted by v^{t-1}. Compute the performance $S(X^t_i)$ for every value of i and rank the results: $S_{(1)}\leqq \ldots \leqq S_{(n)}$. Compute $\gamma^t=S_{percentile}$ and compute $E^t=\text{argmax } v_{i=1 \ldots n}\gamma^t$.
4. For every (target) hyperparameter v^t_j let

$$v^t_j = \frac{\sum_{i=1}^{n} I\{S(X^t_i) \geq \gamma^t\} W(X^t_i; E^t) X^t_{ij}}{\sum_{i=1}^{n} I\{S(X^t_i) \geq \gamma^t\} W(X^t_i; E^t)} \quad (2)$$

5. Let t:=t+1, repeat from step 3 until the stopping condition is met.

The final set v of j (target) hyperparameters v_j is the desired optimal set.

By using the best sample E^t_j in hyperparameter estimation/update formula (2), the better results are reinforced, leading to a more efficient convergence of the method. In addition, discontinuity problems associated with hyperparameters are overcome by the method of the present invention.

The stopping criterion (or condition) is preferably a non-changing value of the parameter γ^t during a number of iterations, but could also be a threshold value. The value $S_{percentile}$ is determined using the $((1-\rho)N)^{th}$ value of S in the ranking $S_{(1)}\leqq \ldots \leqq S_{(n)}$, where rounding up may be used if this value is not an integer. The weighting function $W(X^t_i; E^t)$ used in formula (2) is preferably defined by formula (1) above.

The method of the present invention may be summarized as deriving a target hyperparameter vector from a random sample of hyperparameter vectors while taking the performance of the hyperparameters of the random sample into account.

The restriction which may be applied when generating a random sample at time t in step 3 may be defined by the interval:

$$[v^t_j*(1-\mu), v^t_j*(1+\mu)] \quad (3)$$

where μ is a width parameter and "*" denoted multiplication. If the sample X^t_i lies within this interval, its distance relative to the best sample E^t_j of the previous iteration is limited to the value of μ. It is noted that the use of this interval is preferred but not essential as the method of the present invention could be carried out without using this interval.

The device **2** for determining hyperparameters comprises a processor (μP) **21** and an associated memory (M) **22**. The memory stores both instructions for the processor and data. The device **2** receives parameters P (more in general: feature data), such as the parameters ρ and μ mentioned above, and outputs hyperparameters HP. The stored instructions allow the processor to carry out the method according to the present invention.

The machine learning system **10** shown merely by way of non-limiting example in FIG. **3** comprises a classifier (C) **1** and a device (H) **2** for determining hyperparameters. The machine learning system **10** may further comprise at least one sensor (S) **3** and a display screen (D) **4**. The sensor **3** serves to collect data while the display screen **4** serves to display classification result and other data.

The present invention is based upon the insight that methods similar to the cross-entropy method can be adapted for hyperparameter estimation by adding a best performance preservation feature. The present invention benefits from the additional insight that the best performance in a particular iteration may be preserved by restricting the sample using the best performance of the previous iterations.

It is noted that any terms used in this document should not be construed so as to limit the scope of the present invention. In particular, the words "comprise(s)" and "comprising" are not meant to exclude any elements not specifically stated. Single (circuit) elements may be substituted with multiple (circuit) elements or with their equivalents.

It will be understood by those skilled in the art that the present invention is not limited to the embodiments illustrated above and that many modifications and additions may be made without departing from the scope of the invention as defined in the appending claims.

The invention claimed is:

1. A method of determining hyperparameters of a classifier in a machine learning system by iteratively producing an estimate of a target hyperparameter vector, each iteration comprising the steps of:

drawing a random sample of hyperparameter vectors from a set of possible hyperparameter vectors,

updating the estimate of the target hyperparameter vector by using the random sample, and

selecting, from the random sample of hyperparameter vectors, a hyperparameter vector producing a best result in the present and any previous iterations, and wherein the step of updating the estimate of the target hyperparameter vector uses said hyperparameter vector producing the best result.

2. The method according to claim **1**, further comprising the steps of:

selecting a further hyperparameter vector producing the best result in any previous iterations, and

restricting the random sample of hyperparameter vectors by using the further selected hyperparameter vector.

3. The method according to claim **2**, wherein the step of restricting the random sample of hyperparameter vectors involves using an interval surrounding the further selected hyperparameter vector.

4. The method according to claim **3**, wherein the step of restricting the random sample of hyperparameter vectors is carried out prior to the step of selecting the hyperparameter vector producing the best result in the present and any previous iterations.

5. The method according to claim **3**, wherein the step of restricting the random sample of hyperparameter vectors is carried out during the step of updating the estimate of the target hyperparameter vector.

6. The method according to claim **2**, wherein the step of restricting the random sample of hyperparameter vectors is carried out prior to the step of selecting the hyperparameter vector producing the best result in the present and any previous iterations.

7. The method according to claim **2**, wherein the step of restricting the random sample of hyperparameter vectors is carried out during the step of updating the estimate of the target hyperparameter vector.

8. The method according to claim **1**, wherein the step of updating the estimate of the hyperparameter vector uses a weighting function.

9. The method according to claim **1**, wherein E^t is the selected hyperparameter vector X^t_i at iteration t producing the best result $S(X^t_i)$, the selected hyperparameter vector X^t_i having elements X^t_{ij}, wherein the step of updating the estimate of the target hyperparameter vector comprises the step of determining the hyperparameter v^t_j, where

$$v^t_j = \frac{\sum_{i=1}^{n} I\{S(X^t_i) \geq \gamma^t\} W(X^t_i; E^t) X^t_{ij}}{\sum_{i=1}^{n} I\{S(X^t_i) \geq \gamma^t\} W(X^t_i; E^t)},$$

wherein γ^t is a threshold value and W is a weighting function.

10. The method according to claim **9**, wherein the weighting function W is given by

$$W(X^t_i; E^t) = 1 - \frac{\sqrt{\sum_{j=1}^{m} (X^t_{ij} - E^T_j)^2}}{\sqrt{\sum_{j=1}^{m} (X^t_{ij})^2} \sqrt{\sum_{j=1}^{m} (E^t_j)^2}}.$$

11. The method according to claim **1**, wherein the steps are carried out by a programmed computer apparatus including a processor and a computer readable medium including computer executable instructions.

12. A classifier for use in a machine learning system, the classifier using hyperparameters as control parameters, wherein the hyperparameters are determined according to the set of steps recited in claim **1**.

13. A non-transitory computer readable medium product including computer-executable instructions for carrying out a method of determining hyperparameters of a classifier in a machine learning system by iteratively producing an estimate of a target hyperparameter vector, each iteration comprising the steps of:

drawing a random sample of hyperparameter vectors from a set of possible hyperparameter vectors,

updating the estimate of the target hyperparameter vector by using the random sample, and

selecting, from the random sample of hyperparameter vectors, a hyperparameter vector producing a best result in the present and any previous iterations, and wherein the step of updating the estimate of the target hyperparameter vector uses said hyperparameter vector producing the best result.

14. A device for determining hyperparameters of a classifier in a machine learning system, the device comprising a processor arranged for iteratively producing an estimate of a target hyperparameter vector, each iteration comprising the steps of:

drawing a random sample of hyperparameter vectors from a set of possible hyperparameter vectors,

updating the estimate of the hyperparameter vector by using the random sample, and

selecting, from the random sample of hyperparameter vectors, a hyperparameter vector producing a best result in the present and any previous iterations, and wherein the step of updating the estimate of the target hyperparameter vector uses said hyperparameter vector producing the best result.

15. The device according to claim **14**, wherein the processor is further arranged for:

selecting a further hyperparameter vector producing the best result in any previous iterations, and

restricting the random sample by using the hyperparameter vector producing the best result in any previous iteration.

16. A machine learning system, comprising the device for determining hyperparameters of a classifier defined in claim **14**.

* * * * *

EXHIBIT 3

[redacted]

From: Shaun Hassett <shaunhassett@potterminton.com>
Sent: Friday, May 23, 2025 5:15 PM
To: GOOGLE-KMIZRA
Cc: Mike Jones
Subject: FW: Mizra v. Google - motion for leave to amend

This Message Is From an External Sender

If you are concerned about the message's content, highlight the email in your inbox and click "Report Suspicious" in the Outlook ribbon -or- contact 6Help.

Report Suspicious

Team,

Mike and I just received the below email from Michael Smith. I'll let Michael know we will get back to him in due course.

Best,

Shaun

From: Michael Smith <Michael.Smith@Solidcounsel.com>
Sent: Friday, May 23, 2025 5:03 PM
To: Mike Jones <mikejones@potterminton.com>; Shaun Hassett <shaunhassett@potterminton.com>
Subject: Mizra v. Google - motion for leave to amend

Mike/Shaun,

K.Mizra is planning to seek leave to amend our complaint to make several changes, and we wanted to see if the amendments would be opposed or could be presented an unopposed. In the new complaint, we amend the infringement and damage allegations for the previously asserted '705 patent, we remove from the case the previously asserted '048 patent, and add two new claims of infringement.

K.Mizra recently decided that there is insufficient term remaining on the '048 Patent to keep it in the case. K.Mizra also recently completed its factual investigation concerning Google's infringement of two additional K.Mizra patents: U.S. Patent No. 8,144,717 ("the '717 Patent"), directed to tuning of Hyperparameters used in artificial intelligence models, and U.S. Patent No. 8,438,120 ("the '120 Patent"), directed to mesh networking technology. For the sake of efficiency, would like to assert those claims in this action.

We decided that while it seek those changes, it should also address Google's complaints concerning the allegedly deficient '705 patent factual allegations. Accordingly, K.Mizra seeks leave to file an amended complaint which drops the '048 Patent, adds claims for infringement directed to the '717 and the '120 Patents, and amends the allegations concerning Google's direct infringement of the '705 Patent.

Let me know if you have any questions I can answer.

Michael C. Smith PARTNER

SCHEEF & STONE
SOLID COUNSEL

Michael.Smith@solidcounsel.com | 903.938.8900

Marshall office: 113 East Austin Street, Marshall, Texas 75670

Dallas office: 500 N. Akard Street, Suite 2700, Dallas, Texas 75201
Frisco office: 2600 Network Boulevard, Suite 400, Frisco, Texas 75034

Eastern District Texas Blog - www.EDTexweblog.com
Texas IP Law Blog - www.TexIPLawBlog.com
Texas Business Courts Blog - www.TexasBusinessCourtsBlog.com

ABOTA FOUNDATION
SENIOR LIFE FELLOW

TEXAS BAR FOUNDATION
LIFE FELLOW

Important: This email and any attached files contain information intended for the exclusive use of the person(s) to whom it is addressed and may contain information that is confidential. If you are neither the intended recipient nor responsible for delivering this email to the recipient, any dissemination or copying of this communication is strictly prohibited. Please notify the sender of any unintended recipients and delete the original message without making any copies.

K.Mizra LLC, Ex. 2001, p. 38
Google LLC v. K.Mizra LLC
IPR2025-01437

EXHIBIT 4

DRAFT
5/28/2025

IN THE UNITED STATES DISTRICT COURT
FOR THE WESTERN DISTRICT OF TEXAS
AUSTIN DIVISION

K.MIZRA LLC, Plaintiff, v. GOOGLE LLC, Defendant.	Civil Action No.: 1:25-cv-00236-ADA Jury Trial Demanded

FIRST AMENDED COMPLAINT FOR PATENT INFRINGEMENT

Plaintiff K.Mizra LLC ("K.Mizra") files this First Amended Complaint for patent infringement against Defendant Google LLC ("Google"), alleging as follows:

TABLE OF CONTENTS

INDEX OF EXHIBITS iii

I. INTRODUCTION 1

A. The Asserted Zero Trust Network Security Patent 1

B. The Asserted Hyperparameter Tuning Patent 3

C. The Asserted Mesh Networking Patent 3

D. K.Mizra's Licensing Model 4

II. THE PARTIES 4

III. JURISDICTION AND VENUE 5

IV. GENERAL ALLEGATIONS 5

A. The Zero Trust Network Security Patent and Google's Infringement 5

1. Prior Licensing and Litigation Concerning the Asserted Zero Trust Network Security Patent 6

2. Computer Network Security Problems In 2004 Solved by the Asserted Zero Trust Network Security Patents 6

3. K.Mizra's Asserted Zero Trust Network Security Patent Claims Are Presumed Valid 8

4. Failed IPRs Brought Against the Asserted Zero Trust Network Security Patents 13

5. Google's Infringing Chrome Enterprise Premium Product 14

B. The Asserted Hyperparameter Tuning Patent 17

1. Prior Licensing and Litigation Concerning the Hyperparameter Tuning Patent 17

2. Problems in 2007 Solved By the Asserted Hyperparameter Tuning Patent 18

3. K.Mizra's Asserted Hyperparameter Tuning Patent Claims Are Presumed Valid 22

4. Google's Infringing Vertex AI Studio Product 23

C. The Asserted Mesh Networking Patent 24

1. Prior Licensing and Litigation Concerning the Mesh Networking Patent 24

2. Problems in 2007 Solved By the Mesh Networking Patent 25

3. K.Mizra's Asserted Mesh Networking Patent Claims Are Presumed Valid 26

4. Google's Infringing Nest Devices 27

V. FIRST CLAIM FOR RELIEF (Patent Infringement Under 35 U.S.C. § 271 of the '705 Patent) 32

VI. SECOND CLAIM FOR RELIEF (Patent Infringement Under 35 U.S.C. § 271 of the '120 Patent) 50

VII. THIRD CLAIM FOR RELIEF (Patent Infringement Under 35 U.S.C. § 271 of the '717 Patent) 58

VIII. REQUEST FOR RELIEF 70

IX. DEMAND FOR JURY TRIAL 71

INDEX OF EXHIBITS

Ex	Description
1	U.S. Patent No. 8,234,705
2	U.S. Patent No. 8,234,705 – Preliminary Infringement Contention Claim Chart
3	Next '24_ Introducing Chrome Enterprise Premium, the future of endpoint security _ Google Cloud Blog
4	Chrome Enterprise Premium access protection overview _ BeyondCorp Enterprise _ Google Cloud
5	BeyondCorp Zero Trust Enterprise Security _ Google Cloud
6	U.S. Patent No. 8,438,120
7	U.S. Patent No. 8,144,717
8	Thread Play Services APIs Matter Google Home Developers
9	Connect Thread smart devices with Google TV Streamer (4K) - Android - Google TV Help
10	Nest Connect - Range Extender for Detect & Guard - Google Store
11	Nest debuts Nest Secure home system and has a Thread router called Nest Connect - Stacey on IoT _ Internet of Things news and analysis
12	Nest × Yale Lock - Key-Free Smart Deadbolt - Google Store_1
13	Pixel 9 series arrives at the FCC, includes Thread
14	Chrome Enterprise Premium documentation
15	Endpoint Verification Overview
16	Chrome Enterprise Premium Overview
17	How to prevent account takeovers with new certificate-based access
18	Understand Mutual TLS at Google Cloud
19	Deploy Endpoint Verification to use with certificate-based access
20	New Google API Will Securely Verify Chrome Devices
21	Chrome Verified Access
22	Context Aware Access insights and recommendations are now generally available
23	Block compromised mobile devices using context-aware access
24	Allow users to unblock apps with remediation messages in Context Aware Access
25	Vertex AI Platform Google Cloud
26	Vertex AI custom training overview Google Cloud
27	Vertex AI documentation Google Cloud
28	Overview of hyperparameter tuning Vertex AI Google Cloud
29	Create a hyperparameter tuning job for python package Vertex AI Google Cloud
30	Intro to Model Tuning: Grid and Random Search (excerpts)
31	Comparing Randomized Search and Grid Search for Hyperparameter Estimation in Scikit Learn GeeksforGeeks
32	OpenThread Homepage
33	Nest Hub Smart Displays for your Home - Google Store
34	Developing with OpenThread APIs
35	Network Discovery and Formation OpenThread
36	otNetworkDiagMleCounters Struct Reference OpenThread
37	Node Roles and Types OpenThread
38	https://openthread.io/guides/thread-primer/router-selection

I. INTRODUCTION

1. K.Mizra is a patent licensing company run by experienced management. The company focuses on high-value, high-quality patents and owns patent portfolios originating from a wide array of inventors, including patents and patent portfolios developed by well-known multinational corporations such as IBM, Intel, Rambus and others, as well as from research institutes such as Nederlandse Organisatie voor Toegespast Natuurwetenschappelijk Onderzoek (Netherlands Organization for Applied Scientific Research). By focusing on high-quality patents, K.Mizra provides a secondary market that enables inventors to recoup their research and development investments and to continue their innovations. K.Mizra offers licenses to its patents on reasonable terms and, in this way, plays an important role in the development of technologies that improve businesses and lives.

A. The Asserted Zero Trust Network Security Patent

2. K.Mizra is the owner by assignment of United States Patent No. 8,234,705 ("the '705 Patent") ("the Asserted Zero Trust Network Security Patent"). The Patent wase involved in unsuccessful *Inter Partes* Review proceeding ("IPRs"), several now-resolved federal court litigations, and was originally invented by two highly-respected and prolific inventors—James A. Roskind and Aaron T. Emigh.

3. The Asserted Zero Trust Network Security Patent was originally owned by Dr. Roskind's and Mr. Emigh's company, Radix Labs, LLC. Dr. Roskind and Mr. Emigh were then, and remain today, focused on innovation, conducting new research, developing new technologies, and creating new and innovative computer products and systems.

4. Dr. Roskind, a co-inventor of these Patents, has bachelors, masters, and doctorate degrees from MIT in both electrical engineering and computer science and is a named inventor of

over 300 U.S. patents. He has worked for Netscape as the Chief Architect and as the Netcenter Security Architect and was a co-founder of Infoseek, a company acquired by Disney for $770 million. Dr. Roskind also was a key developer of Google's "transport protocol" that provides the tech giant billions of dollars in value every year.

5. Mr. Emigh, also a co-inventor of the Asserted Zero Trust Network Security Patents, graduated from the University of California, Santa Cruz with degrees in linguistics and computer and information sciences and is a named inventor of over 140 patents. Before working with Dr. Roskind, Mr. Emigh worked in various software-development positions, including software manager, architect, and engineer for Unicom and manager for the software development and technical marketing groups of Philips TriMedia. He has founded or co-founded many companies in addition to Radix Labs, LLC, including CommerceFlow, Inc., which was acquired by eBay for the technology that Mr. Emigh helped develop.

6. After the Asserted Zero Trust Network Security Patent was issued, Dr. Roskind and Mr. Emigh recouped their research and development investment by selling certain technology rights and continued to work independently on their individual technological pursuits. K.Mizra ultimately acquired the Asserted Zero Trust Network Security Patent and licensed it to many major companies operating in the computer technology space. Some of those companies, including previously accused infringers, chose to test the validity of the Asserted Zero Trust Network Security Patent before accepting a license to the Patent.

7. For example, accused infringers of the Asserted Zero Trust Network Security Patent previously sought IPRs by the Patent Trial and Appeal Board ("PTAB") of both of the Asserted Zero Trust Network Security Patents. A Final Written Decision ("Decision") issued in the IPR for the '705 Patent concluded that the petitioners had not shown by a preponderance of the

evidence that the claims at issue were unpatentable. The '705 Patent IPR Decision was appealed to the U.S. Court of Appeals for the Federal Circuit ("CAFC"), resulting in a procedurally-focused remand back to the PTAB. The PTAB recently dismissed the IPR Petition with prejudice.

8. The Asserted Zero Trust Network Security Patent was previously asserted by K.Mizra in this District against Cisco Systems, Inc. in *K.Mizra v. Cisco Systems, Inc*. (Civil Action No. 6:20-cv-01031-ADA), a case assigned to Judge Albright which resolved in September 2024, only a few weeks prior to trial being scheduled to commence. In that action, Judge Albright not only conducted a claim construction hearing and issued several important constructions of disputed claim terms, he also ruled on numerous summary judgment and *Daubert* motions as part of a final pretrial conference. As such, Judge Albright is well-versed in the technology involved with the Asserted Zero Trust Network Security Patent, K.Mizra and other issues here likely to be involved.

B. The Asserted Hyperparameter Tuning Patent

9. K.Mizra is also the owner by assignment of United States Patent No. 8,438,120 ("the '120 Patent" or "the Asserted Hyperparameter Tuning Patent"). The '120 Patent has not previously been asserted or involved in any post issuance proceedings and was invented by Stephan Alexander Raaijmakers of the Nederlandse Organisatie Voor Toegepast-Natuurwetenschappelijk Onderzoek TNO, a well-known and widely respected technology research institution (“TNO”).

C. The Asserted Mesh Networking Patent

10. K.Mizra is also the owner by assignment of United States Patent No. 8,144,717 ("the '717 Patent" or "the Mesh Networking Patent"). The '717 Patent has also not previously been asserted or involved in any post issuance proceedings and was invented by Marinus Johannes Blange and Miodrag Djurica of the TNO.

D. K.Mizra's Licensing Model

11. The '705 Patent, '120 Patent and '717 Patents are collectively referred herein as "the Asserted Patents". K.Mizra remains ready, willing, and able to provide commercially reasonable licenses for its various patented technologies, including its Asserted Patents, to all entities who wish or need to use its covered technologies internally or in connection with products or services offered to others. As outlined below, Google is one such entity.

II. THE PARTIES

12. K.Mizra is a Delaware limited liability company with a mailing address of 777 Brickell Avenue, #500-96031, Miami, Florida 33131, and operates in Florida.

13. K.Mizra is the owner by assignment of the Asserted Patents.

14. Upon information and belief, Google is a corporation organized and existing under the laws of the state of Delaware, with a principal place of business at 1600 Amphitheatre Parkway, Mountain View, California 94043.

15. Upon information and belief, Google maintains an office in this District at 500 West Second Street, Austin, Texas 78701.

16. Upon information and belief, Google employs hundreds of people in its Austin, Texas office, including persons with knowledge of the Google products and technology at issue in this case.

17. Upon information and belief, Google is also registered to do business in Texas and may be served with process by serving Corporation Service Company, Google's registered agent in Texas located at 211 East Seventh Street, Suite 620, Austin, Texas 78701-3218.

III. JURISDICTION AND VENUE

18. This is an action for patent infringement under the patent laws of the United States, 35 U.S.C. §§ 1 *et seq.*, including 35 U.S.C. §§ 271, 281, and 284, among others. The Court has subject-matter jurisdiction over the claims raised in this action pursuant to 28 U.S.C. §§ 1331 and 1338(a).

19. This Court has personal jurisdiction over Google by virtue of, *inter alia*, its conduct of business in this District, its registration to do business in Texas, its appointment of a registered agent in Texas, its employment of hundreds of persons in its Austin office located in this District, and its substantial, continuous, and systematic contacts with the state of Texas and this District. Google intentionally markets and sells its infringing products directly and through agents to residents of Texas, enjoys substantial income from its business activities in the state of Texas and this District, and/or directly, by its own actions, and/or in combination with actions of customers and others under its control, has committed acts of patent infringement in this District at least by selling infringing products in this District.

IV. GENERAL ALLEGATIONS

A. The Zero Trust Network Security Patent and Google's Infringement

20. K.Mizra is the sole owner by assignment of the Asserted Zero Trust Network Security Patent with the full and exclusive right to enforce the Patent. K.Mizra is also entitled to sue to collect damages for all past infringement of the Asserted Zero Trust Network Security Patent.

21. The '705 Patent, titled "Contagion Isolation and Inoculation," was issued by the USPTO to Inventors Roskind and Emigh on July 31, 2012. A true and correct copy of the '705 Patent is attached hereto as Exhibit 1 and incorporated by reference.

22. The Asserted Zero Trust Network Security Patent claims priority to U.S. Provisional Application No. 60/613,909, filed on September 27, 2004 (the "Provisional Application") and is entitled to that priority date.

1. Prior Licensing and Litigation Concerning the Asserted Zero Trust Network Security Patent

23. The Asserted Zero Trust Network Security Patent has been owned by several entities, in addition to Radix and K.Mizra, with some of those entities issuing to third parties certain rights to the technologies covered thereby.

24. K.Mizra has been involved in several actions it was required to institute to protect its patent rights, including actions involving the Asserted Zero Trust Network Security Patent. Most of those actions resulted in the execution of confidential patent license agreements.

25. Google is not and has never been a licensee of the Asserted Zero Trust Network Security Patent nor had any rights to use technologies covered by the Patent. Google thus has no right (and is entitled to no right) to use technology covered by the Asserted Zero Trust Network Security Patent.

2. Computer Network Security Problems In 2004 Solved by the Asserted Zero Trust Network Security Patents

26. The technology described in the Asserted Zero Trust Network Security Patent was invented by Dr. Roskind and Mr. Emigh, two colleagues living in the same area who had similar interests in innovating computer-related technologies. In 2003, the inventors decided to create a business—Radix Labs, LLC—which focused on developing intellectual property related to various computer technologies, including computer network security technologies. The inventors focused on conceiving and reducing to practice inventions that they knew were needed (or soon

would be needed) in the computer networking industry and then on drafting patent applications to capture and protect their technological innovations.

27. In September of 2004, the inventors filed the Provisional Application to which the Asserted Zero Trust Network Security Patent claims priority. The Provisional Application described technology that focused on securing a computer network against the threats to which it was exposed when computer endpoints (e.g., laptop computers, smart phones, tablets, etc.,) were connected to a computer network. The Provisional Application, and by natural extension the Asserted Zero Trust Network Security Patent, also focus on remedying and/or quarantining those threats to mitigate any damage to the secured network.

28. Claims of the Asserted Zero Trust Network Security Patent are directed to technological solutions that address specific challenges grounded in computer network security. Maintaining the security of computer systems and networks is a tremendous concern for modern enterprises, since a breach of an internal network can have severe repercussions, including major financial losses, data theft, disclosure of sensitive information, network disruptions, data corruption, etc. The inventors of the Asserted Zero Trust Network Security Patent understood that while a network security appliance or hardware can be adept at keeping out unwanted external intrusions from the network, the most exploitable vulnerabilities of most networks are the end-user computers that roam throughout various public and private network domains, potentially exposing those computers to infection and then accessing and potentially infecting the entire and presumably secure computer network.

29. For example, the '705 Patent explains that

> Laptop and wireless computers and other mobile systems pose a threat to elements comprising and/or connected to a network service provider, enterprise, or other protected network to which they reconnect after a period of connection to one or more networks and/or systems that are not part of the service provider, enterprise,

> or other protected network. By roaming to unknown domains, such as the Internet, and/or connecting to such domains through public, wireless, and/or otherwise less secure access nodes, such mobile systems may become infected by computer viruses, worms, backdoors, and/or countless other threats and/or exploits and/or have unauthorized software installed; have software installed on the mobile system by an operator of the protected network for the protection of the mobile system and/or the protected network removed or altered without authorization and/or have configurations, settings, security data, and/or other data added, removed, and/or changed in authorized ways and/or by unauthorized person.

See, e.g., Ex. 1 at 1:14–31.

30. The solution to these problems—as specified and claimed in the Asserted Zero Trust Network Security Patent—was an advanced departure from the conventional network access control solutions then in use and was then, as it remains today, patent eligible, highly valuable, novel, and non-obvious technology.

3. K.Mizra's Asserted Zero Trust Network Security Patent Claims Are Presumed Valid

31. K.Mizra asserts that Claims 12, 13, 16 and 19 of the '705 Patent have been directly infringed, either literally or under the doctrine of equivalents by Google. These claims are referred to herein as the "Asserted Zero Trust Network Security Claims."

32. None of the Asserted Zero Trust Network Security Claims are directed to abstract ideas, and each employs inventive concepts and is directed to patent-eligible subject matter. All claims of the Asserted Zero Trust Network Security Patent are also presumed to be valid and enforceable against Google and others.

33. Indeed, the Asserted Zero Trust Network Security Patent's specification and claims demonstrate that the need satisfied by the inventions of the Asserted Zero Trust Network Security Claims was long-felt in the industry and thus unconventional. As one example, the '705 Patent explains that mobile end user devices such as laptops and wireless computers pose a threat to protected network elements because those devices may access unsecure systems and thereby

"become infected by computer viruses, worms, backdoors, and/or countless other threats and/or exploits and/or have unauthorized software installed; have software installed on the mobile system by an operator of the protected network for the protection of the mobile system and/or the protected network removed or altered without authorization; and/or have configurations, settings, security data, and/or other data added, removed, and/or changed in unauthorized ways and/or by unauthorized person[s]." Ex. 1 at 1:23-31. Similarly, stationary systems such as desktop computers "may become infected, e.g., due to receipt and execution of malicious code via a network or other communication and/or a diskette and/or other removable media." Ex. 1 at 1:31-34. This poses a danger to a protected network because, when the user device connects to the protected network, "a system may infect or otherwise harm resources associated with the protected network before measures can be taken to detect and prevent the spread of such infections or harm. Ex. 1 at 1:34-38. The Asserted Zero Trust Network Security Patent's specification further provides that "[t]herefore, there is a need for a reliable way to ensure that a system does not infect or otherwise harm other network resources when connected to a protected network." *Id.* at 1:38-41.

34. The specification (including the provisions quoted above), the figures (including those included below), and the text related to the figures further illustrate the complex, tiered network system architecture of the inventions captured by the Asserted Zero Trust Network Security Claims. These figures include the following:

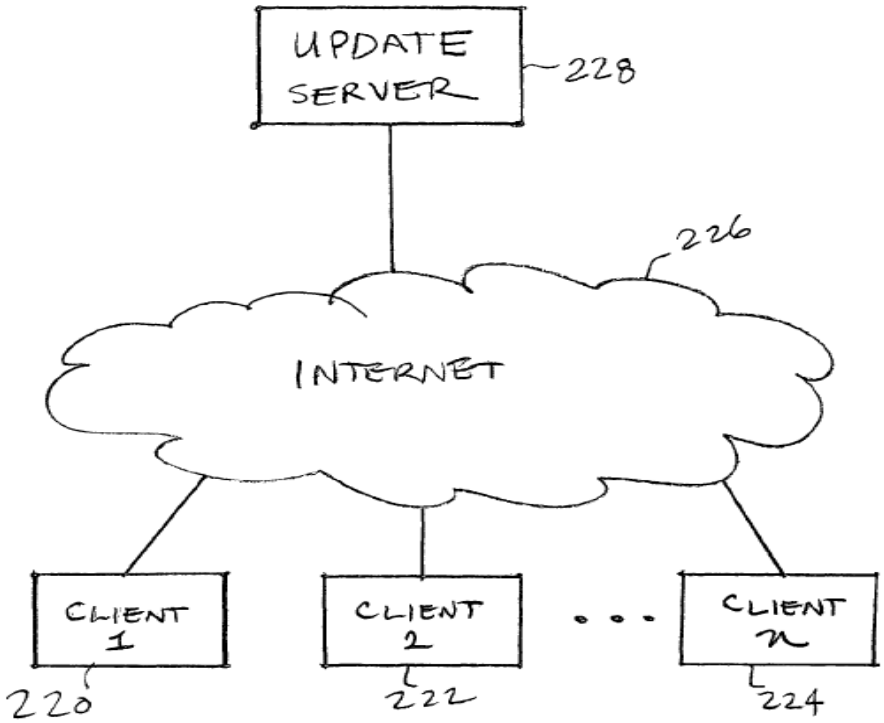
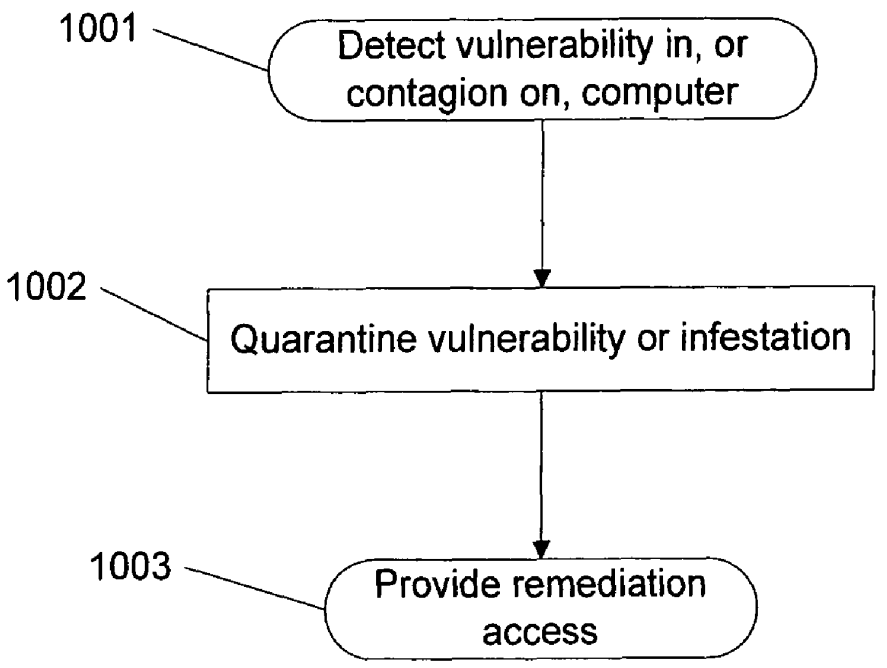


FIG. 2B

See Ex. 1 at Fig. 2B.



See id. at Fig. 10A.

35. The Asserted Zero Trust Network Security Claims focus on being able to receive, understand and act upon information provided by specific tamperproof hardware that must interact with unique software to improve network access control technology and protect a secure computer network and the data stored thereon from infected devices. As such, the Asserted Zero Trust Network Security Claims are eligible as a matter of law for patent protection under step one of *Alice Corp. v. CLS Bank Int'l*, 573 U.S. 208, 216 (2014).

36. All actions and steps recited in the Asserted Zero Trust Network Security Claims, including the act of quarantining endpoints or other computers, if necessary, require the involvement of various hardware components running dedicated software both before, during, and after the selection and isolation of an endpoint device. Said another way, claims directed to allowing a machine to automatically and dynamically select and isolate an unsafe device

attempting to access a secure network is not simply adding a generic computer component to a fundamentally human process. Rather, it is removing the once-necessary human intervention from a fundamentally mechanical process, an "improvement in the functioning of a" networked system that simply cannot be considered directed to an abstract concept. *Enfish LLC v. Microsoft Corp.*, 822 F.3d 1327, 1339 (Fed. Cir. 2016). These technological acts were unconventional and unknown in the art existing in 2004.

37. As the specification confirms, the improvement captured by the Asserted Zero Trust Network Security Claims is not simply quarantining an infected device, but it is instead a multi-faceted network system involving multiple interrelated software and hardware components to protect a network from known and unknown threats. Specifically, the Asserted Zero Trust Network Security Patent discloses that to reduce the burdens of having to manually identify, connect to, isolate, and remove malicious software from an infected endpoint, the networked system can direct an unclean endpoint (referred to in the Patent as a "host computer") attempting to connect to the secure network to a form of remediation, such as downloading a software patch or a software update, removing material from, enabling certain settings, etc. present on the endpoint. *See* Ex. 1 at 1:14–41.

38. Indeed, the inventions of the Asserted Zero Trust Network Security Claims are each tethered to these advances, none of which, individually, and certainly not as an ordered combination, was conventional in the computer network security field in 2004. Indeed, these claims recite then-unconventional systems and technology that automatically and dynamically detect an insecure condition by contacting a trusted computing base located on an endpoint, receiving a response therefrom, determining if that response contains a valid identification of cleanliness, and configuring and implementing any needed remediation action upon. *See, e.g.,* Ex.

1, Claim 19. More specifically, the Asserted Zero Trust Network Security Claims require a system to communicate with a "trusted computing base" to determine when a response includes a valid digitally signed attestation of cleanliness, and to control access to the network accordingly. These Asserted Zero Trust Network Security Claims are thus directed to a machine-implemented solution resolving a machine-specific problem, *i.e.* a machine's difficulty in detecting, isolating, and remediating infected endpoint devices (*e.g.*, host computers) to prevent contagion of and damage to the larger computer network. These technological acts were completely unconventional in the relevant field in 2004.

4. Failed IPRs Brought Against the Asserted Zero Trust Network Security Patents

39. Fortune 100 companies accused of infringing the Asserted Zero Trust Network Security Patent previously filed petitions for IPRs against it, alleging that the claims thereof should be held invalid as either anticipated or obvious considering art not previously considered. Ultimately, the PTAB instituted an IPR against the '705 Patent, with similar third party IPRs that were subsequently filed being joined to the first-filed and instituted IPR.

40. The PTAB eventually issued its decision holding that no claims of the '705 Patent were unpatentable, finding that no asserted prior art reference alone or in combination satisfied the limitation of "providing . . . an IP address of a quarantine server configured to serve the quarantine notification page" that was present in all claims of the '705 Patent.

41. The '705 Patent IPR Decision was then appealed to the CAFC, which reversed the PTAB's Decision on a few narrow procedural issues involving proof that the asserted prior art references would be combined by a person having ordinary skill in the art, as alleged by the petitioners.

42. The IPRs have now all dismissed with prejudice by the PTAB.

5. Google's Infringing Chrome Enterprise Premium Product

43. Google has been and continues to make, sell, use, offer to sell, import into and/or export out of the United States a product known as Chrome Enterprise Premium ("CEP"), a computer network access and security product. These actions directly infringe the Asserted Zero Trust Network Security Patent in violation of 35 U.S.C. § 271(a). (Attached as Exhibit 2 and incorporated herein by this reference, is a preliminary claim chart explaining Google's infringing conduct.)

44. CEP is a secure enterprise solution that provides integrated threat and data protection and offers scalable context-aware access controls for software-as-a-service (SaaS) and Web Apps, helping to mitigate data exfiltration from secure networks. CEP achieves these goals by being designed to operate within an environment of endpoint that include a trusted computing base ("TCB") with a trusted platform module ("TPM") and a Chrome browser. When such a configured endpoint seeks to enter a protected network, the CEP running on Google's cloud obtains posture information stored in the TCB's TPM of the endpoint. CEP, which integrates several security features and functions that Google offers in CEP and in other products, such as Endpoint Verification, Context-Aware Access, certificate-based access, Verified Access, Policy Remediation, and others, then uses the provided information to make decisions about whether the requesting endpoint should be granted access to the protected network, if it should be remediated, quarantined, etc., and then sends to the endpoint appropriate information, etc.

45. As Google has explained it:

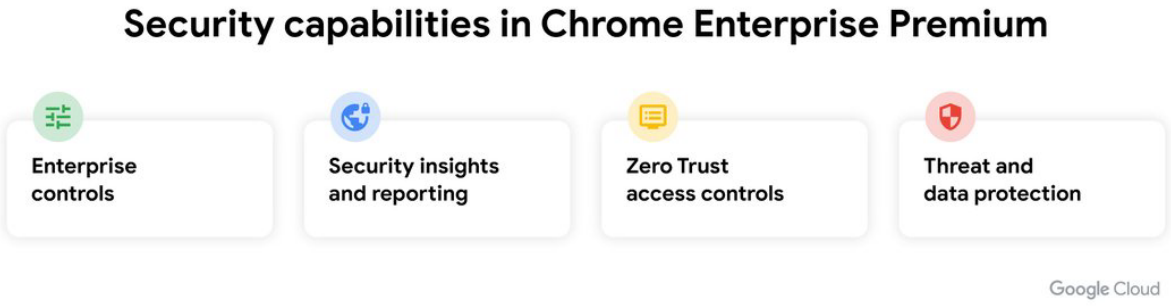
> Browsers are more than just a portal to the Internet: They are the new endpoint where almost every high-value activity and interaction in the enterprise takes place. Authentication, access, communication and collaboration, administration, and even coding are all browser-based activities in the modern enterprise.

Endpoint security is growing more challenging due to remote work, reliance on an extended workforce, and the proliferation of new devices that aren't part of an organization's managed fleet. As these trends continue to accelerate and converge, it's clear that the browser is a natural enforcement point for endpoint security in the modern enterprise. Gartner® research predicts "enterprise browsers will be the core platform for delivering workforce productivity and security software on managed and unmanaged devices for a seamless hybrid work experience."

Get advanced security with Chrome Enterprise Premium

Users around the world already use Chrome on their digital devices to access information and get work done safely across Windows, macOS, ChromeOS, Android, and iOS. As a leader in browser security, Chrome has built features and best practices that have disrupted, and then established, industry norms.

For example, we introduced advanced security sandboxing and novel exploit mitigations, and we lead the industry in our response time to zero-day vulnerabilities. We also continue to advance security, and recently announced work that uses AI to do privacy-preserving, real-time checks of websites to help protect users from increasingly sophisticated attacks. Chrome Enterprise Premium builds on the core capabilities available in Chrome Enterprise, which gives organizations the secure, reliable browser their employees prefer to use with management tools for IT and security teams built in.



Chrome Enterprise Premium offers additional advanced security capabilities, including:

- **Enterprise controls** enforce policies, manage software updates and extensions to align with enterprise policies, and support RDP, SCP, SSH and other TCP protocols;
- **Security insights and reporting** support event reporting, device reporting, and forensic capabilities for enterprise-wide visibility, and can integrate with other Google and third-party security solutions;
- **Context-aware access controls** can be scaled for web applications, can help enforce continuous Zero Trust access to SaaS and web-based apps with context-aware access control, and can mitigate data exfiltration risks for sanctioned and unsanctioned applications; and
- **Threat and data protection** delivers content inspection and data loss prevention, anti-malware, and anti-phishing using frontline intelligence and AI, dynamic URL filtering, and site categorization.

Organizations using Chrome Enterprise Premium benefit from Google's threat intelligence, security features, and Zero Trust access via Google's secure global network. *See* https://cloud.google.com/blog/products/identity-security/introducing-chrome-enterprise-premium, attached as Exhibit 3 and incorporated herein by this reference.

46. CEP is based on Google's "BeyondCorp" product. *See, e.g.*, https://cloud.google.com/beyondcorp-enterprise/docs/access-protection#:~:text=Based%20on%20the%20BeyondCorp%20security,SSH%20access%20to%20virtual%20machines, attached as Exhibit 4 and incorporated herein by this reference; *see also* https://cloud.google.com/beyondcorp?hl=en, attached as Exhibit 5 and incorporated herein by this reference. Google refers to BeyondCorp in some CEP-related documents and website pages. BeyondCorp currently is not an accused product, but K.Mizra reserves the right to allege infringement by BeyondCorp if discovery dictates and provides support for such an allegation.

B. The Asserted Hyperparameter Tuning Patent

47. K.Mizra is the sole owner by assignment of the Asserted Hyperparameter Tuning Patent with the full and exclusive right to bring suit to enforce the Patent. K.Mizra is also entitled to sue to collect damages for all past infringement of the Asserted Hyperparameter Tuning Patent.

48. The '120 Patent, titled "Machine Learning Hyperparameter Estimation," was issued by the USPTO to Inventor Stephen Alexander Raaijmakers on May 7, 2013. A true and correct copy of the '120 Patent is attached hereto as Exhibit 6 and incorporated by reference.

49. The Asserted Hyperparameter Tuning Patent is a U.S. National Phase filing of PCT/NL2008/050247 and claims priority to European Application Nos. 07106963.7 filed April 25, 2007 and 07112037.2 filed July 9, 2007.

1. Prior Licensing and Litigation Concerning the Hyperparameter Tuning Patent

50. The Asserted Hyperparameter Tuning Patent has previously been licensed to various industry participants but has not been involved in any prior litigation.

51. Google is not and has never been a licensee of the Asserted Hyperparameter Tuning Patent nor has it had nor does it have any rights to use technologies covered by the Patent. Google thus has no ownership or other rights (and is entitled to no rights) relating to the Asserted Hyperparameter Tuning Patent.

52. No party licensed to the Asserted Hyperparameter Tuning Patent has used the technology claimed by the Asserted Hyperparameter Tuning Patent in a manner that requires marking under 35 U.S.C. § 287. K.Mizra is therefore not barred from seeking past damages for infringement of the Asserted Hyperparameter Tuning Patent under 35 U.S.C. § 287(a).

2. Problems in 2007 Solved By the Asserted Hyperparameter Tuning Patent

53. The technology described in the Asserted Hyperparameter Tuning Patent was invented by Stephen Raaijmakers and generally relates to improving the operation of Artificial Intelligence ("AI") through the improved automatic, systematic and computer implemented training techniques. More specifically, AI typically uses models to perform specific tasks, and, at its most basic level, an AI model is a program that has been "trained" on a set of data to recognize certain patterns or to make certain decisions without further human intervention. For example, an AI model could be used to automatically detect spam or other unwanted emails in a user's email inbox or distinguish between ripe and unripe fruit to be picked, etc. The '120 Patent uses the term "classifier" to refer to these types of AI models, as they are used to separate inputs (emails or fruit) into different classifications (spam or not spam/ripe or unripe).

54. AI models use variables called "parameters" to determine what result it should produce (e.g., to determine whether to classify an email as spam or not spam). These parameters are not set manually. Instead, they are learned automatically during the training process. During training, the model may be exposed to a labeled dataset and the parameters can be adjusted to minimize prediction errors. For instance, in the spam email example, the AI model may be trained by exposing the model to a large number of emails that have been labeled as either spam or not spam. The model identifies and thus learns the traits and characteristics found in these various emails that are indicative of a spam email or not spam email. Automatically adjusting the parameters of the model to minimize the times when a spam email is marked as not spam, or vice versa, is the goal of parameter-based training.

55. AI models are generally trained for a specific purpose. However, training a full model is time consuming and, in some cases, has given way to the use of foundational models,

which are models that are pre-trained on large, genericized datasets. These foundational models can later be fine-tuned for use in addressing a particular problem or issue by, for example, feeding it a set of additional relevant data and training upon that dataset. By way of example, a foundational AI model for diagnosing medical conditions could be fine-tuned by training it using various medical records. Similarly, a foundational large language model ("LLM") like ChatGPT could be fine-tuned to adjust its conversational tone by training it on sample texts representing a desired communication style. Using a foundational AI model is like getting a shirt and later tailoring it to fit your body, rather than starting with fabric, thread, and a needle.

56. As AI models become more advanced and versatile, they also demand more data and computing power to be properly trained. One of the main challenges in this exploding field of technology is efficiently training these models, which typically is done by configuring variables in a model's learning process. These variables are known as hyperparameters. The hyperparameters may, for example, define the architecture (and thus the size) of the model, as well as, how the model changes based on the training data it digests and that will effect how efficiently and effectively the model is trained. For example, a hyperparameter called the learning rate may be used to control how much a model adjusts its parameters during training. If the learning rate of the model is too low, it will take a long time for the model to reach an acceptable result. On the other hand, if the learning rate is too high, the model may overreact to each training round, bouncing around an optimum result instead of converging on that result. Different hyperparameters may be used depending on whether a user is training a brand-new model or fine-tuning a foundational model. In either case, though, identifying the right hyperparameters and their values during the model training process improves the model's efficiency and thus the predictions or decisions to be output by the model.

57. Claims of the Asserted Hyperparameter Tuning Patent are directed to technological solutions that address specific challenges grounded in the systematic and automatic tuning of hyperparameters so that efficient and effective AI models can be computer generated. Creating these models is a tremendous concern for modern enterprises, since they are now being used in all manner of human endeavor, from drug discovery, to providing banking efficiencies, etc.

58. The inventor of the Asserted Hyperparameter Tuning Patent understood that while basic models that trained on parameters existed prior to 2007, as the data to be analyzed become more and more voluminous and complex, the use of just parameters in developing a model would not be enough. Rather, the additional use of hyperparameters, that could automatically and systematically be optimized by the computing system that was training the model was needed. The inventor also realized that the existing methods for estimating and selecting hyperparameters were unsuitable in many cases. For example, the '120 Patent discloses:

> Hyperparameters may be determined using various methods, for example heuristic or statistical optimization methods. However, many heuristic methods are ad hoc methods, while not all statistical optimization methods are suitable for determining hyperparameters.

The inventor thus recognized that a new paradigm needed to be developed. That paradigm was then well before its time invented and is disclosed in the '120 Patent.

59. For example, the '120 Patent explains that a method of determining hyperparameters in a machine learning system is disclosed. The method is iterative, and with each computer implemented iteration, a random or pseudorandom sample of hyperparameter vectors selected from a set of possible hyperparameter vectors must be considered, updating the estimate of the target hyperparameter vector by selecting for use the best vector in the present and previous iterations. The '120 Patent specification further explains that:

> By selecting from the random sample a or the hyperparameter vector producing the best result in the present and any previous iterations, and using said

selected hyperparameter vector to update the estimate of the target hyperparameter vector, the properties of the best performing hyperparameters are used to guide the next iteration [of the hyperparameter selection process]. In this way, a method similar to the cross-entropy method [used for selection of parameters] can be used effectively for determining hyperparameters, even when the hyperparameters are not continuous and their effects are not transparent. By using the method according to the present invention, an improved classification [i.e. model] accuracy is achieved.

…

In a preferred embodiment, the method according to the Present invention further comprises the steps of:

selecting from the random sample a further hyperparameter vector producing the best result in any previous iterations, and

restricting the random sample of hyperparameter vectors by using the further selected hyperparameter vector.

The step of restricting the random sample preferably involves the use of an interval surrounding the further selected hyperparameter vector which produces the best result in any previous iteration.

…

In a preferred embodiment, E^t is the hyperparameter vector X^t, at iteration t producing the best result $S(X^t_i)$ in the present iteration t and all previous iterations (if any). The random sample X^t_i, has elements X^t_{ij}, while the step of updating the hyperparameters comprises the step of determining the (target) hyperparameter v^t_j, where

$$v^t_j = \frac{\sum_{i=1}^{n} I\{S(X^t_i) \geq \gamma^t\} W(X^t_i; E^t) X^t_{ij}}{\sum_{i=1}^{n} I\{S(X^t_i) \geq \gamma^t\} W(X^t_i; E^t)},$$

wherein γ^t is a threshold value, W is a weighting function and $I\{S(X_i^t) \geqq \gamma^t\}$ is an indicator function which is equal to 1 if $S(X_i^t)$ greater than or equal to the threshold value γ^t, and else is equal to 0. The weighting function W is preferably given by

$$W(X^t_i; E^t) = 1 - \frac{\sqrt{\sum_{j=1}^{m} (X^t_{ij} - E^t_j)^2}}{\sqrt{\sum_{j=1}^{m} (X^t_{ij})^2} \sqrt{\sum_{j=1}^{m} (E^t_j)^2}}.$$

Accordingly, the step of updating the hyperparameters may involve a weighting function based upon a distance function, preferably a Euclidean distance function.

> It is noted that when the Euclidean distance between E^t and X^t_{ij}, is zero, the weighting function W is equal to one.
>
> The method according to the present invention is typically carried out by computer apparatus, for example a general purpose computer system comprising a processor and an associated memory, one part of the memory storing a soft ware program for instructing the processor to carry out the method steps of the present invention, and another part of the memory storing data, said data comprising the hyperparameter values referred to above.

See, e.g., Ex. 6 at 2:1–4:14.

60. The solution to the problems identified in the '120 Patent—as specified and claimed in the Asserted Hyperparameter Tuning Patent—was an advanced departure from the conventional tuning of hyperparameters in AI models in use in 2007 and was then, as it remains today, patent-eligible, highly valuable, novel, and non-obvious technology.

3. K.Mizra's Asserted Hyperparameter Tuning Patent Claims Are Presumed Valid

61. None of the Asserted Hyperparameter Tuning Patent Claims are directed to abstract ideas, and each employs inventive concepts and is directed to patent-eligible subject matter. All claims of the Asserted Hyperparameter Tuning Patent are also presumed to be valid and enforceable against Google and others.

62. The subject matter of the Asserted Hyperparameter Tuning Patent is focused on problems in machine learning systems. *See, e.g.,* Ex. 6 at 1:6-9, 1:13-23, 2:1-19, 3:46-53, 4:9-13. Indeed, hyperparameters are used by machine learning systems to determine the parameters for a classifier in a machine learning system. In other words, the solution described by the claims of the Asserted Hyperparameter Tuning Patent is necessarily rooted in computer technology.

63. Moreover, selecting the appropriate hyperparameters is vital to the speed and efficiency of the training computer and the machine learning system. Selecting inappropriate hyperparameters can lead to myriad problems in training the AI model, including underfitting or

overfitting, which in turn could lead to a training system that is inefficient, takes too long to determine the parameters, and/or consumes massive amounts of power; or a resulting machine learning system with a poorly-trained classifier (because the model never converged) which thus misclassifies a significant amount of data.

64. The Asserted Hyperparameter Tuning Patent Claims are directed to a machine-implemented, iterative process for locating, selecting and utilizing in a model an optimized hyperparameter. As such, the Asserted Hyperparameter Tuning Patent Claims recite inventions with specific applications or improvements to technologies in the marketplace and cannot be considered abstract or patent ineligible under relevant law.

4. Google's Infringing Vertex AI Studio Product

65. Google's Vertex AI Studio is part of the Google Cloud Platform and is a complete system for building, training, and using AI models. It offers a range of tools and APIs (Application Programming Interfaces) that let users either create and train their own models from scratch, or work with Google's foundation models, like Gemini, Imagen, Codey, and Chirp, and tailor them for specific needs. As is here relevant, Vertex AI provides several options for training all manner of models. One of the most powerful of these provides a user with complete control over the model development process by allowing them access to powerful hyperparameter tuning and optimization options. These can be implemented using Google's Vertex AI Vizier. It is Google's provision of this computer implemented hyperparameter tuning ability that takes advantage of iterative, randomized/pseudo-randomized hyperparameter tuning techniques, that makes Google's Vertex AI so valuable to its customers. Google's usurpation of the critical '120 Patent's patented technology without K.Mizra's authorization allows Google to provide to its customers highly tuned

and effective AI models, generating significant revenues and reputational prestige for Google, all to the injury of K.Mizra.

66. Google has been making, selling, using and offering for sale AI products and services that infringe the Asserted Hyperparameter Tuning Patent in violation of 35 U.S.C. § 271 (collectively, "the Accused Hyperparameter Tuning Instrumentalities"). These Accused Hyperparameter Tuning Instrumentalities include, but are not limited to, Google's Vertex AI system, the sale, offer for sale, use, and/or manufacture in the United States of which constitutes infringement of at least claim 1 of the Accused Hyperparameter Tuning Patent, either literally or under the doctrine of equivalents.

C. The Asserted Mesh Networking Patent

67. K.Mizra is the sole owner by assignment of the Asserted Mesh Networking Patent with the full and exclusive right to enforce the Patent. K.Mizra is also entitled to sue to collect damages for all past infringement of the Asserted Mesh Networking Patent.

68. The '717 Patent, titled "Initialization of a Wireless Communication Network," was issued by the USPTO to Inventors Blange and Djurica on March 27, 2012. A true and correct copy of the '717 Patent is attached hereto as Exhibit 7 and incorporated by reference.

69. The Asserted Mesh Networking Patent is a U.S. National Phase filing of PCT/NL2007/0000001, filed on December 2, 2007, and is entitled to that priority date. The Asserted Mesh Networking Patent further claims priority to European Patent Application No. EP 05078044, filed December 30, 2005, and is entitled to that priority date.

1. Prior Licensing and Litigation Concerning the Mesh Networking Patent

70. The Mesh Networking Patent has previously been licensed to various industry participants but has not been involved in any prior litigation.

71. Google is not and has never been a licensee of the Asserted Mesh Networking Patent nor has it had nor does it have any rights to use technologies covered by the Patent. Google thus has no ownership or other rights (and is entitled to no rights) relating to the Asserted Mesh Networking Patent.

72. No party licensed to the Asserted Mesh Networking Patent has used the technology claimed by the Asserted Mesh Networking Patent in a manner that requires marking under 35 U.S.C. § 287. K.Mizra is therefore not barred from seeking past damages for infringement of the Asserted Mesh Networking Patent under 35 U.S.C. § 287(a).

2. Problems in 2007 Solved By the Mesh Networking Patent

73. The technology described in the Mesh Patent was invented by Marinus Johannes Blange and Miodrag Djurica and generally relates to creation of a flexible mesh wireless network.

74. Although wireless communications networks were known in the art prior to the invention, the known networks had several drawbacks. For example, there existed known techniques for routing messages in wireless networks. *See generally* Ex. 7 at 1:13-58. However, as explained in the '717 Patent, these techniques "require[d] transmission of information via sensor stations before the routing topology [of the network] has been defined." *Id.* at 1:59-62. "This complicates the (sensor) stations, because they will have to support different modes of communication while operating in a low-power mode." *Id.* at 1:62-64. As further explained in the '717 Patent, the existing techniques also resulted in either slow or inefficient networks:

> If forwarding of information is only done using the definite routing topology, a central approach is slow (each station has to wait until its upward station has been centrally assigned a route) and a decentral approach is inefficient (a routing topology with an unnecessary amount of branches has been created).

Id. at 1:64-2:2.

75. Another known protocol was a mobile communications protocol which uses two types of networks: a high power network and a low power network. *See generally* Ex. 7 at 2:9-57. However, the '717 Patent notes that "[t]he use of two types of network without central unit makes this system complex and inefficient in bandwidth use." *Id.* at 3:1-2.

76. The solution to these problems—as specified and claimed in the Asserted Mesh Networking Patent—was an advanced departure from the conventional networking solutions then in use and was then, as it remains today, patent eligible, highly valuable, novel, and non-obvious technology.

3. K.Mizra's Asserted Mesh Networking Patent Claims Are Presumed Valid

77. None of the Asserted Mesh Networking Patent Claims are directed to abstract ideas, and each employs inventive concepts and is directed to patent-eligible subject matter. All claims of the Asserted Mesh Networking Patent are also presumed to be valid and enforceable against Google and others.

78. The subject matter of the Asserted Mesh Networking Patent is focused on problems in wireless mesh networks, and in particular problems related to creating a new wireless mesh network and/or attaching a new station to an existing wireless mesh network. *See, e.g.,* Ex. 7 at 1:59-2:8, 2:58-3:2, 3:6-15, 3:59-67, 5:11-14. The solution described by the claims of the Asserted Mesh Networking Patent is necessarily rooted in wireless mesh networking technology and presumptively patent eligible.

79. Moreover, the techniques specified and claimed in the Asserted Mesh Networking Patent Claims provides various technical benefits to the network. For example, Claim 11 of the Asserted Mesh Networking Patent is directed to, among other things, a "processor circuit being configured to switch to establish an associated route for communication with an association unit

during subsequent communication of operating messages to an association unit, the associated route including a source of the association grant, and, if the source of the association grant is not the association unit, the operating route between the source of the association grant and the association unit previously established for the source of the association grant in response to association request messages and/or the association grants for intermediate stations." Use of the claimed technique improves the speed of the network by providing the associated route to each station that joins a network, allowing the source of the association grant to provide the route information directly rather than retrieving it from the association unit, improving the speed with which the network can heal/form itself. It further eliminates unnecessary communications between the source of the association grant and the association unit, conserving bandwidth (and thus allowing more bandwidth for other communications) and conserving power of the various stations located within the associated route. As noted in the Asserted Mesh Networking Patent, power consumption is a key consideration in low power devices often found in mesh networks. Ex. 7 at 5:11-14; *see also id.* at 11:30-42.

80. The Asserted Mesh Networking Claims are directed to wireless networks and stations within wireless networks that implement the techniques described for allowing new stations to efficiently and effectively join the network. As such, the Asserted Mesh Networking Claims recite inventions with specific applications or improvements to technologies in the marketplace and cannot be considered abstract or patent ineligible under relevant law.

4. Google's Infringing Nest Devices

81. Google Nest is a line of products for home networking, allowing a user to control various Google and third-party smart devices throughout the Google Home app. On information and belief, Google Nest was previously known as Google Home.

82. Google offers a wide variety of home networking devices under the Google Nest brand. Many of these devices use a networking protocol called Thread.

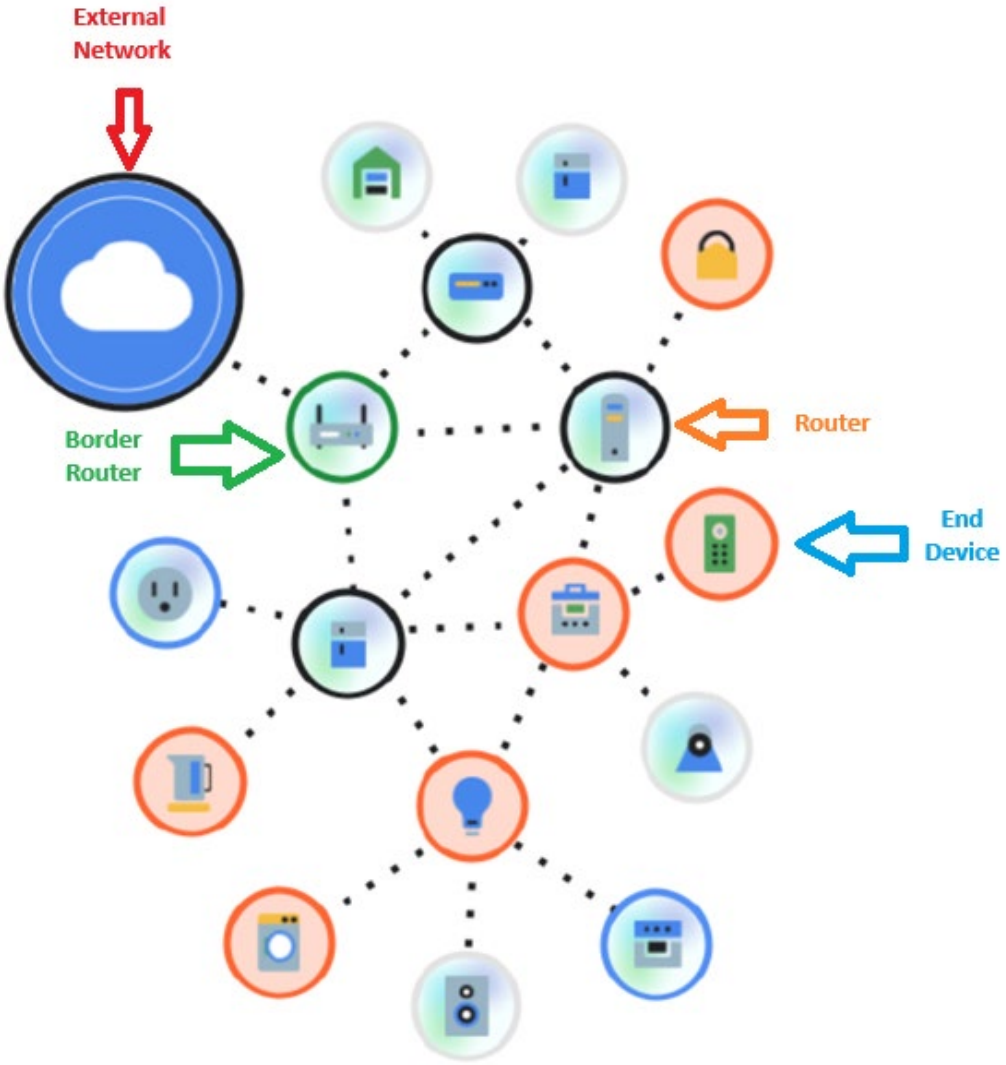
83. Thread is a networking protocol designed for low-power Internet of Things devices operating in an IEEE 802.11.15.4 wireless mesh network. It may be used alongside other networks, such as a Wi-Fi network, in a user's home. Thread was developed by the Thread group, which Google founded with other member companies in 2014.

84. Thread is built into many new smart devices such as smart bulbs, locks, sensors, and the like. Thread is used to create a mesh network in which each new smart device connects to the next nearest smart device, giving users more networking coverage and making the entire network more robust. If one device in the network fails, another can step in to maintain the signal, allowing the network to heal itself rapidly, effectively and efficiently.

85. A device in a Thread network may be either a router, which may forward data packets for other network devices and provides commissioning for devices trying to join the Thread network, or an end device, which communicates with a single router and does not forward packets for other network devices. An end device may convert itself into a router when a new device joins the network.

86. A special case of a router is a border router, which can forward information between a Thread network and a non-Thread network. For example, a border router may allow a user to control the devices on a Thread network through the Wi-Fi network via a border router.

87. An exemplary Thread network including a border router, routers, and end devices is shown below:



Ex. 8, "Thread Play Services APIs," (available at https://developers.home.google.com/matter/thread (accessed May 12, 2025) and incorporated by reference) (annotated).

88. Google offers a wide range of products that use the Thread networking protocol. For example, Google offers several hubs for Google home that include a built-in Thread router, including the Nest Hub, Nest Hub Max, Nest Wifi Pro, and Google TV Streamer.

> Google devices like the Nest WiFi, Google Nest Hub Max and Google Nest Hub (2nd gen) have Thread radios built-into them and act as Thread Border Routers.

Id.; *see also* Ex. 9, "Connect Thread smart devices with Google TV Streamer (4K)," (available at https://support.google.com/chromecast/answer/15178609 (accessed May 21, 2025) and incorporated by reference).

89. In addition, Google hub products such as the Google TV Streamer, Nest Hub (2nd Gen), Hub Nest Max Fuchsia, Nest Wi-Fi Point, and Nest Wi-Fi Router are certified as Built on Thread by the Thread Group.

90. As another example, Google offers a routing device called the Nest Connect.

Nest Connect



Ex. 10, Nest Connect – Range Extender for Detect & Guard – Google Store," (available at https://store.google.com/product/nest_connect (accessed May 13, 2025) and incorporated by reference)). The Nest Connect includes a Thread radio that allows communication between a Thread border router and a peripheral device such as the Nest x Yale Lock.

> In an emailed statement, Grant Erickson, president of the Thread Group said, "Nest Detect relies on Thread's low power architecture to last for long periods of time on one battery. Nest Connect is a Thread router, which seamlessly extends both the reach and connectivity of the Thread network. Nest Guard provides seamless IP-based routing between Thread, Wi-Fi, and cellular, and their interoperability with the Yale Linus Lock is the posterchild of Thread's benefits in action."

Ex. 11, "Nest debuts Nest Secure home system and has a Thread router called Nest Connect," (available at https://staceyoniot.com/nest-debuts-nest-secure-home-system-and-has-a-thread-router-called-nest-connect/ (accessed May 13, 2025) and incorporated by reference).

91. Yet another example, Google offers various peripheral devices that include a Thread radio such as the Nest x Yale Lock.



Ex. 12, "Nest x Yale Lock – Key-Free Smart Deadbolt – Google Store," (available at https://store.google.com/product/nest_x_yale_lock (accessed May 13, 2025) and incorporated by reference).

92. Additional peripheral devices offered by Google that include a Thread radio include the Google Nest Learning Thermostat and the Google Nest Protect and CO alarm.

93. Google has been making, selling, using and offering for sale products including Thread radios that infringe the Asserted Mesh Networking Patent in violation of 35 U.S.C. § 271

(collectively, "the Accused Mesh Networking Instrumentalities"). These Accused Mesh Networking Instrumentalities include, but are not limited to, Google's Nest home networking products identified above, the sale, offer for sale, use, and/or manufacture in the United States of which constitutes infringement of at least claim 11 of the Accused Mesh Networking Patent, either literally or under the doctrine of equivalents.

94. K.Mizra expects that discovery will reveal additional products that include Thread radios and infringe the Asserted Mesh Networking Patents, which may include additional Google Nest products and/or products that are not marketed under the Google Nest brand. For example, there are reports that at least some Google Pixel 9 devices may include a Thread radio. *See, e.g.,* Ex. 13, "Pixel 9 series arrives at the FCC, includes Thread," (available at https://9to5google.com/2024/07/12/pixel-9-fcc/ (accessed May 23, 2025) and incorporated by reference).

V. FIRST CLAIM FOR RELIEF
(Patent Infringement Under 35 U.S.C. § 271 of the '705 Patent)

95. K.Mizra incorporates paragraphs 1 through 94 as though fully set forth herein.

96. Claim 19 of the '705 Patent states:

> [preamble] A computer program product for protecting a network, the computer program product being embodied in a non-transitory computer readable medium and comprising computer instructions for:
>
> [A] detecting an insecure condition on a first host that has connected or is attempting to connect to a protected network,
>
> [B] wherein detecting the insecure condition includes:
>
> > [B1] contacting a trusted computing base associated with a trusted platform module within the first host,

> [B2] receiving a response, and determining whether the response includes a valid digitally signed attestation of cleanliness,
>
> [C] wherein the valid digitally signed attestation of cleanliness includes at least one of an attestation that the trusted computing base has ascertained that the first host is not infested, and an attestation that the trusted computing base has ascertained the presence of a patch or a patch level associated with a software component on the first host;
>
> [D] when it is determined that the response does not include a valid digitally signed attestation of cleanliness, quarantining the first host, including by preventing the first host from sending data to one or more other hosts associated with the protected network,
>
> [E] wherein preventing the first host from sending data to one or more other hosts associated with the protected network includes
>
> > [E1] receiving a service request sent by the first host, serving a quarantine notification page to the first host when the service request comprises a web server request,
> >
> > [E2] and in the event the service request comprises a DNS query, providing in response an IP address of a quarantine server configured to serve the quarantine notification page if a host name that is the subject of the DNS query is not associated with a remediation host configured to provide data usable to remedy the insecure condition; and
>
> [F] permitting the first host to communicate with the remediation host.

Ex. 1 at 22:14–49.

97. The preamble of Claim 19 is nonlimiting. But if it is interpreted to be otherwise, CEP is a "computer program product for protecting a network." For example, Google touts that CEP is "Google Cloud's zero-trust solution that enables an organization's workforce to access web applications securely from anywhere, without the need for VPN and without fear of malware, phishing, and data loss":

> Chrome Enterprise Premium is Google Cloud's zero-trust solution that enables an organization's workforce to access web applications securely from anywhere, without the need for VPN and without fear of malware, phishing, and data loss.
>
> By using Chrome Enterprise Premium, you can manage access for apps on Google Cloud, other clouds, and on-premises, define and enforce access policies based on user, device, and other contextual factors, and make apps more accessible and responsive through Google's global network.

See Ex. 14, p. 1 (available at https://cloud.google.com/beyondcorp-enterprise/docs) (last accessed mid Jan. 2025 and incorporated by reference). Additionally, CEP provides endpoint (*e.g.*, "host" computer,) posture assessments and ensure that endpoints meet security and compliance policies before they are allowed to access a protected network. *See id.* Accordingly, and to the extent the preamble of Claim 19 is deemed limiting, CEP meets the limitation.

98. CEP also meets the requirements of limitation A of Claim 19, which requires "detecting an insecure condition on a first host that has connected or is attempting to connect to a protected network." For example, CEP delivers endpoint posture assessments and ensures that endpoints meet security and compliance policies before they are allowed to connect to secure resources, i.e. network, housed on the Google Cloud:

Endpoint Verification overview

Send feedback

This document describes the basic concepts of Endpoint Verification.

Endpoint Verification lets security administrators or security operations professionals secure Google Cloud, on-premises apps and resources, and Google Workspace apps.

Endpoint Verification is part of Google Cloud Chrome Enterprise Premium and is available to all Google Cloud, Cloud Identity, Google Workspace for Business, and Google Workspace for Enterprise customers.

When to use Endpoint Verification

Use Endpoint Verification when you want an overview of the security posture of the devices that are used to access your organization's resources, such as laptops and desktops.

As a security administrator or security operations professional, your goal is to manage secure access to your organization's resources. The employees of your organization can use either the company-owned devices or their unmanaged personal devices to access the organization's resources. When Endpoint Verification is installed on the devices that access your organization's resources, it collects and reports device inventory information. You can use this device inventory information to manage secure access to your organization's resources.

When paired with the other offerings of Chrome Enterprise Premium, Endpoint Verification helps enforce fine-grained access control on your Google Cloud resources.

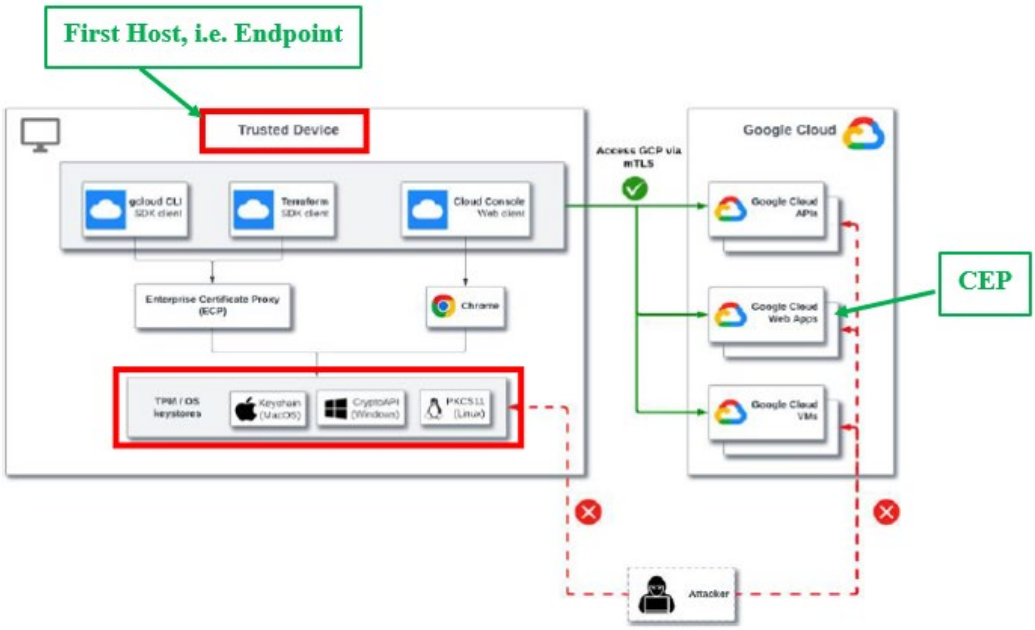
See Ex. 15, Endpoint Verification Overview, p. 1 (available at https://cloud.google.com/endpoint-verification/docs/overview) (last accessed mid Jan. 2025 and incorporated by reference).

Chrome Enterprise Premium presents a security model that allows for greater security posturing and policy for both applications and devices, while providing end users a better user experience no matter where they access from or what type of device they use to do so:

- For administrators:
  - Strengthen security posture to account for dynamic changes in a user's context.
  - Shrink the access perimeter to only those resources that an end user should be accessing.
  - Enforce device security postures for employees, contractors, partners, and customers for access, no matter who manages the devices.
  - Extend security standards with per-user session management and multifactor authentication.
- For end users:
  - Allow all end users to be productive everywhere without compromising security.
  - Allow the right level of access to work applications based on their context.
  - Unlock access to personally-owned devices based on granular access policies.

See Ex. 16, Chrome Enterprise Premium Overview, pp. 1–2 (available at https://cloud.google.com/beyondcorp-enterprise/docs/overview) (last accessed mid Jan. 2025 and incorporated by reference). Endpoint Verification is a Google webapp that is integrated into and used by various Google products, including CEP.

99. CEP further meets limitation B1 of Claim 19, which requires that "detecting [an] insecure condition includes . . . contacting a trusted computing base associated with a trusted platform module within the first host." For example, CEP uses "strong key protection" such as "secure cryptographic storage such as TPMs and OS keystores." *See* Ex._17, How To Prevent Account Takeovers With New Certificate-Based Access, p. 1 (available at https://cloud.google.com/blog/products/identity-security/how-to-prevent-account-takeovers-with-new-certificate-based-access) (last accessed mid Jan. 2025, annotated to show where in the architecture CEP and the First host reside, and incorporated by reference). As the below annotated Google produced figure establishes, many products, including CEP, are integrated into Google Cloud. In operation, what Google calls a Trusted Device (top red box below) and the '705 Patent calls a "first host" or "endpoint" (noted in top green box below), includes a TCB/TPM (see bottom red box below) and uses a Chrome browser, working with other Google Software installed on the Trusted Device, for contact between applications, web apps, virtual machines, etc., operating within the Google Cloud Platform, including CEP.



See id. at 2 (annotated). Some of the information CEP requests from the Trusted Device, i.e. the first host, are digitally-signed certificates housed in the TCB/TPM to determine whether the first host is secure and can be trusted to access protected resources, i.e. networks, also housed within the Google Cloud:

> Certificate-based access can help you build a multi-layered defense against account takeovers, bolster data security, and maintain user trust. At Google, we have used this unique Zero Trust approach for many years as a strong defense to protect our technical infrastructure and employees. Now with CBA, we are extending this same level of security to our Google Cloud customers. Here are the key attributes of this approach:
>
> - **Certificate-based access control**: Granular access policies with X.509 device certificates ensures only legitimate users with the correct certificate can access cloud resources.
> - **Protections beyond initial login**: In contrast to using mTLS only for authentication, CBA also evaluates every authorization request to help safeguard resource access.
> - **Strong key protection:** CBA's use of mTLS leverages secure cryptographic storage such as TPMs and OS keystores. Tooling such as Enterprise Certificate Proxy (ECP) are offered to users that empower them to safeguard private keys helping to ensure keys remain inaccessible to attackers without physical device access.
>
> CBA allows you to enforce certificate-based authentication for all of Google Cloud, including specific resources (VM, Console, API), individual groups or organizations, across multiple end user vectors from browser to gcloud and terraform command-line-interfaces. It is integrated with many services in Google Cloud for effective access enforcement to cloud resources. For example, Context-Aware Access for Google APIs and the Cloud Console, Identity Aware Proxy (IAP) for web apps and workloads, and VPC Service Controls (VPC-SC) for network enforcement.

See id. at 4 (annotated). As the above Google produced system architecture description establishes, CEP works with CBA to detect a possible insecure condition on a Trusted Device by contacting a trusted computing base associated with a trusted platform module that is located within the Trusted Device and meets the requirements of limitation B1 of Claim 19 of the '705 Patent.

100. To be clear, the "first host," i.e. the Trusted Device in Google's vernacular, is not an active element of the Asserted Clams. Rather, it is a component of the "environment" in which the active claim elements must function. Recently, the Federal Circuit explained "environmental" claiming and the role it plays in determining direct infringement. *INVT SPE LLC v. Int'l Trade Comm'n*, 46 F.4th 1361 (Fed. Cir. 2022). As explained in *INVT:*

> [t]he base station is part of "the environment" in which the [actively claimed] user device must function. *Advanced Software Design Corp. v. Fiserv, Inc.,* 641 F.3d 1368, 1374 (Fed. Cir. 2011). The claims have specific requirements for the data signal that the user device's receiving section and data obtaining section handle and

> process when the device is activated and put into operation. . . . To understand whether a user device can ever receive a data signal with the particularized characteristics set forth in the claim, it is necessary to know whether the base station (i.e., the communicating party) is capable of transmitting that particular type of data signal to the user device. ***Therefore, although the recited base station is not "a limitation on the claimed invention itself,"*** *Nazomi Communs., Inc. v. Nokia Corp.,* 739 F.3d 1339, 1345 (Fed. Cir. 2014*)*, ***in the sense that an infringer would not need to, for instance, use, make, or sell the base station***, the base station's operation affects whether the claims are met, *see, e.g.*, *Advanced Software Design,* 641 F.3d at 1373-74.
>
> In *Advanced Software*, the claimed invention was for validating a check, to prevent check fraud, and involved either decrypting or encrypting information on the check. The preamble of the claim set out that the check included "selection information [that] is encrypted" to generate a control code and a "control code [which] is printed on the [check]." We held that these steps in the preamble "define[d] the financial instrument that the claimed system validates" as opposed to setting forth steps that would have to be performed by the accused infringer. *Advanced Software*, 641 F.3d at 1373-74 & 1374 n.1. Nevertheless, the accused infringer would infringe "only by validating checks that [had] been encrypted and printed in accordance with steps described in the preamble." *Id*. at 1374.
>
> Like in *Advanced Software*, the claimed device of the [INVT] patent operates in an environment that involves actions of another device (the communicating party, i.e., the base station). The claimed device's capability of performing the recited functions depends on being supplied a certain modulated and encoded signal, which, in turn, requires the supplier (the communicating party) to actually supply that signal. Because the communicating party (base station) generates the necessary environment, its operations must be known to determine whether the accused device infringes, i.e., is capable of performing the claimed functions.

Id. at 1375 (emphasis added); *see also Uniloc USA, Inc. v. Microsoft Corp*., 632 F.3d 1292, 1308-09 (Fed. Cir. 2011) (holding that claimed "station forming part of a registration system . . . including local licensee unique ID generating means," only "define[d] the environment in which that [remote] registration station must function.").

101. Claim 19 of the asserted '705 Patent recites a "computer program product for protecting a network . . ." and that product performs the following active functions:

> **Receiving** a service request from a first host;

Contacting a trusted computing base (TCB) associated with a trusted platform module (TPM) located within the first host;

Receiving a response from the TCB;

Determining from that response whether the first host is infested or clean;

Quarantining, if infestation is detected, the first host; and

Permitting the first host to communicate with a remediation host.[1]

102. CEP is a computer program product that protects resources found on Google Cloud and performs each active recited function of the claims, i.e., the bolded terms, as explained herein, specifically including in the exhibits hereto, which are incorporated herein by this reference. The rest of the claim language, specifically including the "first host," is environment within which the active verbs of the claim must operate. K.Mizra thus need not establish that Google supplies the "first host" for it to prove that Google directly infringes the asserted claims. That said, though,

[1] Claim 19, with the active verbs highlighted, is here provided for ease of reference:

19. A computer program product for protecting a network, the computer program product being embodied in a non-transitory computer readable medium and comprising computer instructions for:

detecting an insecure condition on a first host that has connected or is attempting to connect to a protected network, wherein detecting the insecure condition includes contacting a trusted computing base associated with a trusted platform module within the first host, receiving a response, and determining whether the response includes a valid digitally signed attestation of cleanliness, wherein the valid digitally signed attestation of cleanliness includes at least one of an attestation that the trusted computing base has ascertained that the first host is not infested, and an attestation that the trusted computing base has ascertained the presence of a patch or a patch level associated with a software component on the first host;

when it is determined that the response does not include a valid digitally signed attestation of cleanliness, quarantining the first host, including by preventing the first host from sending data to one or more other hosts associated with the protected network, wherein preventing the first host from sending data to one or more other hosts associated with the protected network includes receiving a service request sent by the first host, serving a quarantine notification page to the first host when the service request comprises a web server request, and in the event the service request comprises a DNS query, providing in response an IP address of a quarantine server configured to serve the quarantine notification page if a host name that is the subject of the DNS query is not associated with a remediation host configured to provide data usable to remedy the insecure condition; and

permitting the first host to communicate with the remediation host.

Google does supply all relevant portions of the first host and created a secure network architecture in which that first host is to operate.

103. Limitation B2 requires that "detecting the insecure condition" also includes "receiving a response and determining whether the response includes a valid digitally signed attestation of cleanliness." CEP meets this limitation as it receives from the first host computer, i.e. the Trusted Device, requested information and then determines, based on the information received, whether the Trusted Device that is attempting to access secure resources housed in the Google Cloud (*i.e.*, a protected network) is secure and trusted. *See, e.g.*, *id.* This process requires back-and-forth communication between CEP and the Trusted Device concerning the cleanliness of the first host computer/endpoint device:

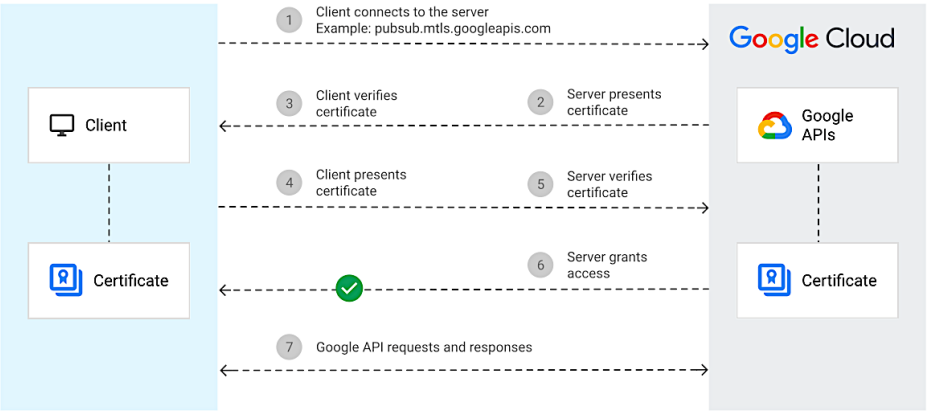
How the Google APIs validate device identity

The TLS protocol uses a technique called public key infrastructure (PKI), which relies on a pair of asymmetric keys: a public key and a private key. Anything encrypted with the private key can be decrypted only with the public key. The Google Cloud APIs use the TLS protocol to verify the identity of a device by decrypting the message encrypted by the private key using the public key of the certificate during the mTLS handshake. The successful decryption proves the possession of the private key which is only available from trusted devices.

To enable the mTLS handshake and validation process, a client must do the following:

- Establish an mTLS connection with the Google APIs by using mTLS-specific API endpoints. The mTLS-specific endpoints have the following format: `[service].mtls.googleapis.com`
- Discover and use the device certificate during the mTLS handshake. If you are using Endpoint Verification for certificate deployment, this type of certificate is automatically discovered and used by the supported clients.

The following diagram illustrates the mTLS handshake between a client and a Google API server:



See Ex. 18, Understand Mutual TLS at Google Cloud, p. 1 (available at https://cloud.google.com/beyondcorp-enterprise/docs/understand-mtls) (last accessed mid Jan. 2025 and incorporated by reference). Additionally, CPE checks digital signatures to determine the cleanliness of the host computer:

Deploy Endpoint Verification to use with certificate-based access

Send feedback

As part of a Chrome Enterprise Premium solution, Endpoint Verification provides critical device trust and security-based access control, and can help enforce fine-grained access control on your Google Cloud resources.

See Ex. 19 (annotated), Deploy Endpoint Verification To Use With Certificate-Based Access, p. 1 (available at https://cloud.google.com/beyondcorp-enterprise/docs/deploy-cba-endpoint-verification) (last accessed mid Jan. 2025 and incorporated by reference). Thus, CPE meets limitation B2 of Claim 19.

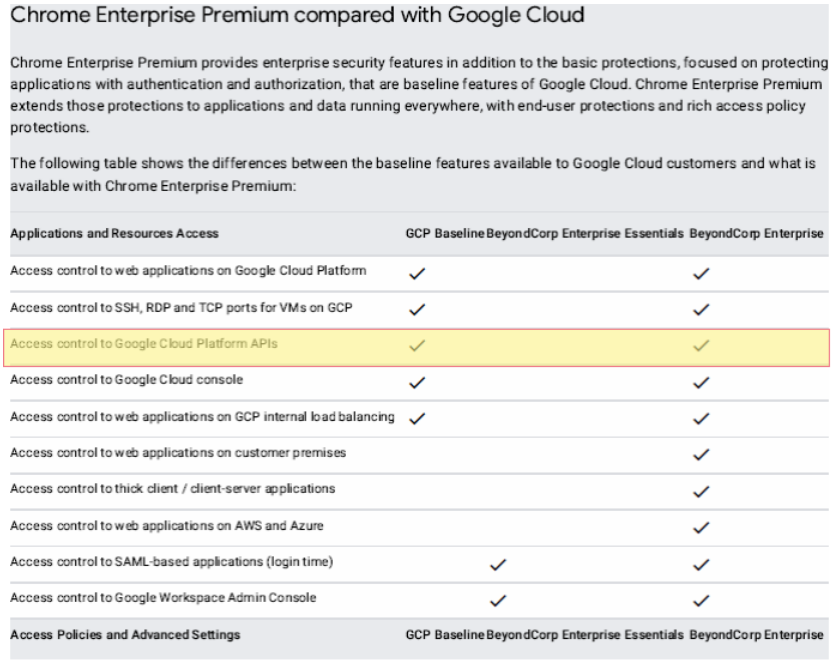
104. Limitation C requires that "the valid digitally signed attestation of cleanliness includes at least one of an attestation that the trusted computing base has ascertained that the first host is not infested, and an attestation that the trusted computing base has ascertained the presence of a patch or a patch level associated with a software component on the first host." CEP checks the compliance of each endpoint device, i.e. Trusted Device, attempting to connect to protected resources housed in the Google Cloud by performing an endpoint control check that involves matching the endpoint's configuration parameters received from the endpoint device with specific device profile attributes, such as antimalware programs, applications, etc.:

> Google announced a new feature and administration serverside API for the Chrome OS called Verified Access. This API will cryptographically validate the identity of any Chrome OS device that has connected to an enterprise's network. Additionally, it will allow the enterprise to verify that any connected devices conform to the company's security policies.
>
> The new Google API uses digital certificates stored in the hardware-based Trusted Platform Modules (TPMs) that are found in every Chrome OS device. The TPM creates a way to ensure the correct security state of the devices has not been altered.

See Ex. 20 (annotated), New Google API Will Securely Verify Chrome Devices, p. 2 (available at https://securityintelligence.com/news/new-google-api-will-securely-verify-chrome-devices/) (last accessed mid Jan. 2025 and incorporated by reference). As already explained and will be further

detailed below, Verified Access is a Google API that is used by numerous Google products to help keep data safe and secure and one of these is CEP.

105. Exhibit 16 provides the following explanation of some CEP functions/features as compared to those offered by the less secure Google Cloud product:



Chrome Enterprise Premium compared with Google Cloud

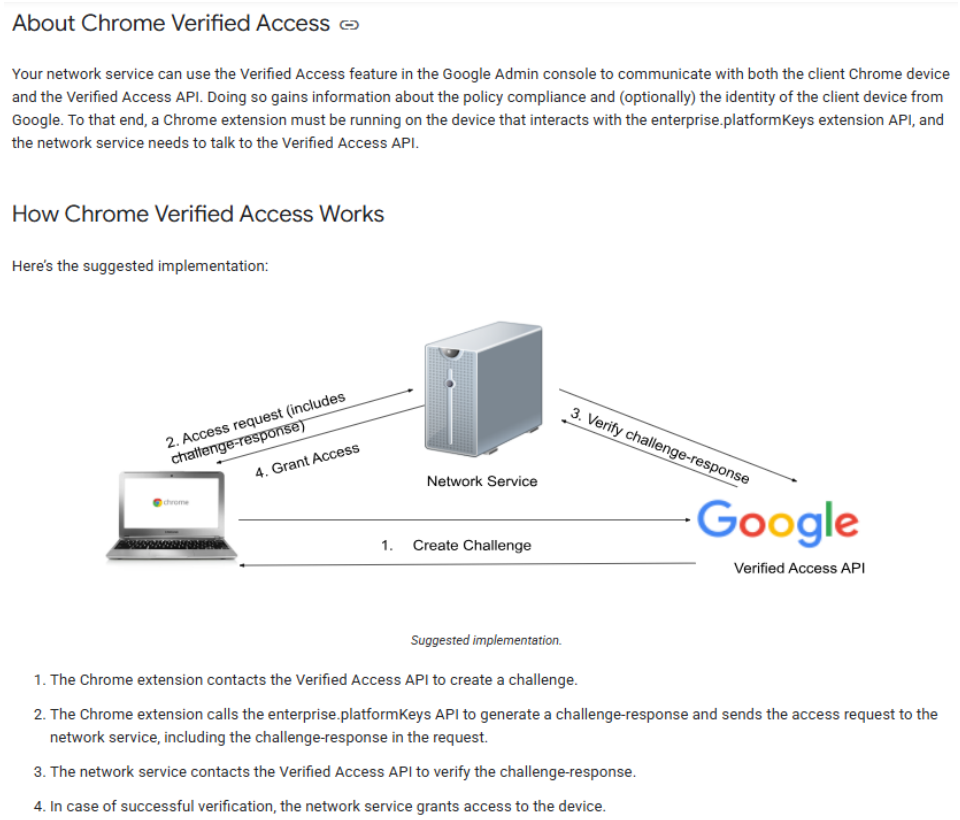
Chrome Enterprise Premium provides enterprise security features in addition to the basic protections, focused on protecting applications with authentication and authorization, that are baseline features of Google Cloud. Chrome Enterprise Premium extends those protections to applications and data running everywhere, with end-user protections and rich access policy protections.

The following table shows the differences between the baseline features available to Google Cloud customers and what is available with Chrome Enterprise Premium:

Applications and Resources Access	GCP Baseline	BeyondCorp Enterprise Essentials	BeyondCorp Enterprise
Access control to web applications on Google Cloud Platform	✓		✓
Access control to SSH, RDP and TCP ports for VMs on GCP	✓		✓
Access control to Google Cloud Platform APIs	✓		✓
Access control to Google Cloud console	✓		✓
Access control to web applications on GCP internal load balancing	✓		✓
Access control to web applications on customer premises			✓
Access control to thick client / client-server applications			✓
Access control to web applications on AWS and Azure			✓
Access control to SAML-based applications (login time)		✓	✓
Access control to Google Workspace Admin Console		✓	✓
Access Policies and Advanced Settings	**GCP Baseline**	**BeyondCorp Enterprise Essentials**	**BeyondCorp Enterprise**

https://cloud.google.com/beyondcorp-enterprise/docs/overview. What this Google document relevantly conveys is that Google Cloud and CEP (formerly called BeyondCorp Enterprise) both provide Access control to Google Cloud Platform APIs (*see* yellow highlighted box above). K.Mizra's reliance on a Google Cloud document to explain how Access Control and Verified Access works in both CEP and Google Cloud, more generally, is not an allegation of infringement against Google Cloud, but rather an explanation of how CEP infringes Claim 19 by calling Google's Verified Access API and thus determining whether the "first host" is "clean" or "infested."

106. More specifically, a Google produced document updated October 16, 2024 and found at https://developers.google.com/chrome/verified-access/overview and attached as Exhibit 21 shows how the Verified Access API is called by Chrome and thus also CEP:



About Chrome Verified Access

Your network service can use the Verified Access feature in the Google Admin console to communicate with both the client Chrome device and the Verified Access API. Doing so gains information about the policy compliance and (optionally) the identity of the client device from Google. To that end, a Chrome extension must be running on the device that interacts with the enterprise.platformKeys extension API, and the network service needs to talk to the Verified Access API.

How Chrome Verified Access Works

Here's the suggested implementation:

Suggested implementation.

1. The Chrome extension contacts the Verified Access API to create a challenge.
2. The Chrome extension calls the enterprise.platformKeys API to generate a challenge-response and sends the access request to the network service, including the challenge-response in the request.
3. The network service contacts the Verified Access API to verify the challenge-response.
4. In case of successful verification, the network service grants access to the device.

CEP, operating in Google Cloud, calls Google's Verified Access API, also operating in the Google Cloud, determining whether the "first host" is "clean" or "infested." If it is the latter, the "first host" is not allowed to access secure resources also stored in the Google Cloud. These functions satisfy the requirements of limitation C of claim 19.

107. Limitation D requires that "when it is determined that the response does not include a valid digitally signed attestation of cleanliness, quarantining the first host, including by preventing the first host from sending data to one or more other hosts associated with the protected network." CEP meets these requirements by quarantining noncompliant, *i.e.*, unclean, endpoint

devices, i.e. Trusted Devices, attempting to connect to protected resources found in the Google Cloud:

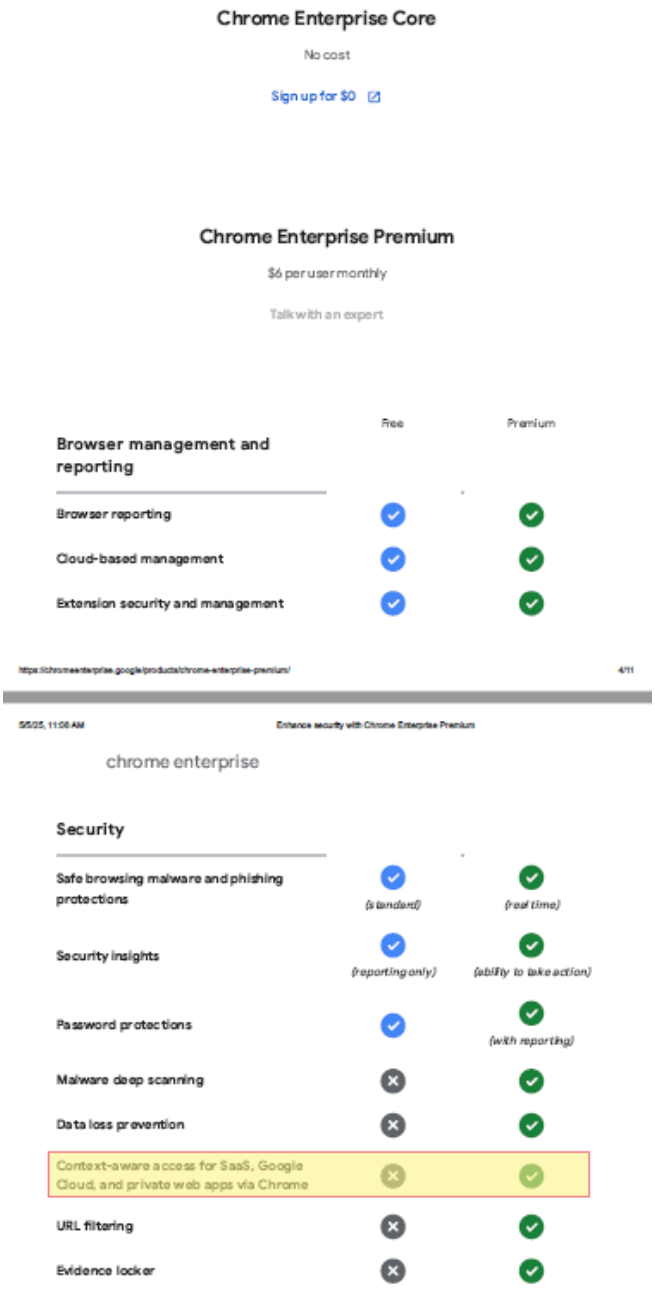
> Using Context-Aware Access, admins can set up different access levels based on a user's identity and the context of the request (location, device security status, IP address). This can help provide granular access controls without the need for a VPN, and give users access to Google Workspace resources based on organizational policies. Insights and recommendations help admins improve the cybersecurity posture of their organization by proactively identifying areas that need attention, significantly reducing the need for admins to identify these risks themselves. For example, if we detect devices with outdated operating system versions accessing corporate Workspace data, we can surface this as an Insight & pair it with a recommendation to block such devices from accessing Workspace data with a few clicks.

See Ex. 22, Context Aware Access Insights and Recommendations Are Now Generally Available, p. 3 (available at https://workspaceupdates.googleblog.com/2024/10/context-aware-access-insights-and-recommendations.html) (last accessed mid Jan. 2025 and incorporated by reference).

> Using context-aware access, you now have the option to automatically block access to Google Workspace data from compromised Android and iOS devices. A device may be counted as compromised if certain unusual events are detected, including devices that are jailbroken, bypassing of security controls, modification of restricted settings, and more.

See Ex. 23, Block Compromised Mobile Devices Using Context-Aware Access, p. 2 (available at https://workspaceupdates.googleblog.com/2024/05/block-compromised-mobile-devices-using-context-aware--access.html) (last accessed mid Jan. 2025 and incorporated by reference). Context-Aware Access is another Google product that is housed in the Googler Cloud that can be used by many other Google products. One of those is CEP, which uses Context-Aware Access to

facilitate quarantining of first hosts," i.e. Trusted Devices. To put a finer point in it, the following Google produced content specifically shows that Context-Aware Access is a security feature of CEP:



CEP meets limitation D of Claim 19.

108. Limitation E1 requires that "preventing the first host from sending data to one or more other hosts associated with the protected network includes . . . receiving a service request sent by the first host [and] serving a quarantine notification page to the first host when the service request comprises a web server request." CEP determines if an endpoint, i.e. Trusted Device, posture information stored in the TMP/TCB matches the profile designated for such a device. If that endpoint information does not match, it is quarantined and restricted from accessing the protected network housed on the Google Cloud, preventing the Trusted Device from sending data to or receiving data from the Google Cloud housed protected resources until problems causing the Trusted Device to be blocked are remediated:

> Using remediation messages and custom messages in Context-Aware Access, you can help users unblock themselves when a policy prevents them from accessing an app. These optional (but recommended) messages can help get users back to productivity and reduce support calls for admins.
>
> For example, say that a user on a mobile device is using Gmail in the office successfully during the day, but is blocked when they try to access Gmail at home in the evening. When remediation messages are enabled, they will see guidance on how to address the reason they are blocked.
>
> Remediation and custom messages are supported for access levels created in both Basic mode and Advanced mode. Also, they are supported for both Core Services and SAML apps.

See Ex. 24, Allow Users to Unblock Apps With Remediation Messages in Context Aware Access, p. 1 (available at https://support.google.com/a/answer/11560430?sjid=18176077754207218303-EU) (last accessed mid Jan. 2025 and incorporated by reference). A quarantine message is also delivered to the unclean endpoint device notifying its user of the quarantine.

Use remediation messages and custom messages to help your users unblock themselves

When blocked, your users can encounter:

- **Default message**—Displays if you have not added remediation messages or custom messages. An example default message is: **Your organization's policy is blocking access to this app**.
- **Remediation messages**—Replaces the default message. The messages are system generated, and correspond to the specific policy violation that blocked the user. Remediation messages can present several remediation options to the user, which they can expand by clicking **Show more options**. In the case of several remediation options, the user needs to complete the steps for any one of the available options to unblock themselves.
- **Custom message**—Adds specific help for the user, such as additional advice on getting unblocked or a helpful link to click. You add custom messages as needed. A custom message can appear in conjunction with the default message, or with remediation messages.

See id. Accordingly, CEP meets limitation E1 of Claim 19.

109. Limitation E2 requires that "preventing the first host from sending data to one or more other hosts associated with the protected network includes…in the event the service request comprises a DNS query, providing in response an IP address of a quarantine server configured to serve the quarantine notification page if a host name that is the subject of the DNS query is not associated with a remediation host configured to provide data usable to remedy the insecure condition." CEP meets these requirements by providing to the Trusted Device a quarantine notification page containing links, or IP address(es), with resources (*i.e.* quarantine servers) configured to resolve the issue causing the quarantine. In other words, CEP provides remediation information to bring the Trusted Device into system security compliance so that it can access protected resources housed in the Google Cloud:

Use remediation messages and custom messages to help your users unblock themselves

When blocked, your users can encounter:

- **Default message**—Displays if you have not added remediation messages or custom messages. An example default message is: **Your organization's policy is blocking access to this app**.
- **Remediation messages**—Replaces the default message. The messages are system generated, and correspond to the specific policy violation that blocked the user. Remediation messages can present several remediation options to the user, which they can expand by clicking **Show more options**. In the case of several remediation options, the user needs to complete the steps for any one of the available options to unblock themselves.
- **Custom message**—Adds specific help for the user, such as additional advice on getting unblocked or a helpful link to click. You add custom messages as needed. A custom message can appear in conjunction with the default message, or with remediation messages.

See id. Accordingly, CEP meets limitation E2 of Claim 19.

110. Limitation F requires "permitting the first host to communicate with the remediation host." CEP meets these requirements by allowing a quarantined endpoint, i.e. Trusted Device, to access remediation resources to help make the device complaint with system safety requirements. *See id.*

111. Google's acts of infringement have occurred within this District and elsewhere throughout the United States.

112. As a result of Google's infringing conduct, K.Mizra has suffered damages. Google is liable to K.Mizra in an amount that adequately compensates K.Mizra for Google's infringement in an amount that is no less than a to-be-calculated fully paid-up, lump-sum, reasonable royalty, together with interest and costs as fixed by this Court under 25 U.S.C. § 284.

VI. SECOND CLAIM FOR RELIEF
(Patent Infringement Under 35 U.S.C. § 271 of the '120 Patent)

113. K.Mizra incorporates paragraphs 1 through 112 as though fully set forth herein.

114. The '120 Patent includes 16 claims.

115. Google has directly infringed one or more claims of the '120 Patent by making, importing, using, offering for sale, and/or selling the Accused Hyperparameter Tuning Instrumentalities, in violation of 35 U.S.C. § 271(a).

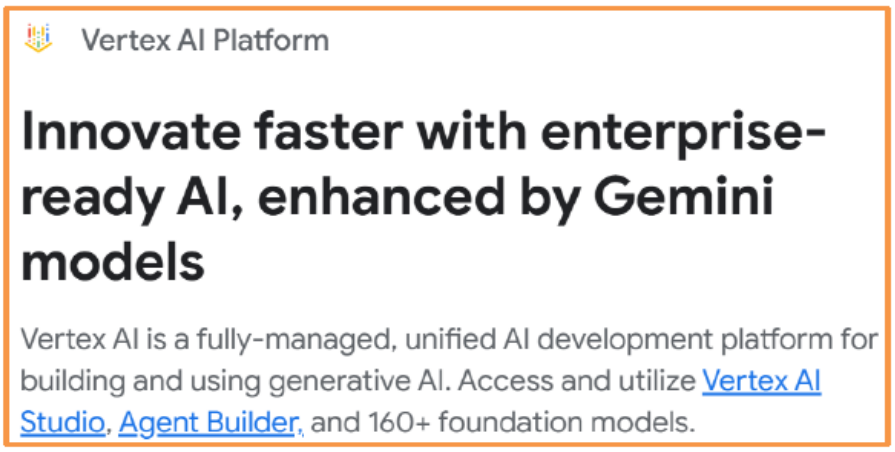
116. Based on publicly available information, the Accused Hyperparameter Tuning Instrumentalities satisfy every limitation of at least Claim 1 of the '120 Patent.

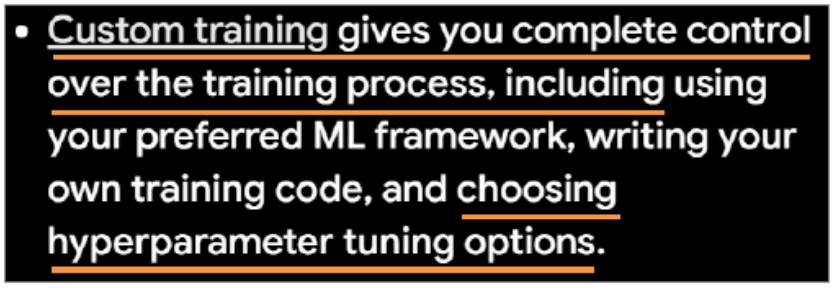
117. Claim 1 of the '120 Patent states:

> [preamble] A method of determining hyperparameters of a classifier in a machine learning system by iteratively producing an estimate of a target hyperparameter vector, each iteration comprising the steps of:
>
> [A] drawing a random sample of hyperparameter vectors from a set of possible hyperparameter vectors,
>
> [B] updating the estimate of the target hyperparameter vector by using the random sample, and
>
> [C] Selecting, from the random sample of hyperparameter vectors, a hyperparameter vector producing a best result in the present and any previous iterations, and wherein the step of updating the estimate of the target hyperparameter vector uses said hyperparameter vector producing the best result.

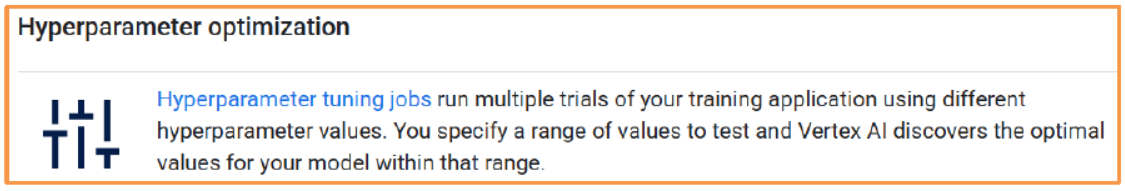
Ex. 6 at 7:14–28.

118. Regarding the preamble of Claim 1, and to the extent the preamble is determined to be limiting, the Asserted Hyperparameter Tuning Instrumentalities provide the features described in the preamble, which recites "A method of determining hyperparameters." For example, Google touts that its Vertex AI Platform is a "fully-managed, unified AI development platform for building and using generative AI."; "Custom training gives you complete control over the training process, including using your preferred ML framework, writing your own training code, and choosing hyperparameter tuning options."; and "Hyperparameter tuning jobs run multiple trials of your training application using different hyperparameter values."





See Ex. 25, "Innovate faster with enterprise-ready AI, enhanced by Gemini models," (available at https://cloud.google.com/vertex-ai?hl=en (accessed May 21, 2025) and incorporated by reference).

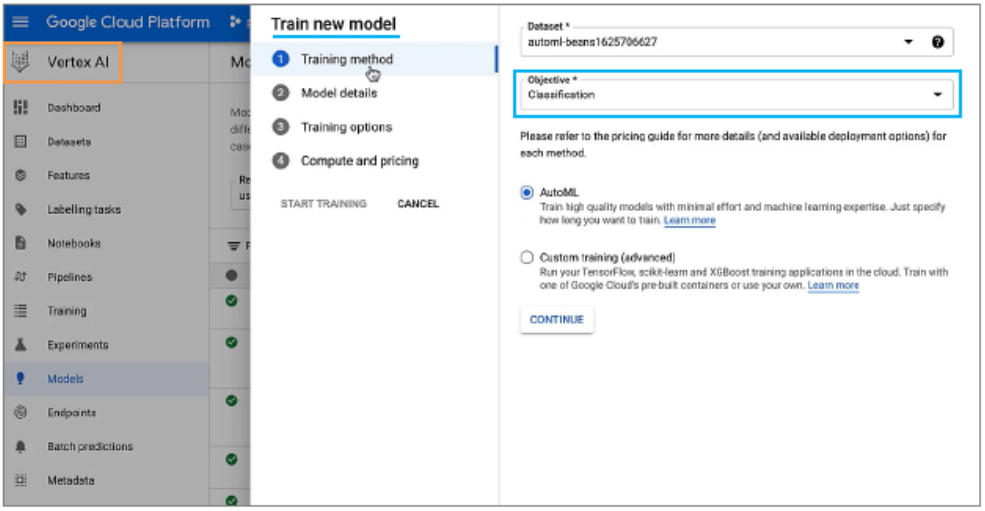


See Ex. 26, "Vertex AI custom training overview," (available at https://cloud.google.com/vertex-ai/docs/training/overview (accessed May 21, 2025) and incorporated by reference). Accordingly, and to the extent the preamble of Claim 1 is deemed limiting, the Accused Hyperparameter Tuning Instrumentalities meet the limitation.

119. The preamble also includes "of a classifier in a machine learning system".

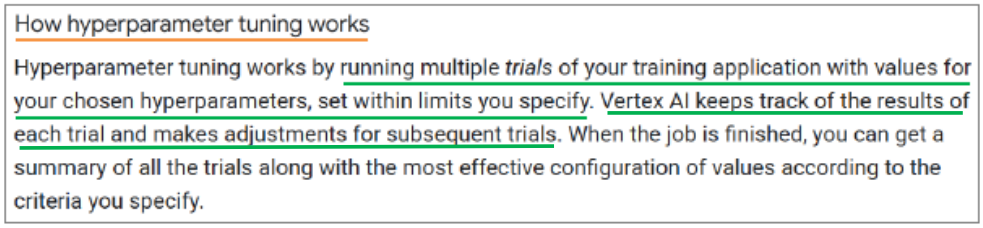
> Vertex AI is a machine learning (ML) platform that lets you train and deploy ML models and AI applications. Vertex AI combines data engineering, data science, and ML engineering workflows, enabling team collaboration using a common toolset. Learn more.

See Ex. 27, "Vertex AI documentation," (available at https://cloud.google.com/vertex-ai/docs (accessed May 21, 2025) and incorporated by reference).

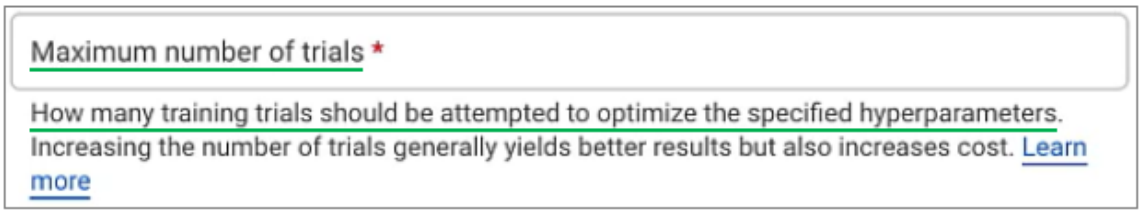


See screenshot above of page at https://www.youtube.com/watch?v=8hZ_cBwNOss (last accessed mid-May 2025) and incorporated by reference). Thus, and again only to the extent that this portion of the preamble of Claim 1 is deemed limiting, the Accused Hyperparameter Tuning Instrumentalities meet the limitation.

120. The preamble further includes "by iteratively producing an estimate of a target hyperparameter vector":

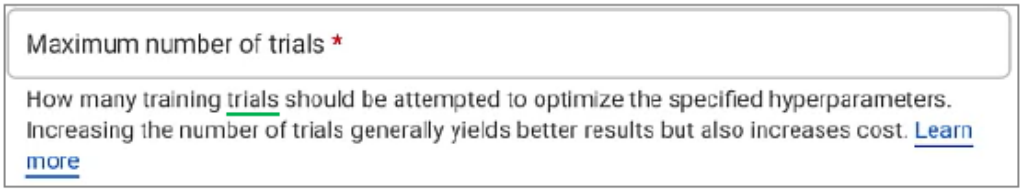


See Ex. 28, "Overview of hyperparameter tuning," (available at https://cloud.google.com/vertex-ai/docs/training/hyperparameter-tuning-overview (accessed May 21, 2025) and incorporated by reference).



See screenshot above of page at https://www.youtube.com/watch?v=8hZ_cBwNOss (last accessed mid-May 2025) and incorporated by reference). Accordingly, and to the extent this portion of the preamble of Claim 1 is deemed limiting, the Accused Hyperparameter Tuning Instrumentalities meet the limitation.

121. The preamble further also includes "each iteration comprising the steps of:"



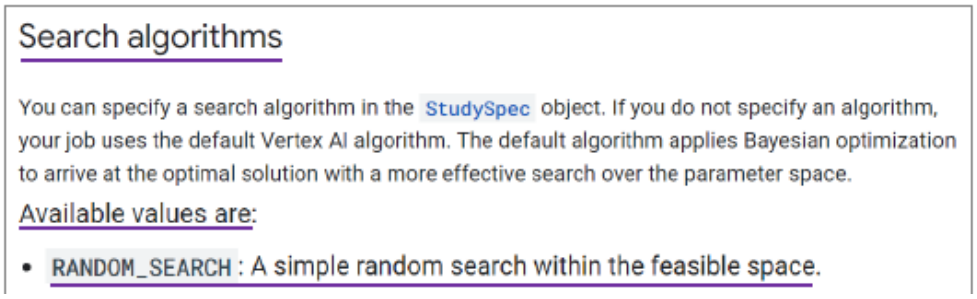
Id. Again, then, to the extent this portion of the preamble of Claim 1 is deemed limiting, the Accused Hyperparameter Tuning Instrumentalities meet the limitation.

122. Accordingly, the Accused Hyperparameter Tuning Instrumentalities meet limitation of the preamble of Claim 1.

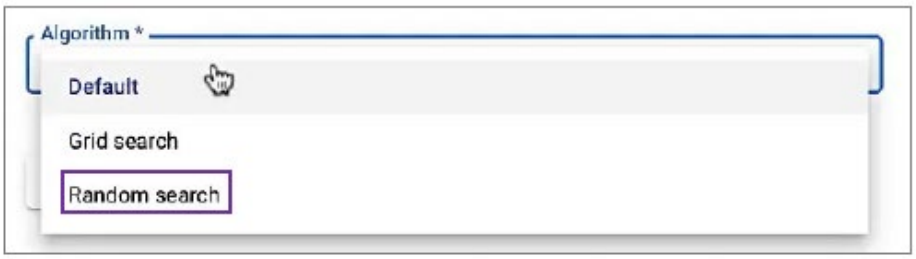
123. Limitations A and B of Claim 1 require:

> [A] drawing a random sample of hyperparameter vectors from a set of possible hyperparameter vectors,
>
> [B] updating the estimate of the target hyperparameter vector by using the random sample, and

Ex. 6 at 7:19–22. Google has disclosed:

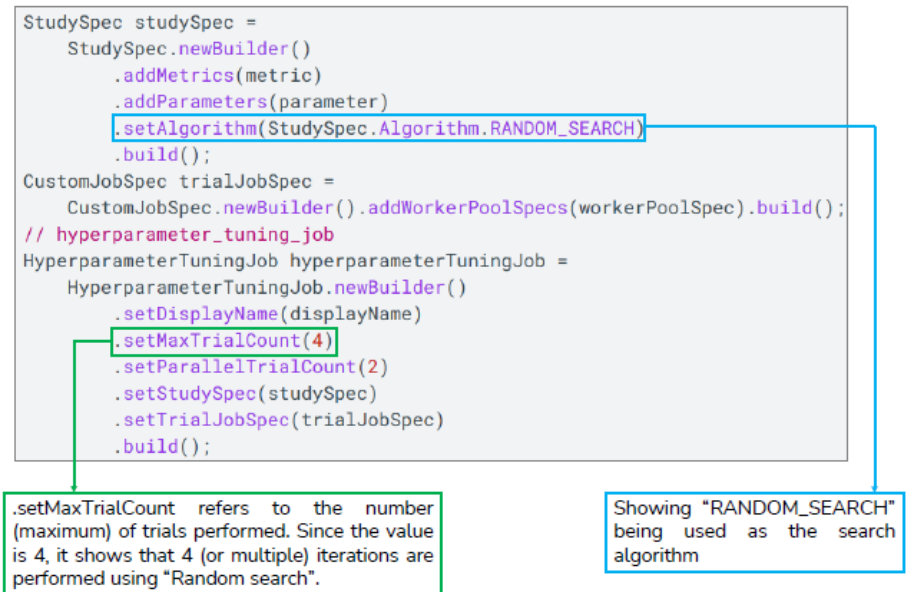


See Ex. 28.



See screenshot above of page at https://www.youtube.com/watch?v=8hZ_cBwNOss (last accessed mid-May, 2025) and incorporated by reference).

124. Limitation A requires "drawing a random sample of hyperparameter vectors from a set of possible hyperparameter vectors" and the Google system meets this requirement:



See Ex. 29, "Create a hyperparameter tuning job for python package," (available at https://cloud.google.com/vertex-ai/docs/samples/aiplatform-create-hyperparameter-tuning-job-python-package-sample?hl=en (accessed May 21, 2025) and incorporated by reference).

125. Accordingly, the Accused Hyperparameter Tuning Instrumentalities meet Limitation A of Claim 1.

126. Limitation B requires "updating the estimate of the target hyperparameter vector by using the random sample". Google has stated:

> **Random search**: set up a grid of hyperparameter values and select *random* combinations to train the model and score. The number of search iterations is set based on time/resources.
>
> Random search is surprisingly efficient compared to grid search. Although grid search will find the optimal value of hyperparameters (assuming they are in your grid) eventually, random search will usually find a "close-enough" value in far fewer iterations. This great paper explains why this is so: grid search spends too much time evaluating unpromising regions of the hyperparameter search space because it has to evaluate every single combination in the grid. Random search in contrast, does a better job of exploring the search space and therefore can usually find a good combination of hyperparameters in far fewer iterations.

See Ex. 30, excerpts of "Intro to Model Tuning: Grid and Random Search," (available at https://www.kaggle.com/code/willkoehrsen/intro-to-model-tuning-grid-and-random-search (last accessed mid mid-May, 2025) and incorporated by reference).

> Randomized search is another method for hyperparameter optimization that can be more efficient than grid search in some cases. With randomized search, instead of specifying a list of values for each hyperparameter, you specify a distribution for each hyperparameter. The randomized search algorithm will then sample values for each hyperparameter from its corresponding distribution and train a model using the sampled values. This process is repeated a specified number of times, and the optimal values for the hyperparameters are chosen based on the performance of the models.

See Ex. 31, "Comparing Randomized Search and Grid Search for Hyperparameter Estimation in Scikit Learn," (available at https://www.geeksforgeeks.org/comparing-randomized-search-and-grid-search-for-hyperparameter-estimation-in-scikit-learn/ (accessed May 21, 2025) and incorporated by reference).

127. Accordingly, the Accused Hyperparameter Tuning Instrumentalities also meet Limitation B of Claim 1.

128. Limitation C requires "selecting, from the random sample of hyperparameter vectors, a hyperparameter vector producing a best result in the present and any previous iterations, and wherein the step of updating the estimate of the target hyperparameter vector uses said hyperparameter vector producing the best result." Google describes:

> How hyperparameter tuning works
>
> Hyperparameter tuning works by running multiple *trials* of your training application with values for your chosen hyperparameters, set within limits you specify. Vertex AI keeps track of the results of each trial and makes adjustments for subsequent trials. When the job is finished, you can get a summary of all the trials along with the most effective configuration of values according to the criteria you specify.

See Ex. 28.

129. Accordingly, the Accused Hyperparameter Tuning Instrumentalities meet Limitation C of Claim 1.

130. Google's acts of infringement have occurred within this District and elsewhere throughout the United States.

131. As a result of Google's infringing conduct, K.Mizra has suffered damages. Google is liable to K.Mizra in an amount that adequately compensates K.Mizra for Google's infringement in an amount that is no less than a to-be-calculated fully paid-up, lump-sum, reasonable royalty, together with interest and costs as fixed by this Court under 25 U.S.C. § 284.

VII. THIRD CLAIM FOR RELIEF
(Patent Infringement Under 35 U.S.C. § 271 of the '717 Patent)

132. K.Mizra incorporates paragraphs 1 through 131 as though fully set forth herein.

133. The '717 Patent includes 20 claims.

134. Google has directly infringed one or more claims of the '717 Patent by making, importing, using, offering for sale, and/or selling the Accused Mesh Networking Instrumentalities, in violation of 35 U.S.C. § 271(a).

135. Based on publicly available information, the Accused Mesh Networking Instrumentalities satisfy every limitation of at least Claim 11 of the '717 Patent.

136. Open Thread is an open-source implementation of the Thread networking protocol released by Google. A description of Open Thread is provided at www.openthread.io. On information and belief, openthread.io is owned and operated by Google.

137. On information and belief, Open Thread is an implementation of the Thread networking technology used in Google Nest products.



OpenThread released by Google is an open-source implementation of Thread® (http://threadgroup.org). Google has released OpenThread to make the networking technology used in Google Nest products more broadly available to developers, in order to accelerate the development of products for the connected home and commercial buildings.

See Ex. 32, Homepage of openthread.io (available at https://openthread.io/ (accessed May 23, 2025) and incorporated by reference).

138. Claim 11 of the '717 Patent states:

> [preamble] A station for use in a wireless communication network, the station comprising:
>
> [A] a transmitter arrangement, a receiver arrangement, and a processor circuit coupled to the transmitter arrangement and the receiver arrangement,

[B] the processor configured to start up in a not-associated state and to cause the transmitter arrangement to transmit association request messages in the not-associated state,

[C] the processor circuit being configured to switch to an associated state upon reception at the receiver arrangement of an association grant in response to one of the association request messages,

[D] the processor circuit being configured to establish an associated route for communication with an association unit during subsequent communication of operating messages to an association unit,

[E] the associated route including a source of the association grant and, if the source of the association grant is not the association unit, the operating route between the source of the association grant and the association unit previously established for the source of the association grant in response to association request messages and/or association grants for intermediate stations,

[F] the processor circuit being configured to cause the transmitter arrangement to initiate and transmit association grants in response to association request messages, but only after switching to the associated state.

139. Regarding the preamble of Claim 11, and to the extent the preamble is determined to be limiting, the Accused Mesh Networking Instrumentalities provide the features described in the preamble, which recites "A station for use in a wireless communication network." For example, Google devices including the Nest WiFi, the Nest Hub Max, and the Nest Hub (2nd Gen) are stations configured for use in a wireless network implementing the Thread networking protocol.



A more reliable smart home network with Thread.

Thread is the latest technology for keeping all your smart home devices connected. Nest Hub Max and Nest Hub (2nd gen) have a built-in Thread border router to connect your Thread mesh network of smart devices to your Wi-Fi network. Your Thread-enabled network will be faster, use less energy, and reach more places throughout and around your home.

Ex. 33, "Nest Hub Smart Displays for your Home – Google Store," (available at https://store.google.com/category/nest_hubs_displays (accessed May 12, 2025) and incorporated by reference).

Thread devices join users' existing home networks through a Thread Border Router. Just like a Wi-Fi router can bridge Wi-Fi and Ethernet devices into a single network, a Thread Border Router allows Thread devices to become part of users' networks.

Google devices like the Nest WiFi, Google Nest Hub Max and Google Nest Hub (2nd gen) have Thread radios built-into them and act as Thread Border Routers.

Ex. 8.

140. Similarly, the Nest Connect is a station configured for use in a Thread wireless network.

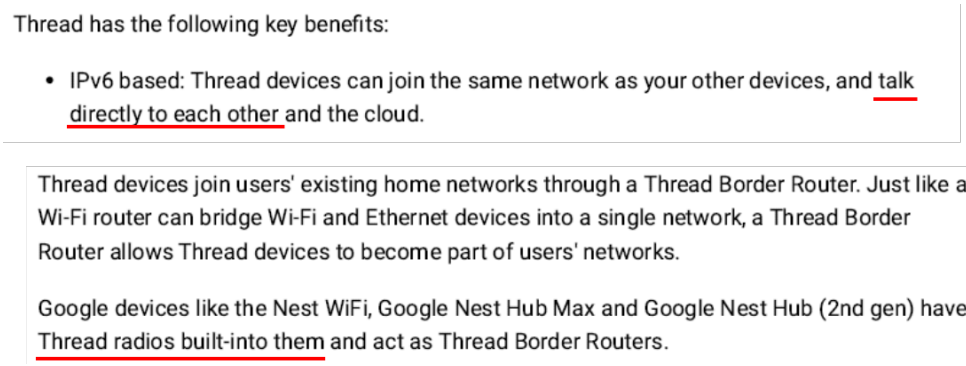
In an emailed statement, Grant Erickson, president of the Thread Group said, "Nest Detect relies on Thread's low power architecture to last for long periods of time on one battery. Nest Connect is a Thread router, which seamlessly extends both the reach and connectivity of the Thread network. Nest Guard provides seamless IP-based routing between Thread, Wi-Fi, and cellular, and their interoperability with the Yale Linus Lock is the posterchild of Thread's benefits in action."

Ex. 11.

141. Accordingly, and to the extent the preamble of Claim 11 is deemed limiting, the Accused Mesh Networking Instrumentalities meet the limitation.

142. Limitation A of Claim 11 of the Asserted Mesh Networking Patent requires a transmitter arrangement, a receiver arrangement, and a processor circuit coupled to the transmitter arrangement and the receiver arrangement.

143. The Accused Mesh Networking Instrumentalities include a Thread radio with a transmitter arrangement and a receiver arrangement, as well as a processor circuit coupled to the transmitter arrangement and receiver arrangement that processes transmissions and receptions. For example, Google's documents confirm that each of the Nest WiFi, the Google Hub Max, and the Google Nest Hub (2nd Gen) include Thread radios.



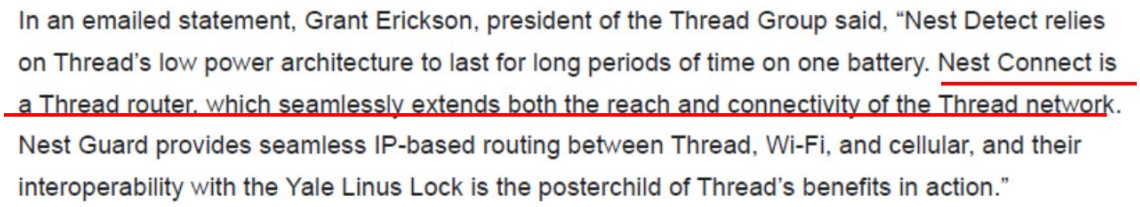
Thread has the following key benefits:

- IPv6 based: Thread devices can join the same network as your other devices, and talk directly to each other and the cloud.

Thread devices join users' existing home networks through a Thread Border Router. Just like a Wi-Fi router can bridge Wi-Fi and Ethernet devices into a single network, a Thread Border Router allows Thread devices to become part of users' networks.

Google devices like the Nest WiFi, Google Nest Hub Max and Google Nest Hub (2nd gen) have Thread radios built-into them and act as Thread Border Routers.

Ex. 8. Similarly, a statement from the Thread Group (of which Google is a member) confirms that the Nest Connect is a Thread router.



In an emailed statement, Grant Erickson, president of the Thread Group said, "Nest Detect relies on Thread's low power architecture to last for long periods of time on one battery. Nest Connect is a Thread router, which seamlessly extends both the reach and connectivity of the Thread network. Nest Guard provides seamless IP-based routing between Thread, Wi-Fi, and cellular, and their interoperability with the Yale Linus Lock is the posterchild of Thread's benefits in action."

Ex. 11. Indeed, in order to communicate with other devices, each Thread device (including the Accused Mesh Networking Instrumentalities) must necessarily include a transmitter arrangement

(to transmit messages from the Thread device to other devices in the Thread network) and a receiver arrangement (to receive communications, including responses to its transmitted messages, from other devices in the Thread network). Thread devices, including the Accused Mesh Networking Instrumentalities, must also include a processor coupled to both the transmitter arrangement and the receiver arrangement to, among other things, initiate transmissions, respond to messages sent by other devices in the Thread network, and process received messages. The operations of the processor are described in further detail in connection with Limitations B through F.

144. Accordingly, the Accused Mesh Instrumentalities meet Limitation A of Claim 11.

145. Limitation B of Claim 11 of the Asserted Mesh Networking Patent requires the processor is configured to start up in a not-associated state and to cause the transmitter arrangement to transmit association request messages in the not-associated state.

146. The Accused Mesh Networking Instrumentalities include this feature. For example, Thread devices, such as the Accused Mesh Networking Devices, have a Detached state when the Thread device has not yet attached to a Thread network.

> **Note:** The Detached state is observed when Thread is enabled but the device is not attached to a Thread network. The Disabled state is observed when Thread is disabled on the device.

Ex. 34, "Developing with OpenThread APIs, (available at https://openthread.io/codelabs/openthread-apis#1 (accessed May 23, 2025) and incorporated by reference).

147. In order to attach to a Thread network, a Thread device such as the Accused Mesh Networking Instrumentalities goes through the Attach process.

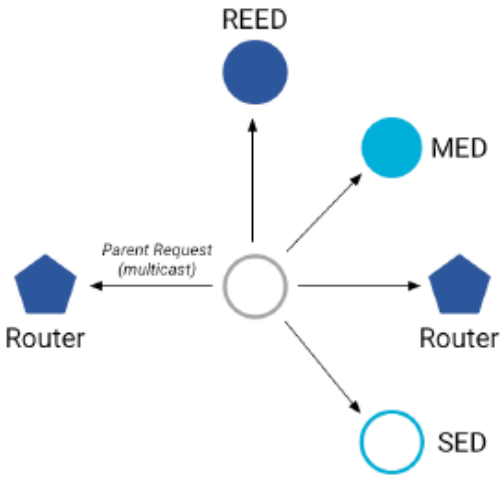
Join an existing network

If the device elects to join an existing network, it configures its Channel, PAN ID, XPAN ID, and Network Name to match that of the target network via Thread Commissioning, then goes through the MLE Attach process to attach as a Child (End Device). This process is used for Child-Parent links.

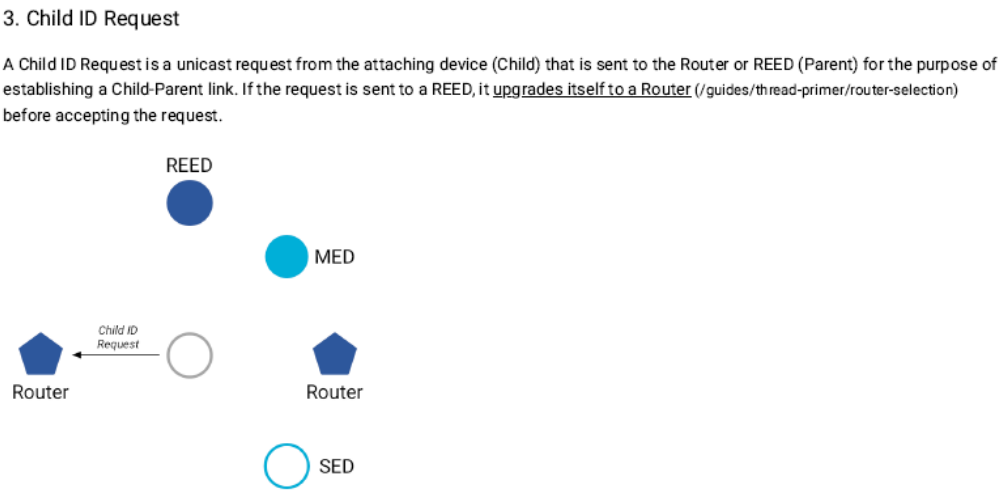
Ex. 35, "Network Discovery and Formation," (available at https://openthread.io/guides/thread-primer/network-discovery (accessed May 20, 2025) and incorporated by reference). As the first step in this process, an attaching device sends a Parent Request message to find neighboring devices that are part of the Thread network.

1. Parent Request

A Parent Request is a multicast request from the attaching device that is used to discover neighboring Routers and Router Eligible End Devices (REEDs) in the target network.



Id. Upon receiving a response, the Thread device sends a message called the Child ID Request to a particular device in the Thread network for the purpose of establishing a link with that device to join the network.



Id. The attaching device may also be referred to as the Child, and the router with which the attaching device is seeking to establish a link may be referred to as the Parent.

148. Accordingly, the Accused Mesh Networking Instrumentalities meet Limitation B of Claim 11.

149. Limitation C of Claim 11 of the Asserted Mesh Networking Patent requires the processor circuit being configured to switch to an associated state upon reception at the receiver arrangement of an association grant in response to one of the association request messages.

150. Thread devices, such as the Accused Mesh Networking Instrumentalities, become attached to a Thread network through the MLE Attach process. As described above in connection with Limitation B, the attaching device begins the process by sending a Parent Request and then exchanges further messages in order to establish a link with a device in the existing Thread network.

Join an existing network

If the device elects to join an existing network, it configures its Channel, PAN ID, XPAN ID, and Network Name to match that of the target network via Thread Commissioning, then goes through the MLE Attach process to attach as a Child (End Device). This process is used for Child-Parent links.

Key Point: Every device, router-capable or not, initially attaches to a Thread network as a Child (End Device).

1. The Child sends a multicast Parent Request (#1_parent_request) to all neighboring Routers and REEDs in the target network.
2. All neighboring Routers and REEDs (if the Parent Request Scan Mask includes REEDs) send Parent Responses (#2_parent_response) with information about themselves.
3. The Child chooses a Parent device and sends a Child ID Request (#3_child_id_request) to it.
4. The Parent sends a Child ID Response (#4_child_id_response) to confirm link establishment.

Id. When the link has been established, the Thread device has attached to the network and thus is in an Attached state. The establishment of the link between the attaching device and the Thread device in the existing Thread network is confirmed by the Child ID Response.

4. Child ID Response

A Child ID Response is a unicast response from the Parent that is sent to the Child to confirm that a Child-Parent link has been established.

Id. Notably, Thread devices such as the Accused Mesh Networking Instrumentalities may exchange information about how long they have been in certain states or roles, suggesting that these devices track their current state/role and when each state/role is entered.

mChildRole (#structot_network_diag_mle_counters_1a33a2daf397cc144009b36dacdaa3c5de)	**uint16_t** Number of times device child role.
mChildTime (#structot_network_diag_mle_counters_1a01062bd8384fe1e360015373a45c68f1)	**uint64_t** Milliseconds device has child role.
mDetachedRole (#structot_network_diag_mle_counters_1a6d1b17ab4d3469693ff19fb596de7661)	**uint16_t** Number of times device detached role.
mDetachedTime (#structot_network_diag_mle_counters_1a75fde6d44ca4db581050d6a71a31bbee)	**uint64_t** Milliseconds device has detached role.

Ex. 36, "otNetworkDiagMleCounters," (available at https://openthread.io/reference/struct/ot-network-diag-mle-counters (accessed May 23, 2025) and incorporated by reference).

151. Accordingly, the Accused Mesh Networking Instrumentalities meet Limitation C of Claim 11.

152. Limitation D of Claim 11 of the Accused Mesh Networking Instrumentalities requires the processor circuit being configured to establish an associated route for communication with an association unit during subsequent communication of operating messages to an association unit.

153. Each of the Accused Mesh Networking Instrumentalities is configured to operate in a Thread network. A Thread network includes a device designated as the Leader.

> Thread Leader
>
> The Thread Leader is a Router that is responsible for managing the set of Routers in a Thread network. It is dynamically self-elected for fault tolerance, and aggregates and distributes network-wide configuration information.

Ex. 37, "Node Roles and Types," (available at https://openthread.io/guides/thread-primer/node-roles-and-types (accessed May 20, 2025) and incorporated by reference). The processors of the Accused Mesh Networking Devices are configured to store information about the route between the attaching device and the Leader as further discussed below in connection with Limitation E of Claim 11.

154. Accordingly, the Accused Mesh Networking Instrumentalities meet Limitation D of Claim 11.

155. Limitation E of Claim 11 requires the associated route includes a source of the association grant and, if the source of the association grant is not the association unit, the operating route between the source of the association grant and the association unit previously established

for the source of the association grant in response to association request messages and/or association grants for intermediate stations.

156. Upon joining a Thread network, Thread devices such as the Accused Mesh Networking Devices receive a Child ID Response which provides information about the route between the Leader and the Thread device. For example, the Child ID Response includes the Source Address (i.e., the address for the Parent Thread device), the Leader RLOC (i.e., the address for the Leader of the Thread network), Network Data such as "more-specific routes," and Route information.

Child ID Response Message Contents

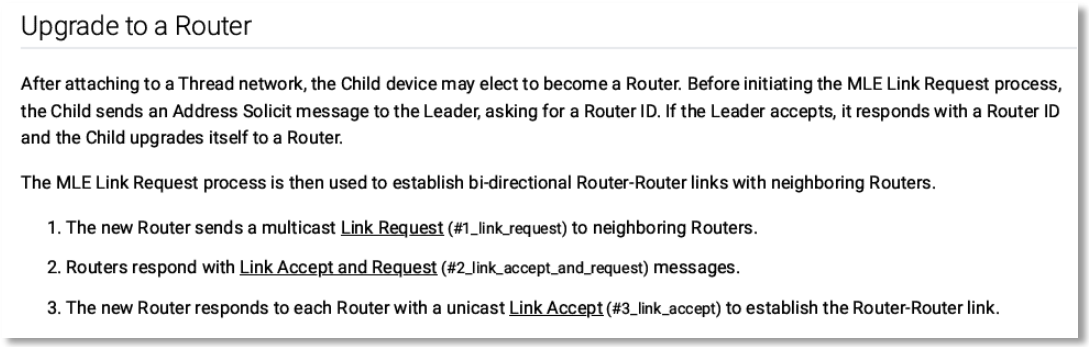
Source Address	Parent's RLOC16
Address16	Child's RLOC16
Leader Data	Information about the Parent's Leader (RLOC, Partition ID, Partition weight)
Network Data	Information about the Thread network (on-mesh prefixes, address autoconfiguration, more-specific routes)
Route (REED only)	Route propagation
Timeout	Inactivity duration before the Parent removes the Child
Address Registration (MEDs and SEDs only)	Confirm registered addresses

Ex. 35. K.Mizra further expects that discovery, including review of source code, will provide further information and confirm this understanding.

157. Accordingly, the Accused Mesh Networking Instrumentalities meet Limitation D of Claim 11.

158. Limitation E of Claim 11 of the Asserted Mesh Networking Patent requires the processor circuit being configured to cause the transmitter arrangement to initiate and transmit association grants in response to association request messages, but only after switching to the associated state.

159. Thread devices such as the Accused Mesh Networking Instrumentalities can send Parent Response and Child ID Response communications once attached to the Thread network, but will not send such responses if the device is disconnected from the network. For example, upon joining a Thread network, Router Eligible End Devices (or REEDs) can upgrade to become routers.



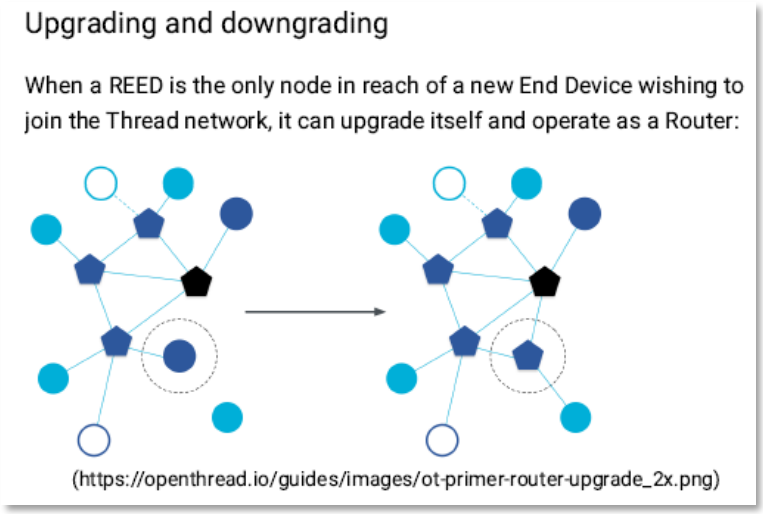
Upgrade to a Router

After attaching to a Thread network, the Child device may elect to become a Router. Before initiating the MLE Link Request process, the Child sends an Address Solicit message to the Leader, asking for a Router ID. If the Leader accepts, it responds with a Router ID and the Child upgrades itself to a Router.

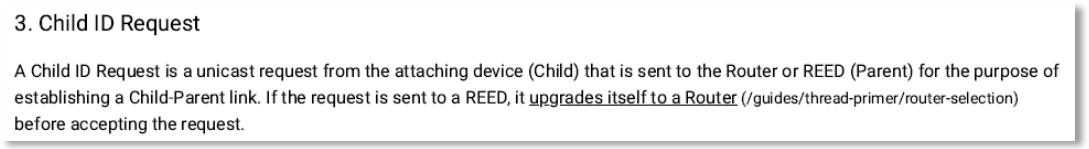
The MLE Link Request process is then used to establish bi-directional Router-Router links with neighboring Routers.

1. The new Router sends a multicast Link Request (#1_link_request) to neighboring Routers.
2. Routers respond with Link Accept and Request (#2_link_accept_and_request) messages.
3. The new Router responds to each Router with a unicast Link Accept (#3_link_accept) to establish the Router-Router link.

Ex. 38, "Router Selection," (available at https://openthread.io/guides/thread-primer/router-selection (accessed May 27, 2025 and incorporated by reference).



Ex. 37. Upon upgrading to a router, the Thread device is capable of accepting a request from an attaching device to join the Thread network.


3. Child ID Request

A Child ID Request is a unicast request from the attaching device (Child) that is sent to the Router or REED (Parent) for the purpose of establishing a Child-Parent link. If the request is sent to a REED, it upgrades itself to a Router (/guides/thread-primer/router-selection) before accepting the request.

Ex. 35.

160. Accordingly, the Accused Mesh Networking Instrumentalities meet limitation of the Limitation E of Claim 11.

161. Google's acts of infringement have occurred within this District and elsewhere throughout the United States.

162. As a result of Google's infringing conduct, K.Mizra has suffered damages. Google is liable to K.Mizra in an amount that adequately compensates K.Mizra for Google's infringement in an amount that is no less than a to-be-calculated fully paid-up, lump-sum, reasonable royalty, together with interest and costs as fixed by this Court under 25 U.S.C. § 284.

VIII. REQUEST FOR RELIEF

WHEREFORE, K.Mizra respectfully requests the Court find in its favor and against Google, and that the Court grant K.Mizra at least the following relief:

A. Judgment that Google has infringed, literally and/or under the doctrine of equivalents, one or more claims of the Asserted Patents;

B. Awarding damages to K.Mizra in an amount to be proven at trial and in the form of a fully paid-up, lump sum, reasonable royalty that considers and runs through expiration of all the Asserted Patents;

C. Awarding enhanced damages, as appropriate, under 35 U.S.C. § 284;

D. Awarding K.Mizra's costs (including internal and external costs and disbursements) and declaring this an exceptional case and awarding K.Mizra its attorneys' fees in accordance with 35 U.S.C. § 285;

E. Pre-judgment and post-judgment interest at the maximum rate permitted by law on the damages caused by reason of Google's infringing activities and other conduct complained of herein; and

F. Awarding such other and further relief as this Court deems just and proper under the circumstances.

IX. DEMAND FOR JURY TRIAL

Pursuant to Fed. R. Civ. P. 38(b), K.Mizra hereby demands a trial by jury on all issues so triable.

Dated: _____

Respectfully submitted,

By: /s/ DRAFT

Michael C. Smith
Texas Bar No. 18650410
michael.smith@solidcounsel.com
Scheef & Stone, LLP
113 E. Austin Street
Marshall, TX 75670
(903) 938-8900

Robert R. Brunelli (admitted *pro hac vice*)
CO State Bar No. 20070
rbrunelli@sheridanross.com
Bart A. Starr (admitted *pro hac vice*)
CO State Bar No. 50446
bstarr@sheridanross.com
Brian S. Boerman (admitted *pro hac vice*)
CO State Bar No. 50834
bboerman@sheridanross.com
Gene Volchenko (admitted *pro hac vice*)
IL State Bar No. 6342818
gvolchenko@sheridanross.com
SHERIDAN ROSS P.C.
1560 Broadway, Suite 1200
Denver, CO 80202
Telephone: 303-863-9700
Facsimile: 303-863-0223
litigation@sheridanross.com

Of Counsel:
Claire A. Henry
Texas Bar No. 24053063
Miller Fair Henry PLLC
1507 Bill Owens Parkway
Longview, TX 75604
Telephone: (903) 757-6400
Facsimile: (903) 757-2323
claire@millerfairhenry.com

Attorneys for Plaintiff K.Mizra LLC

CERTIFICATE OF SERVICE

I hereby certify that a true and correct copy of the above and foregoing document has been served on _____ upon all counsel of record who are deemed to have consented to electronic service via the Court's CM/ECF system.

/s/ DRAFT
Lori R. Brown
lbrown@sheridanross.com
SHERIDAN ROSS P.C.
1560 Broadway, Suite 1200
Denver, CO 80202
Telephone: 303-863-9700
Facsimile: 303-863-0223

EXHIBIT 5

DRAFT
5/28/2025

IN THE UNITED STATES DISTRICT COURT
FOR THE WESTERN DISTRICT OF TEXAS
AUSTIN DIVISION

K.MIZRA LLC,

Plaintiff,

v.

GOOGLE LLC,

Defendant.

Civil Action No.: 1:25-cv-00236-ADA

Jury Trial Demanded

FIRST AMENDED COMPLAINT FOR PATENT INFRINGEMENT

Plaintiff K.Mizra LLC ("K.Mizra") files this First Amended Complaint for patent infringement against Defendant Google LLC ("Google"), alleging as follows:

K.Mizra LLC, Ex. 2001, p. 118
Google LLC v. K.Mizra LLC
IPR2025-01437

TABLE OF CONTENTS

INDEX OF EXHIBITS iii

I. INTRODUCTION 1

A. The Asserted Zero Trust Network Security Patent 1

B. The Asserted Hyperparameter Tuning Patent 3

C. The Asserted Mesh Networking Patent 4

D. K.Mizra's Licensing Model 4

II. THE PARTIES 4

III. JURISDICTION AND VENUE 5

IV. GENERAL ALLEGATIONS 6

A. The Zero Trust Network Security Patent and Google's Infringement 6

1. Prior Licensing and Litigation Concerning the Asserted Zero Trust Network Security Patent 6

2. Computer Network Security Problems In 2004 Solved by the Asserted Zero Trust Network Security Patents 7

3. K.Mizra's Asserted Zero Trust Network Security Patent Claims Are Presumed Valid 9

4. Failed IPRs Brought Against the Asserted Zero Trust Network Security Patents 15

5. Google's Infringing Chrome Enterprise Premium Product 16

B. The Asserted Hyperparameter Tuning Patent 19

1. Prior Licensing and Litigation Concerning the Hyperparameter Tuning Patent 19

2. Problems in 2007 Solved By the Asserted Hyperparameter Tuning Patent 20

3. K.Mizra's Asserted Hyperparameter Tuning Patent Claims Are Presumed Valid 24

4. Google's Infringing Vertex AI Studio Product 25

K.Mizra LLC, Ex. 2001, p. 119
Google LLC v. K.Mizra LLC
IPR2025-01437

C. The Asserted Mesh Networking Patent 26

1. Prior Licensing and Litigation Concerning the Mesh Networking Patent 27

2. Problems in 2007 Solved By the Mesh Networking Patent 28

3. K.Mizra's Asserted Mesh Networking Patent Claims Are Presumed Valid 29

4. Google's Infringing Nest Devices 30

V. FIRST CLAIM FOR RELIEF (Patent Infringement Under 35 U.S.C. § 271 of the '705 Patent) 35

VI. SECOND CLAIM FOR RELIEF (Patent Infringement Under 35 U.S.C. § 271 of the '120 Patent) 56

VII. THIRD CLAIM FOR RELIEF (Patent Infringement Under 35 U.S.C. § 271 of the '717 Patent) 69

VIII. REQUEST FOR RELIEF 81

IX. DEMAND FOR JURY TRIAL 82

K.Mizra LLC, Ex. 2001, p. 120
Google LLC v. K.Mizra LLC
IPR2025-01437

INDEX OF EXHIBITS

Ex	Description
1	U.S. Patent No. 8,234,705
2	U.S. Patent No. 8,234,705 – Preliminary Infringement Contention Claim Chart
3	Next '24_ Introducing Chrome Enterprise Premium, the future of endpoint security _ Google Cloud Blog
4	Chrome Enterprise Premium access protection overview _ BeyondCorp Enterprise Google Cloud
5	BeyondCorp Zero Trust Enterprise Security _ Google Cloud
6	U.S. Patent No. 8,438,120
7	U.S. Patent No. 8,144,717
8	Thread Play Services APIs Matter Google Home Developers
9	Connect Thread smart devices with Google TV Streamer (4K) - Android - Google TV Help
10	Nest Connect - Range Extender for Detect & Guard - Google Store
11	Nest debuts Nest Secure home system and has a Thread router called Nest Connect - Stacey on IoT Internet of Things news and analysis
12	Nest × Yale Lock - Key-Free Smart Deadbolt - Google Store_1
13	Pixel 9 series arrives at the FCC, includes Thread
14	Chrome Enterprise Premium documentation
15	Endpoint Verification Overview
16	Chrome Enterprise Premium Overview
17	How to prevent account takeovers with new certificate-based access
18	Understand Mutual TLS at Google Cloud
19	Deploy Endpoint Verification to use with certificate-based access
20	New Google API Will Securely Verify Chrome Devices
21	Chrome Verified Access
22	Context Aware Access insights and recommendations are now generally available
23	Block compromised mobile devices using context-aware access
24	Allow users to unblock apps with remediation messages in Context Aware Access
25	Vertex AI Platform Google Cloud
26	Vertex AI custom training overview Google Cloud
27	Vertex AI documentation Google Cloud
28	Overview of hyperparameter tuning Vertex AI Google Cloud
29	Create a hyperparameter tuning job for python package Vertex AI Google Cloud
30	Intro to Model Tuning: Grid and Random Search (excerpts)
31	Comparing Randomized Search and Grid Search for Hyperparameter Estimation in Scikit Learn GeeksforGeeks
32	OpenThread Homepage
33	Nest Hub Smart Displays for your Home - Google Store
34	Developing with OpenThread APIs
35	Network Discovery and Formation OpenThread
36	otNetworkDiagMleCounters Struct Reference OpenThread
37	Node Roles and Types OpenThread
38	https://openthread.io/guides/thread-primer/router-selection

K.Mizra LLC, Ex. 2001, p. 121
Google LLC v. K.Mizra LLC
IPR2025-01437

I. INTRODUCTION

1. K.Mizra is a patent licensing company run by experienced management. The company focuses on high-value, high-quality patents and owns patent portfolios originating from a wide array of inventors, including patents and patent portfolios developed by well-known multinational corporations such as IBM, Intel, Rambus and others, as well as from research institutes such as Nederlandse Organisatie voor Toegespast Natuurwetenschappelijk Onderzoek (Netherlands Organization for Applied Scientific Research). By focusing on high-quality patents, K.Mizra provides a secondary market that enables inventors to recoup their research and development investments and to continue their innovations. K.Mizra offers licenses to its patents on reasonable terms and, in this way, plays an important role in the development of technologies that improve businesses and lives.

A. The Asserted Zero Trust Network Security ~~Patents~~Patent

2. K.Mizra is the owner by assignment of United States Patent ~~Nos~~No. 8,234,705 ("the '705 Patent") ~~and 9,516,048~~ ("the ~~'048 Patent") (collectively, "the~~ Asserted Zero Trust Network Security ~~Patents"). These Patents were~~Patent"). The Patent wase involved in unsuccessful *Inter Partes* Review ~~proceedings~~proceeding ("IPRs"), several now-resolved federal court litigations, and ~~were~~was originally invented by two highly-respected and prolific inventors—James A. Roskind and Aaron T. Emigh.

3. The Asserted Zero Trust Network Security ~~Patents were~~Patent was originally owned by Dr. Roskind's and Mr. Emigh's company, Radix Labs, LLC. Dr. Roskind and Mr. Emigh were then, and remain today, focused on innovation, conducting new research, developing new technologies, and creating new and innovative computer products and systems.

K.Mizra LLC, Ex. 2001, p. 122
Google LLC v. K.Mizra LLC
IPR2025-01437

4. Dr. Roskind, a co-inventor of these Patents, has bachelors, masters, and doctorate degrees from MIT in both electrical engineering and computer science and is a named inventor of over 300 U.S. patents. He has worked for Netscape as the Chief Architect and as the Netcenter Security Architect and was a co-founder of Infoseek, a company acquired by Disney for $770 million. Dr. Roskind also was a key developer of Google's "transport protocol" that provides the tech giant billions of dollars in value every year.

5. Mr. Emigh, also a co-inventor of the Asserted Zero Trust Network Security Patents, graduated from the University of California, Santa Cruz with degrees in linguistics and computer and information sciences and is a named inventor of over 140 patents. Before working with Dr. Roskind, Mr. Emigh worked in various software-development positions, including software manager, architect, and engineer for Unicom and manager for the software development and technical marketing groups of Philips TriMedia. He has founded or co-founded many companies in addition to Radix Labs, LLC, including CommerceFlow, Inc., which was acquired by eBay for the technology that Mr. Emigh helped develop.

6. After the Asserted ~~Patents were~~Zero Trust Network Security Patent was issued, Dr. Roskind and Mr. Emigh recouped their research and development investment by selling certain technology rights and continued to work independently on their individual technological pursuits. K.Mizra ultimately acquired the Asserted Zero Trust Network Security ~~Patents~~Patent and licensed ~~them~~it to many major companies operating in the computer technology space. Some of those companies, including previously accused infringers, chose to test the validity of the Asserted Zero Trust Network Security ~~Patents~~Patent before ~~settling their lawsuits involving those Patents.~~ accepting a license to the Patent.

K.Mizra LLC, Ex. 2001, p. 123
Google LLC v. K.Mizra LLC
IPR2025-01437

~~6.~~7. For example, accused infringers of the Asserted Zero Trust Network Security ~~Patents~~Patent previously sought IPRs by the Patent Trial and Appeal Board ("PTAB") of both of the Asserted Zero Trust Network Security Patents. A Final Written Decision ("Decision") issued in the IPR for the '705 Patent concluded that the petitioners had not shown by a preponderance of the evidence that the claims at issue were unpatentable. ~~Based on the results of the '705 Patent IPR, an IPR for the similar '048 Patent was never instituted.~~ The '705 Patent IPR Decision was appealed to the U.S. Court of Appeals for the Federal Circuit ("CAFC"), resulting in a procedurally-focused remand back to the PTAB. The PTAB recently dismissed the IPR Petition with prejudice.

~~7.~~8. The Asserted ~~Patents were~~Zero Trust Network Security Patent was previously asserted by K.Mizra in this District against Cisco Systems, Inc. in *K.Mizra v. Cisco Systems, Inc*. (Civil Action No. 6:20-cv-01031-ADA), a case assigned to Judge Albright which resolved in September 2024, only a few weeks prior to trial being scheduled to commence. In that action, Judge Albright not only conducted a claim construction hearing and issued several important constructions of disputed claim terms, he also ruled on numerous summary judgment and *Daubert* motions as part of a final pretrial conference. As such, Judge Albright is well-versed in the technology involved with the Asserted Zero Trust Network Security ~~Patents~~Patent, K.Mizra and other issues here likely to be involved.

B. The Asserted Hyperparameter Tuning Patent

9. K.Mizra is also the owner by assignment of United States Patent No. 8,438,120 ("the '120 Patent" or "the Asserted Hyperparameter Tuning Patent"). The '120 Patent has not previously been asserted or involved in any post issuance proceedings and was invented by Stephan Alexander Raaijmakers of the Nederlandse Organisatie Voor Toegepast-

K.Mizra LLC, Ex. 2001, p. 124
Google LLC v. K.Mizra LLC
IPR2025-01437

Natuurwetenschappelijk Onderzoek TNO, a well-known and widely respected technology research institution ("TNO").

C. The Asserted Mesh Networking Patent

10. K.Mizra is also the owner by assignment of United States Patent No. 8,144,717 ("the '717 Patent" or "the Mesh Networking Patent"). The '717 Patent has also not previously been asserted or involved in any post issuance proceedings and was invented by Marinus Johannes Blange and Miodrag Djurica of the TNO.

D. K.Mizra's Licensing Model

~~8.~~11. The '705 Patent, '120 Patent and '717 Patents are collectively referred herein as "the Asserted Patents". K.Mizra remains ready, willing, and able to provide commercially- reasonable licenses for its various patented technologies, including its Asserted Patents, to all entities who wish or need to use its covered technologies internally or in connection with products or services offered to others. As outlined below, Google is one such entity.

II. THE PARTIES

~~9.~~12. K.Mizra is a Delaware limited liability company with a mailing address of 777 Brickell Avenue, #500-96031, Miami, Florida 33131, and operates in Florida.

~~10.~~13. K.Mizra is the owner by assignment of the Asserted Patents.

~~11.~~14. Upon information and belief, Google is a corporation organized and existing under the laws of the state of Delaware, with a principal place of business at 1600 Amphitheatre Parkway, Mountain View, California 94043.

~~12.~~15. Upon information and belief, Google maintains an office in this District at 500 West Second Street, Austin, Texas 78701.

K.Mizra LLC, Ex. 2001, p. 125
Google LLC v. K.Mizra LLC
IPR2025-01437

~~13.~~16. Upon information and belief, Google employs hundreds of people in its Austin, Texas office, including persons with knowledge of the Google products and technology at issue in this case.

~~14.~~17. Upon information and belief, Google is also registered to do business in Texas and may be served with process by serving Corporation Service Company, Google's registered agent in Texas located at 211 East Seventh Street, Suite 620, Austin, Texas 78701-3218.

III. JURISDICTION AND VENUE

~~15.~~18. This is an action for patent infringement under the patent laws of the United States, 35 U.S.C. §§ 1 *et seq.*, including 35 U.S.C. §§ 271, 281, and 284, among others. The Court has subject-matter jurisdiction over the claims raised in this action pursuant to 28 U.S.C. §§ 1331 and 1338(a).

~~16.~~19. This Court has personal jurisdiction over Google by virtue of, *inter alia*, its conduct of business in this District, its registration to do business in Texas, its appointment of a registered agent in Texas, its employment of hundreds of persons in its Austin office located in this District, and its substantial, continuous, and systematic contacts with the state of Texas and this District. Google intentionally markets and sells its infringing products directly and through agents to residents of Texas, enjoys substantial income from its business activities in the state of Texas and this District, and/or directly, by its own actions, and/or in combination with actions of customers and others under its control, has committed acts of patent infringement in this District at least by selling infringing products in this District.

K.Mizra LLC, Ex. 2001, p. 126
Google LLC v. K.Mizra LLC
IPR2025-01437

IV. GENERAL ALLEGATIONS

A. The ~~Asserted~~ Zero Trust Network Security ~~Patents~~Patent and Google's Infringement

~~17.~~20. K.Mizra is the sole owner by assignment of the Asserted Zero Trust Network Security ~~Patents~~Patent with the full and exclusive right to ~~bring suit to~~ enforce ~~them~~the Patent. K.Mizra is also entitled to sue to collect damages for all past infringement of the Asserted Zero Trust Network Security ~~Patents~~Patent.

~~18.~~21. The '705 Patent, titled "Contagion Isolation and Inoculation," was issued by the USPTO to Inventors Roskind and Emigh on July 31, 2012. A true and correct copy of the '705 Patent is attached hereto as Exhibit 1 and incorporated by reference.

~~19. The '048 Patent, titled "Contagion Isolation and Inoculation Via Quarantine," was issued by the USPTO to Inventors Emigh and Roskind on December 6, 2016. A true and correct copy of the '048 Patent is attached hereto as Exhibit 2 and incorporated by reference.~~

~~20.~~22. The Asserted Zero Trust Network Security ~~Patents share similar (and in some respects, identical) specifications and~~Patent claims~~, with both claiming~~ priority to U.S. Provisional Application No. 60/613,909, filed on September 27, 2004 (the "Provisional Application~~").~~") and is entitled to that priority date.

1. Prior Licensing ~~And~~and Litigation ~~Of The~~Concerning the Asserted ~~Patents~~Zero Trust Network Security Patent

~~21.~~23. The Asserted Zero Trust Network Security ~~Patents have~~Patent has been owned by several entities, in addition to Radix and K.Mizra, with some of those entities issuing to third parties certain rights to the technologies covered thereby.

~~22.~~24. K.Mizra has been involved in ~~a number of~~several actions it was required to institute to protect its patent rights, including actions involving the Asserted Zero Trust Network Security

K.Mizra LLC, Ex. 2001, p. 127
Google LLC v. K.Mizra LLC
IPR2025-01437

~~Patents~~Patent. Most of those actions resulted in the execution of confidential patent license agreements.

~~23.~~25. Google is not and has never been a licensee of the Asserted Zero Trust Network Security ~~Patents~~Patent nor had ~~or has~~ any rights to use technologies covered by ~~them~~the Patent. Google thus has no ~~ownership or other rights~~right (and is entitled to no ~~rights) relating to~~right) to use technology covered by the Asserted Zero Trust Network Security ~~Patents~~Patent.

2. Computer Network Security Problems In 2004 Solved ~~By The~~by the Asserted Zero Trust Network Security Patents

26. The technology described in the Asserted Zero Trust Network Security ~~Patents~~Patent was invented by Dr. Roskind and Mr. Emigh, two colleagues living in the same area who had similar interests in innovating computer-related technologies. In 2003, the inventors decided to create a business—Radix Labs, LLC—which focused on developing intellectual property related to various computer technologies, including computer network security technologies. The inventors focused on conceiving and reducing to practice inventions that they knew were needed (or soon would be needed) in the computer networking industry and then on drafting patent applications to capture and protect their technological innovations.

~~24.~~27. In September of 2004, the inventors filed the Provisional Application to which ~~both~~the Asserted Zero Trust Network Security ~~Patents claim~~Patent claims priority. The Provisional Application described technology that focused on securing a computer network against the threats to which it was exposed when computer endpoints (e.g., laptop computers~~)~~, smart phones, tablets, etc.,) were connected to a computer network. The Provisional Application, and by natural extension the Asserted Zero Trust Network Security ~~Patents~~Patent, also focus on remedying ~~identified threats~~ and/or quarantining those threats to mitigate any damage to the secured network.

K.Mizra LLC, Ex. 2001, p. 128
Google LLC v. K.Mizra LLC
IPR2025-01437

~~25.~~28. Claims of the Asserted Zero Trust Network Security ~~Patents~~Patent are directed to technological solutions that address specific challenges grounded in computer network security. Maintaining the security of computer systems and networks is a tremendous concern for modern enterprises, since a breach of an internal network can have severe repercussions, including major financial losses, data theft, disclosure of sensitive information, network disruptions, data corruption, etc. The inventors of the Asserted Zero Trust Network Security ~~Patents~~Patent understood that while a network security appliance or hardware can be adept at keeping out unwanted external intrusions from the network, the most exploitable vulnerabilities of most networks are the end-user computers that roam throughout various public and private network domains, potentially exposing those computers to infection and then accessing and potentially infecting the entire and presumably secure computer network.

~~26.~~29. For example, the '705 Patent explains that

> Laptop and wireless computers and other mobile systems pose a threat to elements comprising and/or connected to a network service provider, enterprise, or other protected network to which they reconnect after a period of connection to one or more networks and/or systems that are not part of the service provider, enterprise, or other protected network. By roaming to unknown domains, such as the Internet, and/or connecting to such domains through public, wireless, and/or otherwise less secure access nodes, such mobile systems may become infected by computer viruses, worms, backdoors, and/or countless other threats and/or exploits and/or have unauthorized software installed; have software installed on the mobile system by an operator of the protected network for the protection of the mobile system and/or the protected network removed or altered without authorization and/or have configurations, settings, security data, and/or other data added, removed, and/or changed in authorized ways and/or by unauthorized person.

See, e.g., Ex. 1 at 1:14–31.

~~27.~~30. The solution to these problems—as specified and claimed in the Asserted Zero Trust Network Security ~~Patents~~Patent—was an advanced departure from the conventional network

K.Mizra LLC, Ex. 2001, p. 129
Google LLC v. K.Mizra LLC
IPR2025-01437

access control solutions then in use and was then, as it remains today, patent eligible, highly valuable, novel, and non-obvious technology.

3. K.Mizra's Asserted Zero Trust Network Security Patent Claims Are Presumed Valid

~~28.~~31. K.Mizra asserts that ~~at least,~~Claims 12, 13, 16 and ~~without limitation, Claim~~ 19 of the '705 Patent ~~and Claim 17 of the '048 Patent~~ have been directly infringed, either literally or under the doctrine of equivalents~~. K.Mizra reserves the right to assert additional claims of the Asserted Zero Trust Network Security Patents, including both independent and dependent claims, pursuant to the Court's (and other applicable) rules and procedures and as discovery progresses.~~ by Google. These claims are referred to herein as the "Asserted Zero Trust Network Security Claims."

~~29.~~32. None of the Asserted Zero Trust Network Security Claims are directed to abstract ideas, and each employs inventive concepts and is directed to patent-eligible subject matter. All claims of the Asserted Zero Trust Network Security ~~Patents~~Patent are also presumed to be valid and enforceable against Google and others.

~~30.~~33. Indeed, the Asserted Zero Trust Network Security ~~Patents' similar common~~Patent's specification and claims demonstrate that the need satisfied by the inventions of the Asserted Zero Trust Network Security Claims was long-felt in the industry and thus unconventional. As one example, the ~~'048~~'705 Patent ~~provides~~explains that ~~"[u]nwanted~~mobile end user devices such as laptops and wireless computers pose a threat to protected network elements because those devices may access unsecure systems and thereby "become infected by computer viruses, worms, backdoors, and/or countless other threats and/or exploits and/or have unauthorized software installed; have software installed on the mobile system by an operator of the protected network for the protection of the mobile system and/or the protected network removed or altered without

K.Mizra LLC, Ex. 2001, p. 130
Google LLC v. K.Mizra LLC
IPR2025-01437

authorization; and/or have configurations, settings, security data, and/or other data added, removed, and/or changed in unauthorized ways and/or by unauthorized person[s]." Ex. 1 at 1:23-31. Similarly, stationary systems such as desktop computers "may become infected, e.g., due to receipt and execution of malicious ~~network communications, such as spam, phishing, work propagation, etc., hamper productivity and the use and enjoyment of computer and network~~ code via a network or other communication and/or a diskette and/or other removable media." Ex. 1 at 1:31-34. This poses a danger to a protected network because, when the user device connects to the protected network, "a system may infect or otherwise harm resources ~~by end users, burden affected networks~~associated with ~~unauthorized~~the protected network before measures can be taken to detect and~~/or undesired traffic, and expose recipients to the risk~~ prevent the spread of ~~theft, fraud, etc."~~such infections or harm. Ex. ~~2~~1 at 1:~~22–27~~34-38. The Asserted Zero Trust Network Security ~~Patents'~~Patent's specification further provides that "[t]herefore, there is a need for ~~an effective~~a reliable way to ~~intercept and take corrective action with respect to unauthorized, unwanted, and/or otherwise malicious electronic mail and/or other network communications~~ensure that ~~better protects the network and provides protection to destination hosts that are~~a system does not ~~protected by destination or destination mail or messaging server-based filtering software~~infect or otherwise harm other network resources when connected to a protected network." *Id.* at 1:~~46–52~~38-41.

~~31.~~34. The specification (including the provisions quoted above), the figures (including those included below), and the text related to the figures further illustrate the complex, tiered network system architecture of the inventions captured by the Asserted Zero Trust Network Security Claims. These figures include the following:

K.Mizra LLC, Ex. 2001, p. 131
Google LLC v. K.Mizra LLC
IPR2025-01437

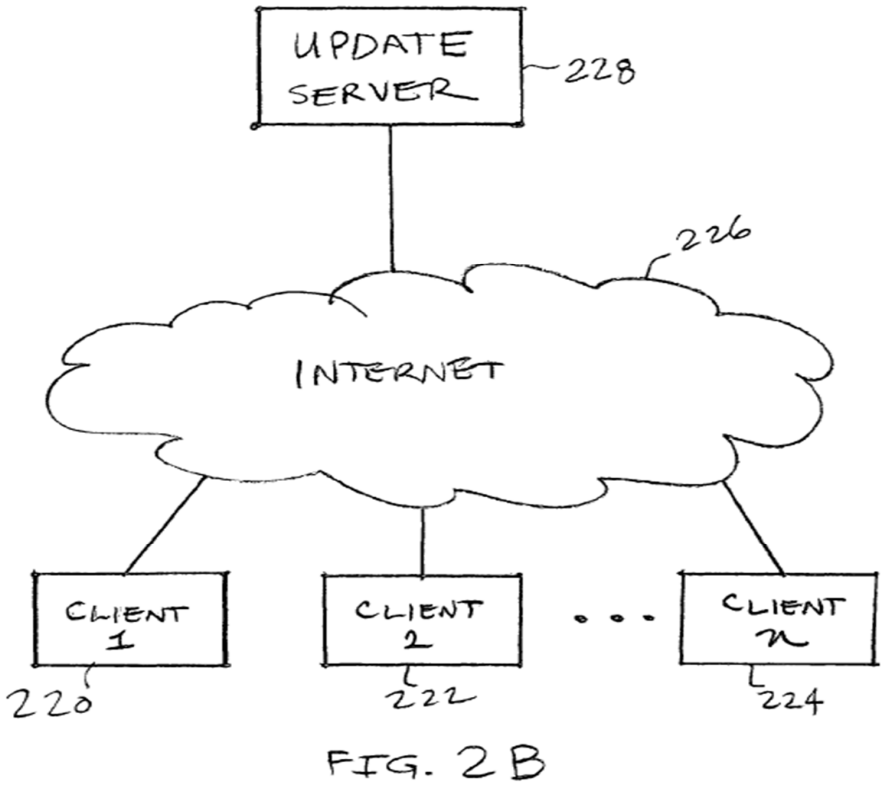
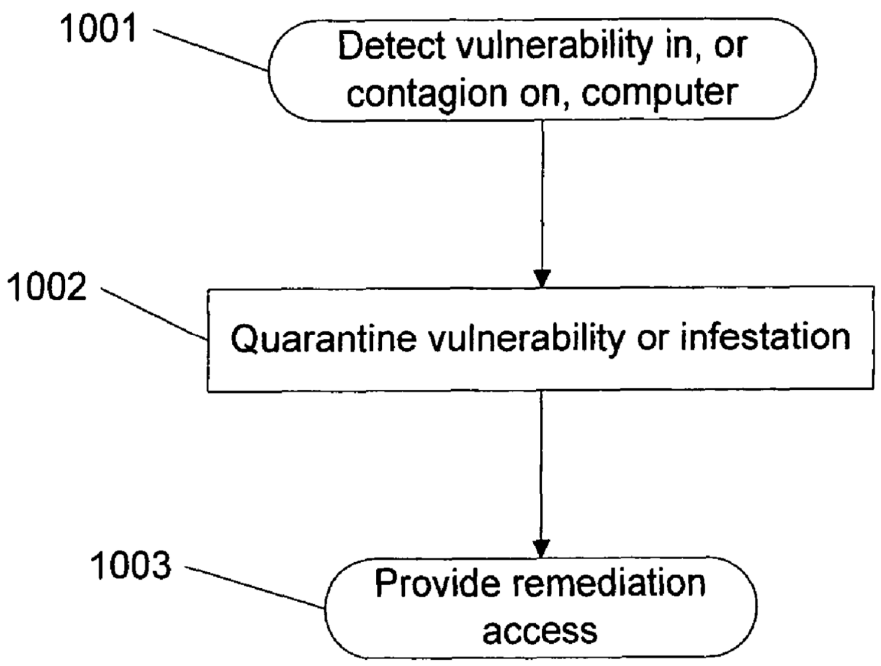


FIG. 2B

See Ex. 1 at Fig. 2B.

K.Mizra LLC, Ex. 2001, p. 132
Google LLC v. K.Mizra LLC
IPR2025-01437



See id. at Fig. 10A.

~~32.~~35. ~~The foregoing demonstrates that the inventions of the~~The Asserted Zero Trust Network Security Claims focus on being able to receive, understand and act upon information provided by specific tamperproof hardware that must interact with unique software to improve network access control technology and protect a secure computer network and the data stored thereon from infected devices. As such, the Asserted Zero Trust Network Security Claims are eligible as a matter of law for patent protection under step one of *Alice Corp. v. CLS Bank Int'l*, 573 U.S. 208, 216 (2014).

~~33.~~36. All actions and steps recited in the Asserted Zero Trust Network Security Claims, including the act of quarantining endpoints or other computers, if necessary, require the involvement of various hardware components running dedicated software both before, during, and after the selection and isolation of an ~~object.~~endpoint device. Said another way, ~~a claim~~claims

K.Mizra LLC, Ex. 2001, p. 133
Google LLC v. K.Mizra LLC
IPR2025-01437

directed to allowing a machine to automatically and dynamically select and isolate an unsafe device attempting to access a secure network is not simply adding a generic computer component to a fundamentally human process. Rather, it is removing the once-necessary human intervention from a fundamentally mechanical process, an "improvement in the functioning of a" networked system that simply cannot be considered directed to an abstract concept. *Enfish LLC v. Microsoft Corp.*, 822 F.3d 1327, 1339 (Fed. Cir. 2016). These technological acts were unconventional and unknown in the art existing in 2004.

37. As the specification confirms, the improvement captured by the Asserted Zero Trust Network Security Claims ~~are~~is not simply quarantining an infected device, but it is instead a multi-faceted network system involving multiple interrelated software and hardware components to protect a network from known and unknown threats. Specifically, the ~~similar specifications of the~~ Asserted ~~Patents disclose~~Zero Trust Network Security Patent discloses that to reduce the burdens of having to manually identify, connect to, isolate, and remove malicious software from an infected ~~device~~endpoint, the networked system can direct an unclean endpoint (referred to in the Patent as a "host computer") attempting to connect to the secure network~~, known as the host computer,~~ to a form of remediation, such as downloading a software patch or a software update, removing material from ~~the host computer and/or~~, enabling certain settings, etc. present on the ~~host computer.~~endpoint. *See* Ex. 1 at 1:14–41.

~~34.~~38. Indeed, the inventions of the Asserted Zero Trust Network Security Claims are each tethered to these advances ~~over the art in the 2005 time frame, reciting methods and systems~~, none of which, individually, and certainly not as an ordered combination, was conventional in the computer network security field in 2004. Indeed, these claims recite then-unconventional systems and technology that automatically and dynamically detect an insecure condition by contacting a

K.Mizra LLC, Ex. 2001, p. 134
Google LLC v. K.Mizra LLC
IPR2025-01437

trusted computing base located on an endpoint, receiving a response therefrom, determining if that response contains a valid identification of cleanliness, and configuring and implementing ~~a~~any needed remediation action ~~based on what is discovered about the state of an endpoint or "host" computer.~~upon. *See, e.g.,* Ex. 1, ~~Claims 12 and~~Claim 19~~; Ex. 2, Claims 10 and 17~~. More specifically, the Asserted Zero Trust Network Security Claims require a system to communicate with a "trusted computing base" to determine when a response includes a valid digitally signed attestation of cleanliness, and to control access to the network accordingly. These Asserted Zero Trust Network Security Claims are thus directed to a machine-implemented solution resolving a machine-specific problem, *i.e.* a machine's difficulty in detecting, isolating, and remediating infected endpoint devices (*e.g.*, host computers) to prevent contagion of and damage to the larger computer network. These technological acts were completely unconventional in the relevant field in 2004.

~~35. The Asserted Zero Trust Network Security Claims are thus directed to a machine-implemented process for (1) determining whether the host computer is required to be quarantined, (2) isolating and inoculating the contagions (including directing the host to software programs and/or code designed to identify undesirable and/or unauthorized states) by quarantining the host, (3) limiting access to the network by the host computer so that the unsafe condition thereof can be remedied, and (4) allowing for remediation of an unsafe or infected host computer. As such, the Asserted Zero Trust Network Security Claims recite inventions with specific applications or improvements to technologies in the marketplace and cannot be considered abstract or patent ineligible under relevant law.~~

K.Mizra LLC, Ex. 2001, p. 135
Google LLC v. K.Mizra LLC
IPR2025-01437

4. Failed IPRs Brought Against the Asserted Zero Trust Network Security Patents

~~36.~~39. Fortune 100 companies accused of infringing the Asserted Zero Trust Network Security ~~Patents have~~Patent previously filed petitions for IPRs against ~~each of the Asserted Patents~~it, alleging that the claims ~~of the Asserted Patents~~thereof should be held invalid as either anticipated or obvious considering art not previously considered. Ultimately, the PTAB instituted an IPR against the '705 Patent, with similar third party IPRs that were subsequently filed being joined to the first-filed and instituted IPR.

~~37.~~40. The PTAB eventually issued its decision holding that no claims of the '705 Patent were unpatentable, finding that no asserted prior art reference alone or in combination satisfied the limitation of "providing . . . an IP address of a quarantine server configured to serve the quarantine notification page" that was present in all claims of the '705 Patent.

~~38. Similarly, petitions for IPR were filed against the '048 Patent, but the PTAB denied institution of those, stating that the '048 Patent IPRs were "closely related" to the '705 Patent IPR petitions, that the petitions were based on and cited the same prior art and that "the challenged claims [were] materially the same" as those recited in the '705 patent's claims. The PTAB then offered that it had already "issued a Final Written Decision in [the '705 Patent's IPR], finding no claims of the '705 patent unpatentable" and that the '048 Patent IPR petitions were being denied institution for the same reasons.~~

~~39.~~41. The '705 Patent IPR Decision was then appealed to the CAFC, which reversed the PTAB's Decision on a few narrow procedural issues involving proof that the asserted prior art references would be combined by a person having ordinary skill in the art, as alleged by the petitioners.

K.Mizra LLC, Ex. 2001, p. 136
Google LLC v. K.Mizra LLC
IPR2025-01437

~~40.~~42. ~~A motion to dismiss the IPR was subsequently filed and the IPR has been~~The IPRs have now all dismissed with prejudice by the PTAB.

5. Google's ~~Accused Instrumentalities And Services~~Infringing Chrome Enterprise Premium Product

43. ~~Google has been making, selling, using, and offering for sale~~ Google has been and continues to make, sell, use, offer to sell, import into and/or export out of the United States a product known as Chrome Enterprise Premium ("CEP"), a computer network access and security ~~products, systems, and services that~~product. These actions directly infringe the Asserted Zero Trust Network Security ~~Patents~~Patent in violation of 35 U.S.C. § 271(a). (Attached as Exhibit 2 and incorporated herein by this reference, is a preliminary claim chart explaining Google's infringing conduct.)

44. CEP is a secure enterprise solution that provides integrated threat and data protection and offers scalable context-aware access controls for software-as-a-service (SaaS) and Web Apps, helping to mitigate data exfiltration from secure networks. CEP achieves these goals by being designed to operate within an environment of endpoint that include a trusted computing base ("TCB") with a trusted platform module ("TPM") and a Chrome browser. When such a configured endpoint seeks to enter a protected network, the CEP running on Google's cloud obtains posture information stored in the TCB's TPM of the endpoint. CEP, which integrates several security features and functions that Google offers in CEP and in other products, such as Endpoint Verification, Context-Aware Access, certificate-based access, Verified Access, Policy Remediation, and others, then uses the provided information to make decisions about whether the requesting endpoint should be granted access to the protected network, if it should be remediated, quarantined, etc., and then sends to the endpoint appropriate information, etc.

K.Mizra LLC, Ex. 2001, p. 137
Google LLC v. K.Mizra LLC
IPR2025-01437

45. As Google has explained it:

Browsers are more than just a portal to the Internet: They are the new endpoint where almost every high-value activity and interaction in the enterprise takes place. Authentication, access, communication and collaboration, administration, and even coding are all browser-based activities in the modern enterprise.

Endpoint security is growing more challenging due to remote work, reliance on an extended workforce, and the proliferation of new devices that aren't part of an organization's managed fleet. As these trends continue to accelerate and converge, it's clear that the browser is a natural enforcement point for endpoint security in the modern enterprise. Gartner® research predicts "enterprise browsers will be the core platform for delivering workforce productivity and security software on managed and unmanaged devices for a seamless hybrid work experience."

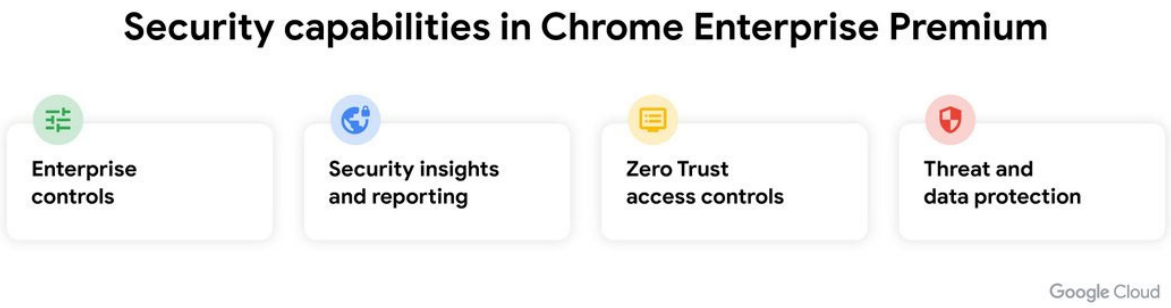
Get advanced security with Chrome Enterprise Premium

Users around the world already use Chrome on their digital devices to access information and get work done safely across Windows, macOS, ChromeOS, Android, and iOS. As a leader in browser security, Chrome has built features and best practices that have disrupted, and then established, industry norms.

For example, we introduced advanced security sandboxing and novel exploit mitigations, and we lead the industry in our response time to zero-day vulnerabilities. We also continue to advance security, and recently announced work that uses AI to do privacy-preserving, real-time checks of websites to help protect users from increasingly sophisticated attacks.

Chrome Enterprise Premium builds on the core capabilities available in Chrome Enterprise, which gives organizations the secure, reliable browser their employees prefer to use with management tools for IT and security teams built in.

K.Mizra LLC, Ex. 2001, p. 138
Google LLC v. K.Mizra LLC
IPR2025-01437



Chrome Enterprise Premium offers additional advanced security capabilities, including:

- **Enterprise controls** enforce policies, manage software updates and extensions to align with enterprise policies, and support RDP, SCP, SSH and other TCP protocols;
- **Security insights and reporting** support event reporting, device reporting, and forensic capabilities for enterprise-wide visibility, and can integrate with other Google and third-party security solutions;
- **Context-aware access controls** can be scaled for web applications, can help enforce continuous Zero Trust access to SaaS and web-based apps with context-aware access control, and can mitigate data exfiltration risks for sanctioned and unsanctioned applications; and
- **Threat and data protection** delivers content inspection and data loss prevention, anti-malware, and anti-phishing using frontline intelligence and AI, dynamic URL filtering, and site categorization.

Organizations using Chrome Enterprise Premium benefit from Google's threat intelligence, security features, and Zero Trust access via Google's secure global network. *See* https://cloud.google.com/blog/products/identity-security/introducing-chrome-enterprise-premium, attached as Exhibit 3 and incorporated herein by this reference.

K.Mizra LLC, Ex. 2001, p. 139
Google LLC v. K.Mizra LLC
IPR2025-01437

46. CEP is based on Google's "BeyondCorp" product. *See, e.g.*, https://cloud.google.com/beyondcorp-enterprise/docs/access-protection#:~:text=Based%20on%20the%20BeyondCorp%20security,SSH%20access%20to%20virtual%20machines, attached as Exhibit 4 and incorporated herein by this reference; *see also* https://cloud.google.com/beyondcorp?hl=en, attached as Exhibit 5 and incorporated herein by this reference. Google refers to BeyondCorp in some CEP-related documents and website pages. BeyondCorp currently is not an accused product, but K.Mizra reserves the right to allege infringement by BeyondCorp if discovery dictates and provides support for such an allegation.

B. The Asserted Hyperparameter Tuning Patent

47. K.Mizra is the sole owner by assignment of the Asserted Hyperparameter Tuning Patent with the full and exclusive right to bring suit to enforce the Patent. K.Mizra is also entitled to sue to collect damages for all past infringement of the Asserted Hyperparameter Tuning Patent.

48. The '120 Patent, titled "Machine Learning Hyperparameter Estimation," was issued by the USPTO to Inventor Stephen Alexander Raaijmakers on May 7, 2013. A true and correct copy of the '120 Patent is attached hereto as Exhibit 6 and incorporated by reference.

49. The Asserted Hyperparameter Tuning Patent is a U.S. National Phase filing of PCT/NL2008/050247 and claims priority to European Application Nos. 07106963.7 filed April 25, 2007 and 07112037.2 filed July 9, 2007.

1. Prior Licensing and Litigation Concerning the Hyperparameter Tuning Patent

50. The Asserted Hyperparameter Tuning Patent has previously been licensed to various industry participants but has not been involved in any prior litigation.

51. Google is not and has never been a licensee of the Asserted Hyperparameter Tuning Patent nor has it had nor does it have any rights to use technologies covered by the Patent. Google

K.Mizra LLC, Ex. 2001, p. 140
Google LLC v. K.Mizra LLC
IPR2025-01437

thus has no ownership or other rights (and is entitled to no rights) relating to the Asserted Hyperparameter Tuning Patent.

52. No party licensed to the Asserted Hyperparameter Tuning Patent has used the technology claimed by the Asserted Hyperparameter Tuning Patent in a manner that requires marking under 35 U.S.C. § 287. K.Mizra is therefore not barred from seeking past damages for infringement of the Asserted Hyperparameter Tuning Patent under 35 U.S.C. § 287(a).

2. Problems in 2007 Solved By the Asserted Hyperparameter Tuning Patent

53. The technology described in the Asserted Hyperparameter Tuning Patent was invented by Stephen Raaijmakers and generally relates to improving the operation of Artificial Intelligence ("AI") through the improved automatic, systematic and computer implemented training techniques. More specifically, AI typically uses models to perform specific tasks, and, at its most basic level, an AI model is a program that has been "trained" on a set of data to recognize certain patterns or to make certain decisions without further human intervention. For example, an AI model could be used to automatically detect spam or other unwanted emails in a user's email inbox or distinguish between ripe and unripe fruit to be picked, etc. The '120 Patent uses the term "classifier" to refer to these types of AI models, as they are used to separate inputs (emails or fruit) into different classifications (spam or not spam/ripe or unripe).

54. AI models use variables called "parameters" to determine what result it should produce (e.g., to determine whether to classify an email as spam or not spam). These parameters are not set manually. Instead, they are learned automatically during the training process. During training, the model may be exposed to a labeled dataset and the parameters can be adjusted to minimize prediction errors. For instance, in the spam email example, the AI model may be trained by exposing the model to a large number of emails that have been labeled as either spam or not

K.Mizra LLC, Ex. 2001, p. 141
Google LLC v. K.Mizra LLC
IPR2025-01437

spam. The model identifies and thus learns the traits and characteristics found in these various emails that are indicative of a spam email or not spam email. Automatically adjusting the parameters of the model to minimize the times when a spam email is marked as not spam, or vice versa, is the goal of parameter-based training.

55. AI models are generally trained for a specific purpose. However, training a full model is time consuming and, in some cases, has given way to the use of foundational models, which are models that are pre-trained on large, genericized datasets. These foundational models can later be fine-tuned for use in addressing a particular problem or issue by, for example, feeding it a set of additional relevant data and training upon that dataset. By way of example, a foundational AI model for diagnosing medical conditions could be fine-tuned by training it using various medical records. Similarly, a foundational large language model ("LLM") like ChatGPT could be fine-tuned to adjust its conversational tone by training it on sample texts representing a desired communication style. Using a foundational AI model is like getting a shirt and later tailoring it to fit your body, rather than starting with fabric, thread, and a needle.

56. As AI models become more advanced and versatile, they also demand more data and computing power to be properly trained. One of the main challenges in this exploding field of technology is efficiently training these models, which typically is done by configuring variables in a model's learning process. These variables are known as hyperparameters. The hyperparameters may, for example, define the architecture (and thus the size) of the model, as well as, how the model changes based on the training data it digests and that will effect how efficiently and effectively the model is trained. For example, a hyperparameter called the learning rate may be used to control how much a model adjusts its parameters during training. If the learning rate of the model is too low, it will take a long time for the model to reach an acceptable result. On the

K.Mizra LLC, Ex. 2001, p. 142
Google LLC v. K.Mizra LLC
IPR2025-01437

other hand, if the learning rate is too high, the model may overreact to each training round, bouncing around an optimum result instead of converging on that result. Different hyperparameters may be used depending on whether a user is training a brand-new model or fine-tuning a foundational model. In either case, though, identifying the right hyperparameters and their values during the model training process improves the model's efficiency and thus the predictions or decisions to be output by the model.

57. Claims of the Asserted Hyperparameter Tuning Patent are directed to technological solutions that address specific challenges grounded in the systematic and automatic tuning of hyperparameters so that efficient and effective AI models can be computer generated. Creating these models is a tremendous concern for modern enterprises, since they are now being used in all manner of human endeavor, from drug discovery, to providing banking efficiencies, etc.

58. The inventor of the Asserted Hyperparameter Tuning Patent understood that while basic models that trained on parameters existed prior to 2007, as the data to be analyzed become more and more voluminous and complex, the use of just parameters in developing a model would not be enough. Rather, the additional use of hyperparameters, that could automatically and systematically be optimized by the computing system that was training the model was needed. The inventor also realized that the existing methods for estimating and selecting hyperparameters were unsuitable in many cases. For example, the '120 Patent discloses:

> Hyperparameters may be determined using various methods, for example heuristic or statistical optimization methods. However, many heuristic methods are ad hoc methods, while not all statistical optimization methods are suitable for determining hyperparameters.

The inventor thus recognized that a new paradigm needed to be developed. That paradigm was then well before its time invented and is disclosed in the '120 Patent.

K.Mizra LLC, Ex. 2001, p. 143
Google LLC v. K.Mizra LLC
IPR2025-01437

59. For example, the '120 Patent explains that a method of determining hyperparameters in a machine learning system is disclosed. The method is iterative, and with each computer implemented iteration, a random or pseudorandom sample of hyperparameter vectors selected from a set of possible hyperparameter vectors must be considered, updating the estimate of the target hyperparameter vector by selecting for use the best vector in the present and previous iterations. The '120 Patent specification further explains that:

> By selecting from the random sample a or the hyperparameter vector producing the best result in the present and any previous iterations, and using said selected hyperparameter vector to update the estimate of the target hyperparameter vector, the properties of the best performing hyperparameters are used to guide the next iteration [of the hyperparameter selection process]. In this way, a method similar to the cross-entropy method [used for selection of parameters] can be used effectively for determining hyperparameters, even when the hyperparameters are not continuous and their effects are not transparent. By using the method according to the present invention, an improved classification [i.e. model] accuracy is achieved.
>
> …
>
> In a preferred embodiment, the method according to the Present invention further comprises the steps of:
>
> selecting from the random sample a further hyperparameter vector producing the best result in any previous iterations, and
>
> restricting the random sample of hyperparameter vectors by using the further selected hyperparameter vector.
>
> The step of restricting the random sample preferably involves the use of an interval surrounding the further selected hyperparameter vector which produces the best result in any previous iteration.
>
> …
>
> In a preferred embodiment, E^t is the hyperparameter vector X^t, at iteration t producing the best result $S(X^t_i)$ in the present iteration t and all previous iterations (if any). The random sample X^t_i, has elements X^t_{ij}, while the step of updating the hyperparameters comprises the step of determining the (target) hyperparameter v^t_i, where

K.Mizra LLC, Ex. 2001, p. 144
Google LLC v. K.Mizra LLC
IPR2025-01437

$$v_j^t = \frac{\sum_{i=1}^{n} I\{S(X_i^t) \geq \gamma^t\} W(X_i^t; E^t) X_{ij}^t}{\sum_{i=1}^{n} I\{S(X_i^t) \geq \gamma^t\} W(X_i^t; E^t)},$$

wherein γ^t is a threshold value, W is a weighting function and $I\{S(X_i^t) \geqq \gamma^t\}$ is an indicator function which is equal to 1 if $S(X_i^t)$ greater than or equal to the threshold value γ^t, and else is equal to 0. The weighting function W is preferably given by

$$W(X_i^t; E^t) = 1 - \frac{\sqrt{\sum_{j=1}^{m} (X_{ij}^t - E_j^t)^2}}{\sqrt{\sum_{j=1}^{m} (X_{ij}^t)^2} \sqrt{\sum_{j=1}^{m} (E_j^t)^2}}.$$

Accordingly, the step of updating the hyperparameters may involve a weighting function based upon a distance function, preferably a Euclidean distance function. It is noted that when the Euclidean distance between E^t and X^t_{ij}, is zero, the weighting function W is equal to one.

The method according to the present invention is typically carried out by computer apparatus, for example a general purpose computer system comprising a processor and an associated memory, one part of the memory storing a soft ware program for instructing the processor to carry out the method steps of the present invention, and another part of the memory storing data, said data comprising the hyperparameter values referred to above.

See, e.g., Ex. 6 at 2:1–4:14.

60. The solution to the problems identified in the '120 Patent—as specified and claimed in the Asserted Hyperparameter Tuning Patent—was an advanced departure from the conventional tuning of hyperparameters in AI models in use in 2007 and was then, as it remains today, patent-eligible, highly valuable, novel, and non-obvious technology.

3. K.Mizra's Asserted Hyperparameter Tuning Patent Claims Are Presumed Valid

61. None of the Asserted Hyperparameter Tuning Patent Claims are directed to abstract ideas, and each employs inventive concepts and is directed to patent-eligible subject matter. All

K.Mizra LLC, Ex. 2001, p. 145
Google LLC v. K.Mizra LLC
IPR2025-01437

claims of the Asserted Hyperparameter Tuning Patent are also presumed to be valid and enforceable against Google and others.

62. The subject matter of the Asserted Hyperparameter Tuning Patent is focused on problems in machine learning systems. *See, e.g.,* Ex. 6 at 1:6-9, 1:13-23, 2:1-19, 3:46-53, 4:9-13. Indeed, hyperparameters are used by machine learning systems to determine the parameters for a classifier in a machine learning system. In other words, the solution described by the claims of the Asserted Hyperparameter Tuning Patent is necessarily rooted in computer technology.

63. Moreover, selecting the appropriate hyperparameters is vital to the speed and efficiency of the training computer and the machine learning system. Selecting inappropriate hyperparameters can lead to myriad problems in training the AI model, including underfitting or overfitting, which in turn could lead to a training system that is inefficient, takes too long to determine the parameters, and/or consumes massive amounts of power; or a resulting machine learning system with a poorly-trained classifier (because the model never converged) which thus misclassifies a significant amount of data.

64. The Asserted Hyperparameter Tuning Patent Claims are directed to a machine-implemented, iterative process for locating, selecting and utilizing in a model an optimized hyperparameter. As such, the Asserted Hyperparameter Tuning Patent Claims recite inventions with specific applications or improvements to technologies in the marketplace and cannot be considered abstract or patent ineligible under relevant law.

4. Google's Infringing Vertex AI Studio Product

65. Google's Vertex AI Studio is part of the Google Cloud Platform and is a complete system for building, training, and using AI models. It offers a range of tools and APIs (Application Programming Interfaces) that let users either create and train their own models from scratch, or

K.Mizra LLC, Ex. 2001, p. 146
Google LLC v. K.Mizra LLC
IPR2025-01437

work with Google's foundation models, like Gemini, Imagen, Codey, and Chirp, and tailor them for specific needs. As is here relevant, Vertex AI provides several options for training all manner of models. One of the most powerful of these provides a user with complete control over the model development process by allowing them access to powerful hyperparameter tuning and optimization options. These can be implemented using Google's Vertex AI Vizier. It is Google's provision of this computer implemented hyperparameter tuning ability that takes advantage of iterative, randomized/pseudo-randomized hyperparameter tuning techniques, that makes Google's Vertex AI so valuable to its customers. Google's usurpation of the critical '120 Patent's patented technology without K.Mizra's authorization allows Google to provide to its customers highly tuned and effective AI models, generating significant revenues and reputational prestige for Google, all to the injury of K.Mizra.

~~41.~~66. Google has been making, selling, using and offering for sale AI products and services that infringe the Asserted Hyperparameter Tuning Patent in violation of 35 U.S.C. § 271 (collectively, "the Accused ~~Zero Trust Network Security~~Hyperparameter Tuning Instrumentalities"). These Accused ~~Zero Trust Network Security~~Hyperparameter Tuning Instrumentalities include, but are not limited to, ~~Google Chrome Enterprise Premium~~Google's Vertex AI system, the sale, offer for sale, use, and/or manufacture in the United States of which constitutes infringement of at least claim 1 of the ~~Asserted Zero Trust Network Security Claims~~Accused Hyperparameter Tuning Patent, either literally or under the doctrine of equivalents.

~~42. Upon information and belief, Google's Chrome Enterprise Premium product was based upon and formerly known as Google's "BeyondCorp Enterprise" product.~~

C. The Asserted Mesh Networking Patent

K.Mizra LLC, Ex. 2001, p. 147
Google LLC v. K.Mizra LLC
IPR2025-01437

67. K.Mizra is the sole owner by assignment of the Asserted Mesh Networking Patent with the full and exclusive right to enforce the Patent. K.Mizra is also entitled to sue to collect damages for all past infringement of the Asserted Mesh Networking Patent.

68. The '717 Patent, titled "Initialization of a Wireless Communication Network," was issued by the USPTO to Inventors Blange and Djurica on March 27, 2012. A true and correct copy of the '717 Patent is attached hereto as Exhibit 7 and incorporated by reference.

69. The Asserted Mesh Networking Patent is a U.S. National Phase filing of PCT/NL2007/0000001, filed on December 2, 2007, and is entitled to that priority date. The Asserted Mesh Networking Patent further claims priority to European Patent Application No. EP 05078044, filed December 30, 2005, and is entitled to that priority date.

1. Prior Licensing and Litigation Concerning the Mesh Networking Patent

70. The Mesh Networking Patent has previously been licensed to various industry participants but has not been involved in any prior litigation.

71. Google is not and has never been a licensee of the Asserted Mesh Networking Patent nor has it had nor does it have any rights to use technologies covered by the Patent. Google thus has no ownership or other rights (and is entitled to no rights) relating to the Asserted Mesh Networking Patent.

72. No party licensed to the Asserted Mesh Networking Patent has used the technology claimed by the Asserted Mesh Networking Patent in a manner that requires marking under 35 U.S.C. § 287. K.Mizra is therefore not barred from seeking past damages for infringement of the Asserted Mesh Networking Patent under 35 U.S.C. § 287(a).

K.Mizra LLC, Ex. 2001, p. 148
Google LLC v. K.Mizra LLC
IPR2025-01437

2. Problems in 2007 Solved By the Mesh Networking Patent

73. The technology described in the Mesh Patent was invented by Marinus Johannes Blange and Miodrag Djurica and generally relates to creation of a flexible mesh wireless network.

74. Although wireless communications networks were known in the art prior to the invention, the known networks had several drawbacks. For example, there existed known techniques for routing messages in wireless networks. *See generally* Ex. 7 at 1:13-58. However, as explained in the '717 Patent, these techniques "require[d] transmission of information via sensor stations before the routing topology [of the network] has been defined." *Id.* at 1:59-62. "This complicates the (sensor) stations, because they will have to support different modes of communication while operating in a low-power mode." *Id.* at 1:62-64. As further explained in the '717 Patent, the existing techniques also resulted in either slow or inefficient networks:

> If forwarding of information is only done using the definite routing topology, a central approach is slow (each station has to wait until its upward station has been centrally assigned a route) and a decentral approach is inefficient (a routing topology with an unnecessary amount of branches has been created).

Id. at 1:64-2:2.

75. Another known protocol was a mobile communications protocol which uses two types of networks: a high power network and a low power network. *See generally* Ex. 7 at 2:9-57. However, the '717 Patent notes that "[t]he use of two types of network without central unit makes this system complex and inefficient in bandwidth use." *Id.* at 3:1-2.

76. The solution to these problems—as specified and claimed in the Asserted Mesh Networking Patent—was an advanced departure from the conventional networking solutions then in use and was then, as it remains today, patent eligible, highly valuable, novel, and non-obvious technology.

K.Mizra LLC, Ex. 2001, p. 149
Google LLC v. K.Mizra LLC
IPR2025-01437

3. K.Mizra's Asserted Mesh Networking Patent Claims Are Presumed Valid

77. None of the Asserted Mesh Networking Patent Claims are directed to abstract ideas, and each employs inventive concepts and is directed to patent-eligible subject matter. All claims of the Asserted Mesh Networking Patent are also presumed to be valid and enforceable against Google and others.

78. The subject matter of the Asserted Mesh Networking Patent is focused on problems in wireless mesh networks, and in particular problems related to creating a new wireless mesh network and/or attaching a new station to an existing wireless mesh network. *See, e.g.,* Ex. 7 at 1:59-2:8, 2:58-3:2, 3:6-15, 3:59-67, 5:11-14. The solution described by the claims of the Asserted Mesh Networking Patent is necessarily rooted in wireless mesh networking technology and presumptively patent eligible.

79. Moreover, the techniques specified and claimed in the Asserted Mesh Networking Patent Claims provides various technical benefits to the network. For example, Claim 11 of the Asserted Mesh Networking Patent is directed to, among other things, a "processor circuit being configured to switch to establish an associated route for communication with an association unit during subsequent communication of operating messages to an association unit, the associated route including a source of the association grant, and, if the source of the association grant is not the association unit, the operating route between the source of the association grant and the association unit previously established for the source of the association grant in response to association request messages and/or the association grants for intermediate stations." Use of the claimed technique improves the speed of the network by providing the associated route to each station that joins a network, allowing the source of the association grant to provide the route information directly rather than retrieving it from the association unit, improving the speed with

K.Mizra LLC, Ex. 2001, p. 150
Google LLC v. K.Mizra LLC
IPR2025-01437

which the network can heal/form itself. It further eliminates unnecessary communications between the source of the association grant and the association unit, conserving bandwidth (and thus allowing more bandwidth for other communications) and conserving power of the various stations located within the associated route. As noted in the Asserted Mesh Networking Patent, power consumption is a key consideration in low power devices often found in mesh networks. Ex. 7 at 5:11-14; *see also id.* at 11:30-42.

80. The Asserted Mesh Networking Claims are directed to wireless networks and stations within wireless networks that implement the techniques described for allowing new stations to efficiently and effectively join the network. As such, the Asserted Mesh Networking Claims recite inventions with specific applications or improvements to technologies in the marketplace and cannot be considered abstract or patent ineligible under relevant law.

4. Google's Infringing Nest Devices

81. Google Nest is a line of products for home networking, allowing a user to control various Google and third-party smart devices throughout the Google Home app. On information and belief, Google Nest was previously known as Google Home.

82. Google offers a wide variety of home networking devices under the Google Nest brand. Many of these devices use a networking protocol called Thread.

83. Thread is a networking protocol designed for low-power Internet of Things devices operating in an IEEE 802.11.15.4 wireless mesh network. It may be used alongside other networks, such as a Wi-Fi network, in a user's home. Thread was developed by the Thread group, which Google founded with other member companies in 2014.

84. Thread is built into many new smart devices such as smart bulbs, locks, sensors, and the like. Thread is used to create a mesh network in which each new smart device connects to

K.Mizra LLC, Ex. 2001, p. 151
Google LLC v. K.Mizra LLC
IPR2025-01437

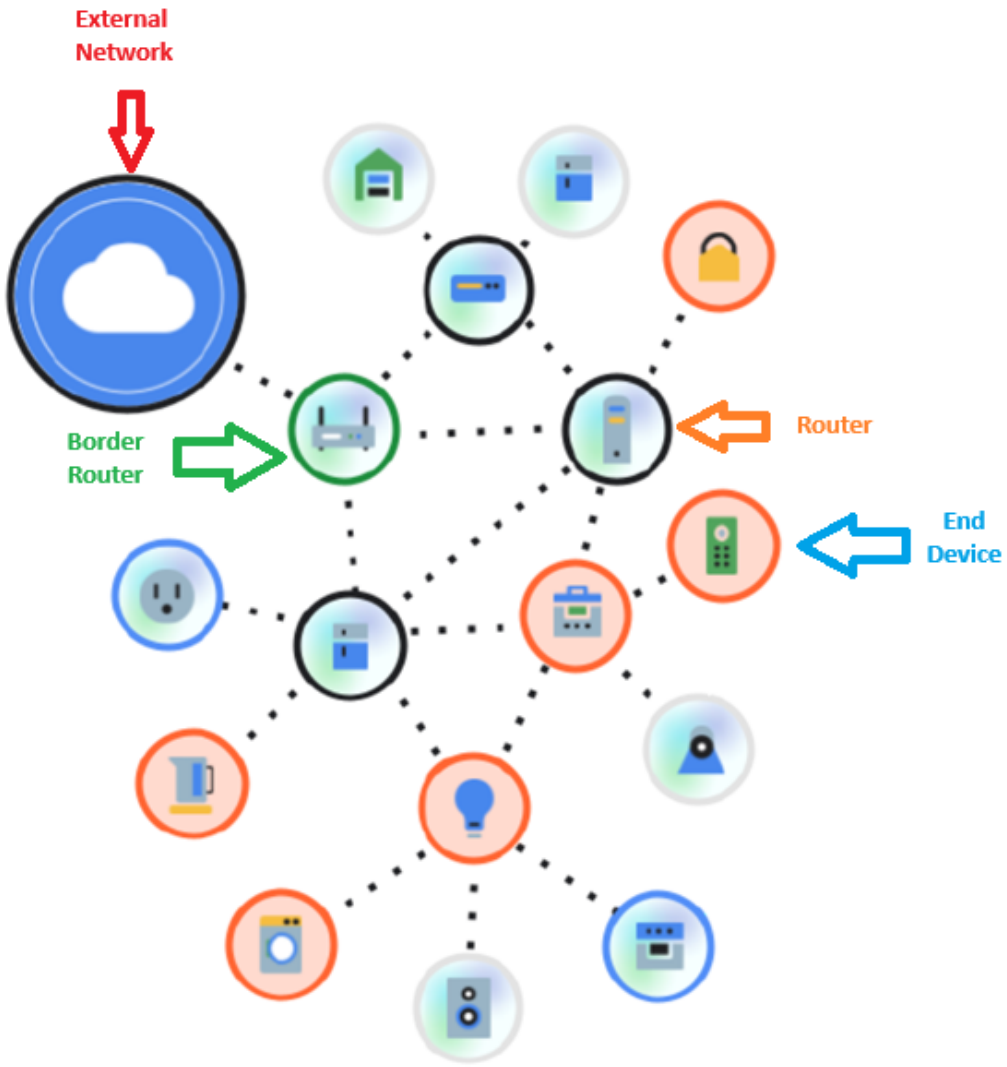
the next nearest smart device, giving users more networking coverage and making the entire network more robust. If one device in the network fails, another can step in to maintain the signal, allowing the network to heal itself rapidly, effectively and efficiently.

85. A device in a Thread network may be either a router, which may forward data packets for other network devices and provides commissioning for devices trying to join the Thread network, or an end device, which communicates with a single router and does not forward packets for other network devices. An end device may convert itself into a router when a new device joins the network.

86. A special case of a router is a border router, which can forward information between a Thread network and a non-Thread network. For example, a border router may allow a user to control the devices on a Thread network through the Wi-Fi network via a border router.

87. An exemplary Thread network including a border router, routers, and end devices is shown below:

K.Mizra LLC, Ex. 2001, p. 152
Google LLC v. K.Mizra LLC
IPR2025-01437



Ex. 8, "Thread Play Services APIs," (available at https://developers.home.google.com/matter/thread (accessed May 12, 2025) and incorporated by reference) (annotated).

88. Google offers a wide range of products that use the Thread networking protocol. For example, Google offers several hubs for Google home that include a built-in Thread router, including the Nest Hub, Nest Hub Max, Nest Wifi Pro, and Google TV Streamer.

K.Mizra LLC, Ex. 2001, p. 153
Google LLC v. K.Mizra LLC
IPR2025-01437

Google devices like the Nest WiFi, Google Nest Hub Max and Google Nest Hub (2nd gen) have Thread radios built-into them and act as Thread Border Routers.

Id.; *see also* Ex. 9, "Connect Thread smart devices with Google TV Streamer (4K)," (available at https://support.google.com/chromecast/answer/15178609 (accessed May 21, 2025) and incorporated by reference).

89. In addition, Google hub products such as the Google TV Streamer, Nest Hub (2nd Gen), Hub Nest Max Fuchsia, Nest Wi-Fi Point, and Nest Wi-Fi Router are certified as Built on Thread by the Thread Group.

90. As another example, Google offers a routing device called the Nest Connect.

Nest Connect



Ex. 10, Nest Connect – Range Extender for Detect & Guard – Google Store," (available at https://store.google.com/product/nest_connect (accessed May 13, 2025) and incorporated by reference)). The Nest Connect includes a Thread radio that allows communication between a Thread border router and a peripheral device such as the Nest x Yale Lock.

K.Mizra LLC, Ex. 2001, p. 154
Google LLC v. K.Mizra LLC
IPR2025-01437

> In an emailed statement, Grant Erickson, president of the Thread Group said, "Nest Detect relies on Thread's low power architecture to last for long periods of time on one battery. Nest Connect is a Thread router, which seamlessly extends both the reach and connectivity of the Thread network. Nest Guard provides seamless IP-based routing between Thread, Wi-Fi, and cellular, and their interoperability with the Yale Linus Lock is the posterchild of Thread's benefits in action."

Ex. 11, "Nest debuts Nest Secure home system and has a Thread router called Nest Connect," (available at https://staceyoniot.com/nest-debuts-nest-secure-home-system-and-has-a-thread-router-called-nest-connect/ (accessed May 13, 2025) and incorporated by reference).

91. Yet another example, Google offers various peripheral devices that include a Thread radio such as the Nest x Yale Lock.



Ex. 12, "Nest x Yale Lock – Key-Free Smart Deadbolt – Google Store," (available at https://store.google.com/product/nest_x_yale_lock (accessed May 13, 2025) and incorporated by reference).

92. Additional peripheral devices offered by Google that include a Thread radio include the Google Nest Learning Thermostat and the Google Nest Protect and CO alarm.

K.Mizra LLC, Ex. 2001, p. 155
Google LLC v. K.Mizra LLC
IPR2025-01437

93. Google has been making, selling, using and offering for sale products including Thread radios that infringe the Asserted Mesh Networking Patent in violation of 35 U.S.C. § 271 (collectively, "the Accused Mesh Networking Instrumentalities"). These Accused Mesh Networking Instrumentalities include, but are not limited to, Google's Nest home networking products identified above, the sale, offer for sale, use, and/or manufacture in the United States of which constitutes infringement of at least claim 11 of the Accused Mesh Networking Patent, either literally or under the doctrine of equivalents.

94. K.Mizra expects that discovery will reveal additional products that include Thread radios and infringe the Asserted Mesh Networking Patents, which may include additional Google Nest products and/or products that are not marketed under the Google Nest brand. For example, there are reports that at least some Google Pixel 9 devices may include a Thread radio. *See, e.g.*, Ex. 13, "Pixel 9 series arrives at the FCC, includes Thread," (available at https://9to5google.com/2024/07/12/pixel-9-fcc/ (accessed May 23, 2025) and incorporated by reference).

~~V.~~ FIRST CLAIM FOR RELIEF
~~VI.~~V.
(Patent Infringement Under 35 U.S.C. § 271 of the '705 Patent)

~~43.~~95. K.Mizra incorporates paragraphs 1 through ~~42~~94 as though fully set forth herein.

~~44. The '705 Patent includes 19 claims.~~

~~45. Google has infringed and is infringing one or more claims of the '705 Patent by making, importing, using, offering for sale, and/or selling the Accused Zero Trust Network Security Instrumentalities, in violation of 35 U.S.C. § 271(a).~~

K.Mizra LLC, Ex. 2001, p. 156
Google LLC v. K.Mizra LLC
IPR2025-01437

~~46. Upon information and belief, and based upon publicly-available information, the Accused Zero Trust Network Security Instrumentalities meet each element of at least Claim 19 of the '705 Patent.~~

~~47.~~96. Claim 19 of the '705 Patent states:

> [preamble] A computer program product for protecting a network, the computer program product being embodied in a non-transitory computer readable medium and comprising computer instructions for:
>
> [A] detecting an insecure condition on a first host that has connected or is attempting to connect to a protected network,
>
> [B] wherein detecting the insecure condition includes:
>
> > [B1] contacting a trusted computing base associated with a trusted platform module within the first host,
> >
> > [B2] receiving a response, and determining whether the response includes a valid digitally signed attestation of cleanliness,
>
> [C] wherein the valid digitally signed attestation of cleanliness includes at least one of an attestation that the trusted computing base has ascertained that the first host is not infested, and an attestation that the trusted computing base has ascertained the presence of a patch or a patch level associated with a software component on the first host;
>
> [D] when it is determined that the response does not include a valid digitally signed attestation of cleanliness, quarantining the first host, including by preventing the first host from sending data to one or more other hosts associated with the protected network,
>
> [E] wherein preventing the first host from sending data to one or more other hosts associated with the protected network includes
>
> > [E1] receiving a service request sent by the first host, serving a quarantine notification page to the first host when the service request comprises a web server request,
> >
> > [E2] and in the event the service request comprises a DNS query, providing in response an IP address of a quarantine

K.Mizra LLC, Ex. 2001, p. 157
Google LLC v. K.Mizra LLC
IPR2025-01437

> server configured to serve the quarantine notification page if a host name that is the subject of the DNS query is not associated with a remediation host configured to provide data usable to remedy the insecure condition; and
>
> [F] permitting the first host to communicate with the remediation host.

Ex. 1 at 22:14–49.

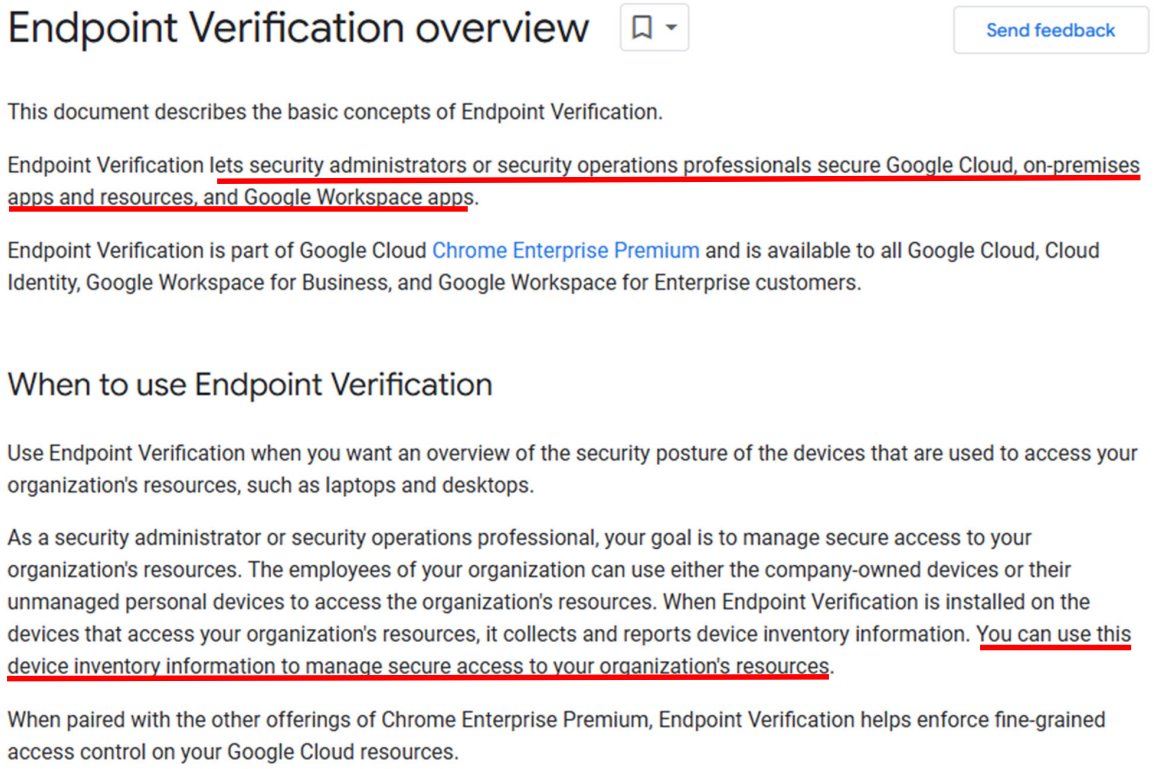
~~48.~~97. ~~Regarding the~~The preamble of Claim 19~~, and to the extent the preamble~~ is ~~determined~~nonlimiting. But if it is interpreted to be ~~limiting (which it is not), the Accused Instrumentalities provide the features described in the preamble, which recites~~otherwise, CEP is a "computer program product for protecting a network." For example, Google touts that ~~its Chrome Enterprise product~~CEP is "Google Cloud's zero-trust solution that enables an organization's workforce to access web applications securely from anywhere, without the need for VPN and without fear of malware, phishing, and data loss":

> Chrome Enterprise Premium is Google Cloud's zero-trust solution that enables an organization's workforce to access web applications securely from anywhere, without the need for VPN and without fear of malware, phishing, and data loss.
>
> By using Chrome Enterprise Premium, you can manage access for apps on Google Cloud, other clouds, and on-premises, define and enforce access policies based on user, device, and other contextual factors, and make apps more accessible and responsive through Google's global network.

See Ex. ~~3, Chrome Enterprise Premium documentation~~14, p. 1 (available at https://cloud.google.com/beyondcorp-enterprise/docs) (last accessed mid Jan. 2025 and incorporated by reference). Additionally, ~~Google's Chrome Enterprise products deliver~~CEP provides endpoint (*e.g.*, "host" computer~~),~~) posture assessments and ensure that endpoints meet security and compliance policies before they are allowed to access a protected network. *See id.* Accordingly, and to the extent the preamble of Claim 19 is deemed limiting, ~~the Accused Zero Trust Network Security Instrumentalities meet~~CEP meets the limitation.

K.Mizra LLC, Ex. 2001, p. 158
Google LLC v. K.Mizra LLC
IPR2025-01437

~~49.~~98. ~~Limitation~~CEP also meets the requirements of limitation A of Claim 19, which requires "detecting an insecure condition on a first host that has connected or is attempting to connect to a protected network." ~~The Accused Zero Trust Network Security Instrumentalities also meet all the requirements of limitation A of Claim 19.~~ For example, ~~Google's Chrome Enterprise products deliver~~CEP delivers endpoint posture assessments and ~~ensure~~ensures that endpoints meet security and compliance policies before they are allowed to connect to ~~the~~secure resources, i.e. network, housed on the Google Cloud:



Endpoint Verification overview

Send feedback

This document describes the basic concepts of Endpoint Verification.

Endpoint Verification lets security administrators or security operations professionals secure Google Cloud, on-premises apps and resources, and Google Workspace apps.

Endpoint Verification is part of Google Cloud Chrome Enterprise Premium and is available to all Google Cloud, Cloud Identity, Google Workspace for Business, and Google Workspace for Enterprise customers.

When to use Endpoint Verification

Use Endpoint Verification when you want an overview of the security posture of the devices that are used to access your organization's resources, such as laptops and desktops.

As a security administrator or security operations professional, your goal is to manage secure access to your organization's resources. The employees of your organization can use either the company-owned devices or their unmanaged personal devices to access the organization's resources. When Endpoint Verification is installed on the devices that access your organization's resources, it collects and reports device inventory information. You can use this device inventory information to manage secure access to your organization's resources.

When paired with the other offerings of Chrome Enterprise Premium, Endpoint Verification helps enforce fine-grained access control on your Google Cloud resources.

See Ex. 415, Endpoint Verification Overview, p. 1 (available at https://cloud.google.com/endpoint-verification/docs/overview) (last accessed mid Jan. 2025 and incorporated by reference).

K.Mizra LLC, Ex. 2001, p. 159
Google LLC v. K.Mizra LLC
IPR2025-01437

> Chrome Enterprise Premium presents a security model that allows for greater security posturing and policy for both applications and devices, while providing end users a better user experience no matter where they access from or what type of device they use to do so:
>
> - For administrators:
>   - Strengthen security posture to account for dynamic changes in a user's context.
>   - Shrink the access perimeter to only those resources that an end user should be accessing.
>   - Enforce device security postures for employees, contractors, partners, and customers for access, no matter who manages the devices.
>   - Extend security standards with per-user session management and multifactor authentication.
> - For end users:
>   - Allow all end users to be productive everywhere without compromising security.
>   - Allow the right level of access to work applications based on their context.
>   - Unlock access to personally-owned devices based on granular access policies.

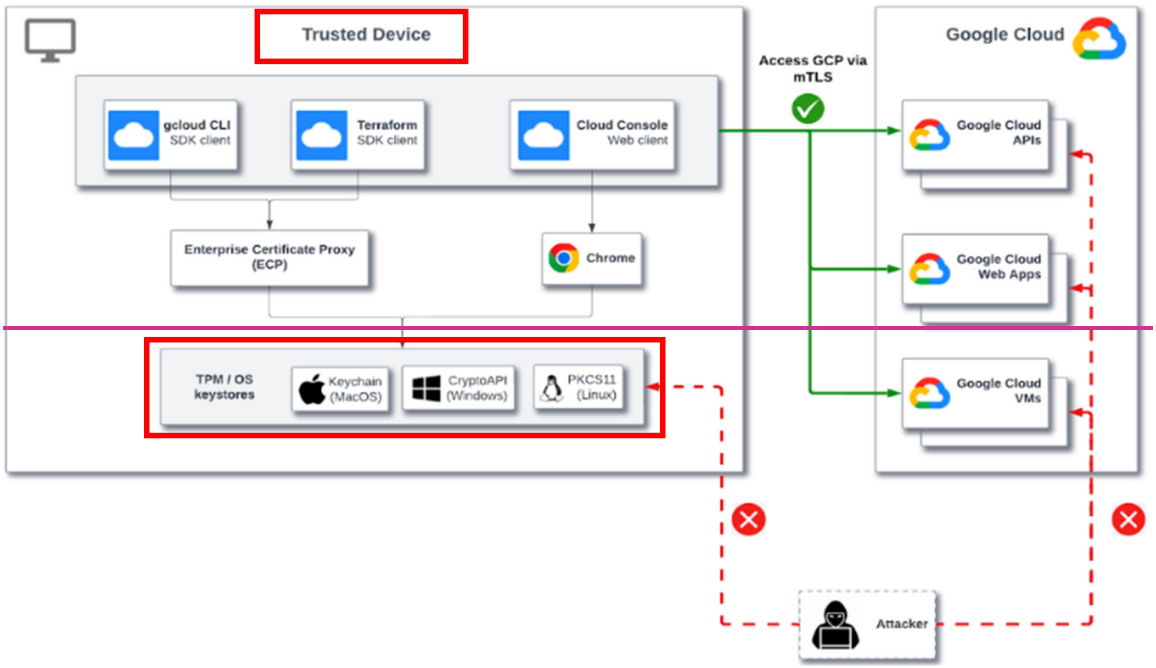
See Ex. ~~5~~16, Chrome Enterprise Premium Overview, pp. 1–2 (available at https://cloud.google.com/beyondcorp-enterprise/docs/overview) (last accessed mid Jan. 2025 and incorporated by reference). ~~Accordingly, the Accused Instrumentalities meet limitation A of Claim 19~~Endpoint Verification is a Google webapp that is integrated into and used by various Google products, including CEP.

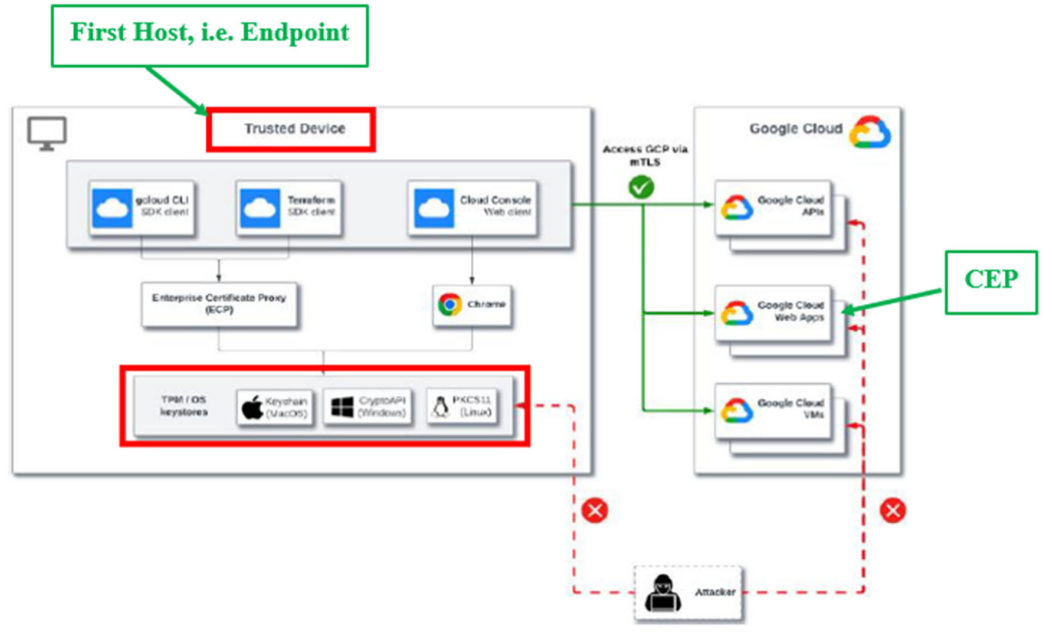
~~50.~~99. ~~Limitation~~CEP further meets limitation B1 of Claim 19, which requires that "detecting [an] insecure condition includes . . . contacting a trusted computing base associated with a trusted platform module within the first host." ~~The Accused Zero Trust Network Security Instrumentalities meet these requirements through, for example, the Google's Chrome Enterprise product which~~ For example, CEP uses "strong key protection" such as "secure cryptographic storage such as TPMs and OS keystores." *See* Ex. ~~6.~~ 17, How To Prevent Account Takeovers With New Certificate-Based Access, p. 1 (available at https://cloud.google.com/blog/products/identity-security/how-to-prevent-account-takeovers-with-new-certificate-based-access) (last accessed mid Jan. ~~2025 and incorporated by~~

K.Mizra LLC, Ex. 2001, p. 160
Google LLC v. K.Mizra LLC
IPR2025-01437

~~reference).~~2025, annotated to show where in the architecture CEP and the First host reside, and incorporated by reference). As the below annotated Google produced figure establishes, many products, including CEP, are integrated into Google Cloud. In operation, what Google calls a Trusted Device (top red box below) and the '705 Patent calls a "first host" or "endpoint" (noted in top green box below), includes a TCB/TPM (see bottom red box below) and uses a Chrome browser, working with other Google Software installed on the Trusted Device, for contact between applications, web apps, virtual machines, etc., operating within the Google Cloud Platform, including CEP.

K.Mizra LLC, Ex. 2001, p. 161
Google LLC v. K.Mizra LLC
IPR2025-01437





See id. at 2~~. The Accused Zero Trust Network Security Instrumentalities obtain~~ (annotated).

Some of the information CEP requests from the Trusted Device, i.e. the first host~~ computer~~

~~through~~, are digitally-signed certificates housed in the TCB/TPM to determine whether the first host ~~computer~~ is secure and can be trusted to access ~~the~~ protected ~~network:~~resources, i.e. networks, also housed within the Google Cloud:

> Certificate-based access can help you build a multi-layered defense against account takeovers, bolster data security, and maintain user trust. At Google, we have used this unique Zero Trust approach for many years as a strong defense to protect our technical infrastructure and employees. Now with CBA, we are extending this same level of security to our Google Cloud customers. Here are the key attributes of this approach:
>
> - **Certificate-based access control**: Granular access policies with X.509 device certificates ensures only legitimate users with the correct certificate can access cloud resources.
> - **Protections beyond initial login**: In contrast to using mTLS only for authentication, CBA also evaluates every authorization request to help safeguard resource access.
> - **Strong key protection:** CBA's use of mTLS leverages secure cryptographic storage such as TPMs and OS keystores. Tooling such as Enterprise Certificate Proxy (ECP) are offered to users that empower them to safeguard private keys helping to ensure keys remain inaccessible to attackers without physical device access.
>
> CBA allows you to enforce certificate-based authentication for all of Google Cloud, including specific resources (VM, Console, API), individual groups or organizations, across multiple end user vectors from browser to gcloud and terraform command-line-interfaces. It is integrated with many services in Google Cloud for effective access enforcement to cloud resources. For example, Context-Aware Access for Google APIs and the Cloud Console, Identity Aware Proxy (IAP) for web apps and workloads, and VPC Service Controls (VPC-SC) for network enforcement.

See id. ~~at 4. Therefore, the Accused Zero Trust Network Security Instrumentalities meet limitation B1 of Claim 19.~~at 4 (annotated). As the above Google produced system architecture description establishes, CEP works with CBA to detect a possible insecure condition on a Trusted Device by contacting a trusted computing base associated with a trusted platform module that is located within the Trusted Device and meets the requirements of limitation B1 of Claim 19 of the '705 Patent.

100. To be clear, the "first host," i.e. the Trusted Device in Google's vernacular, is not an active element of the Asserted Clams. Rather, it is a component of the "environment" in which

K.Mizra LLC, Ex. 2001, p. 163
Google LLC v. K.Mizra LLC
IPR2025-01437

the active claim elements must function. Recently, the Federal Circuit explained "environmental" claiming and the role it plays in determining direct infringement. *INVT SPE LLC v. Int'l Trade Comm'n*, 46 F.4th 1361 (Fed. Cir. 2022). As explained in *INVT:*

> [t]he base station is part of "the environment" in which the [actively claimed] user device must function. *Advanced Software Design Corp. v. Fiserv, Inc.*, 641 F.3d 1368, 1374 (Fed. Cir. 2011). The claims have specific requirements for the data signal that the user device's receiving section and data obtaining section handle and process when the device is activated and put into operation. . . . To understand whether a user device can ever receive a data signal with the particularized characteristics set forth in the claim, it is necessary to know whether the base station (i.e., the communicating party) is capable of transmitting that particular type of data signal to the user device. ***Therefore, although the recited base station is not "a limitation on the claimed invention itself,"*** *Nazomi Communs., Inc. v. Nokia Corp.*, 739 F.3d 1339, 1345 (Fed. Cir. 2014)***, in the sense that an infringer would not need to, for instance, use, make, or sell the base station***, the base station's operation affects whether the claims are met, *see, e.g.*, *Advanced Software Design*, 641 F.3d at 1373-74.
>
> In *Advanced Software*, the claimed invention was for validating a check, to prevent check fraud, and involved either decrypting or encrypting information on the check. The preamble of the claim set out that the check included "selection information [that] is encrypted" to generate a control code and a "control code [which] is printed on the [check]." We held that these steps in the preamble "define[d] the financial instrument that the claimed system validates" as opposed to setting forth steps that would have to be performed by the accused infringer. *Advanced Software*, 641 F.3d at 1373-74 & 1374 n.1. Nevertheless, the accused infringer would infringe "only by validating checks that [had] been encrypted and printed in accordance with steps described in the preamble." *Id*. at 1374.
>
> Like in *Advanced Software*, the claimed device of the [INVT] patent operates in an environment that involves actions of another device (the communicating party, i.e., the base station). The claimed device's capability of performing the recited functions depends on being supplied a certain modulated and encoded signal, which, in turn, requires the supplier (the communicating party) to actually supply that signal. Because the communicating party (base station) generates the necessary environment, its operations must be known to determine whether the accused device infringes, i.e., is capable of performing the claimed functions.

Id. at 1375 (emphasis added); *see also Uniloc USA, Inc. v. Microsoft Corp.*, 632 F.3d 1292, 1308-09 (Fed. Cir. 2011) (holding that claimed "station forming part of a registration system . . . including local licensee unique ID generating means," only "define[d] the environment in which

K.Mizra LLC, Ex. 2001, p. 164
Google LLC v. K.Mizra LLC
IPR2025-01437

that [remote] registration station must function.").

101. Claim 19 of the asserted '705 Patent recites a "computer program product for protecting a network . . ." and that product performs the following active functions:

Receiving a service request from a first host;

Contacting a trusted computing base (TCB) associated with a trusted platform module (TPM) located within the first host;

Receiving a response from the TCB;

Determining from that response whether the first host is infested or clean;

Quarantining, if infestation is detected, the first host; and

Permitting the first host to communicate with a remediation host.[1]

102. CEP is a computer program product that protects resources found on Google Cloud and performs each active recited function of the claims, i.e., the bolded terms, as explained herein,

[1] Claim 19, with the active verbs highlighted, is here provided for ease of reference:

19. A computer program product for protecting a network, the computer program product being embodied in a non-transitory computer readable medium and comprising computer instructions for:

detecting an insecure condition on a first host that has connected or is attempting to connect to a protected network, wherein detecting the insecure condition includes contacting a trusted computing base associated with a trusted platform module within the first host, receiving a response, and determining whether the response includes a valid digitally signed attestation of cleanliness, wherein the valid digitally signed attestation of cleanliness includes at least one of an attestation that the trusted computing base has ascertained that the first host is not infested, and an attestation that the trusted computing base has ascertained the presence of a patch or a patch level associated with a software component on the first host;

when it is determined that the response does not include a valid digitally signed attestation of cleanliness, quarantining the first host, including by preventing the first host from sending data to one or more other hosts associated with the protected network, wherein preventing the first host from sending data to one or more other hosts associated with the protected network includes receiving a service request sent by the first host, serving a quarantine notification page to the first host when the service request comprises a web server request, and in the event the service request comprises a DNS query, providing in response an IP address of a quarantine server configured to serve the quarantine notification page if a host name that is the subject of the DNS query is not associated with a remediation host configured to provide data usable to remedy the insecure condition; and

permitting the first host to communicate with the remediation host.

K.Mizra LLC, Ex. 2001, p. 165
Google LLC v. K.Mizra LLC
IPR2025-01437

specifically including in the exhibits hereto, which are incorporated herein by this reference. The rest of the claim language, specifically including the "first host," is environment within which the active verbs of the claim must operate. K.Mizra thus need not establish that Google supplies the "first host" for it to prove that Google directly infringes the asserted claims. That said, though, Google does supply all relevant portions of the first host and created a secure network architecture in which that first host is to operate.

~~51.~~103. Limitation B2 requires that "detecting the insecure condition" also includes "receiving a response and determining whether the response includes a valid digitally signed attestation of cleanliness." ~~The Accused Zero Trust Network Security Instrumentalities also meet all the requirements of~~CEP meets this limitation ~~B2,~~ as ~~the Chrome Enterprise products receive~~it receives from the first host computer, i.e. the Trusted Device, requested information and then determines, based on the information received, whether ~~a user (*i.e.*, a first "host")~~ the Trusted Device that is attempting to access ~~cloud~~secure resources housed in the Google Cloud (*i.e.*, a protected network) is secure and trusted. *See, e.g.*, *id.* This process requires back-and-forth communication between ~~a host computer~~CEP and the ~~Chrome Enterprise products about~~Trusted Device concerning the cleanliness of the first host computer/endpoint device:

K.Mizra LLC, Ex. 2001, p. 166
Google LLC v. K.Mizra LLC
IPR2025-01437

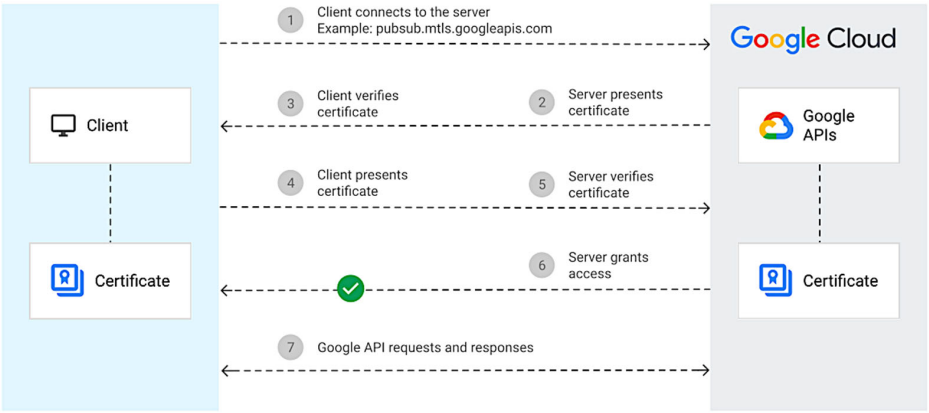
How the Google APIs validate device identity

The TLS protocol uses a technique called public key infrastructure (PKI), which relies on a pair of asymmetric keys: a public key and a private key. Anything encrypted with the private key can be decrypted only with the public key. The Google Cloud APIs use the TLS protocol to verify the identity of a device by decrypting the message encrypted by the private key using the public key of the certificate during the mTLS handshake. The successful decryption proves the possession of the private key which is only available from trusted devices.

To enable the mTLS handshake and validation process, a client must do the following:

- Establish an mTLS connection with the Google APIs by using mTLS-specific API endpoints. The mTLS-specific endpoints have the following format: `[service].mtls.googleapis.com`
- Discover and use the device certificate during the mTLS handshake. If you are using Endpoint Verification for certificate deployment, this type of certificate is automatically discovered and used by the supported clients.

The following diagram illustrates the mTLS handshake between a client and a Google API server:



See Ex. ~~7~~18, Understand Mutual TLS at Google Cloud, p. 1 (available at https://cloud.google.com/beyondcorp-enterprise/docs/understand-mtls) (last accessed mid Jan. 2025 and incorporated by reference). Additionally, ~~Google's Chrome Enterprise Premium products check~~CPE checks digital signatures to determine the cleanliness of the host computer:

Deploy Endpoint Verification to use with certificate-based access

Send feedback

As part of a Chrome Enterprise Premium solution, Endpoint Verification provides critical device trust and security-based access control, and can help enforce fine-grained access control on your Google Cloud resources.

See Ex. ~~8,~~19 (annotated), Deploy Endpoint Verification To Use With Certificate-Based Access, p. 1 (available at https://cloud.google.com/beyondcorp-enterprise/docs/deploy-cba-endpoint-verification) (last accessed mid Jan. 2025 and incorporated by reference). ~~Thus, the Accused Zero Trust Network Security Instrumentalities meet~~Thus, CPE meets limitation B2 of Claim 19.

~~52.~~104. Limitation C requires that "the valid digitally signed attestation of cleanliness includes at least one of an attestation that the trusted computing base has ascertained that the first host is not infested, and an attestation that the trusted computing base has ascertained the presence of a patch or a patch level associated with a software component on the first host." ~~The Accused Zero Trust Network Security Instrumentalities meet these requirements as the Chrome Enterprise products check~~CEP checks the compliance of each endpoint device, i.e. Trusted Device, attempting to connect to ~~the~~ protected ~~network~~resources housed in the Google Cloud by performing an endpoint control check that involves matching the ~~endpoint~~endpoint's configuration parameters received from the endpoint device with specific device profile attributes, such as antimalware programs ~~and~~, applications~~:~~, etc.:

> Google announced a new feature and administration serverside API for the Chrome OS called Verified Access. This API will cryptographically validate the identity of any Chrome OS device that has connected to an enterprise's network. Additionally, it will allow the enterprise to verify that any connected devices conform to the company's security policies.
>
> The new Google API uses digital certificates stored in the hardware-based Trusted Platform Modules (TPMs) that are found in every Chrome OS device. The TPM creates a way to ensure the correct security state of the devices has not been altered.

See Ex. ~~9,~~20 (annotated), New Google API Will Securely Verify Chrome Devices, p. 2 (available at https://securityintelligence.com/news/new-google-api-will-securely-verify-chrome-devices/)

K.Mizra LLC, Ex. 2001, p. 168
Google LLC v. K.Mizra LLC
IPR2025-01437

(last accessed mid Jan. 2025 and incorporated by reference). ~~Accordingly, the Accused Zero Trust Network Security Instrumentalities meet limitation C of Claim 19~~ As already explained and will be further detailed below, Verified Access is a Google API that is used by numerous Google products to help keep data safe and secure and one of these is CEP.

105. Exhibit 16 provides the following explanation of some CEP functions/features as compared to those offered by the less secure Google Cloud product:

Chrome Enterprise Premium compared with Google Cloud

Chrome Enterprise Premium provides enterprise security features in addition to the basic protections, focused on protecting applications with authentication and authorization, that are baseline features of Google Cloud. Chrome Enterprise Premium extends those protections to applications and data running everywhere, with end-user protections and rich access policy protections.

The following table shows the differences between the baseline features available to Google Cloud customers and what is available with Chrome Enterprise Premium:

Applications and Resources Access	GCP Baseline	BeyondCorp Enterprise Essentials	BeyondCorp Enterprise
Access control to web applications on Google Cloud Platform	✓		✓
Access control to SSH, RDP and TCP ports for VMs on GCP	✓		✓
Access control to Google Cloud Platform APIs	✓		✓
Access control to Google Cloud console	✓		✓
Access control to web applications on GCP internal load balancing	✓		✓
Access control to web applications on customer premises			✓
Access control to thick client / client-server applications			✓
Access control to web applications on AWS and Azure			✓
Access control to SAML-based applications (login time)		✓	✓
Access control to Google Workspace Admin Console		✓	✓
Access Policies and Advanced Settings	**GCP Baseline**	**BeyondCorp Enterprise Essentials**	**BeyondCorp Enterprise**

https://cloud.google.com/beyondcorp-enterprise/docs/overview. What this Google document relevantly conveys is that Google Cloud and CEP (formerly called BeyondCorp Enterprise) both provide Access control to Google Cloud Platform APIs (*see* yellow highlighted box above). K.Mizra's reliance on a Google Cloud document to explain how Access Control and Verified Access works in both CEP and Google Cloud, more generally, is not an allegation of infringement against Google Cloud, but rather an explanation of how CEP infringes Claim 19 by calling Google's Verified Access API and thus determining whether the "first host" is "clean" or

"infested."

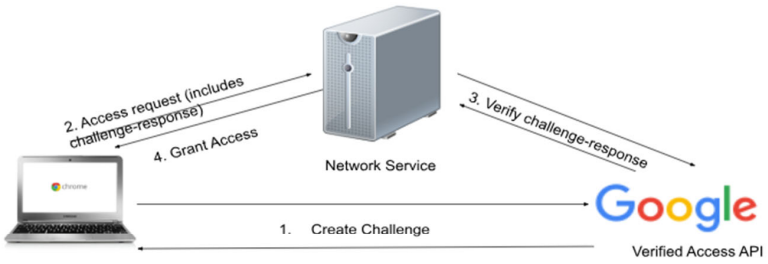
106. More specifically, a Google produced document updated October 16, 2024 and found at https://developers.google.com/chrome/verified-access/overview and attached as Exhibit 21 shows how the Verified Access API is called by Chrome and thus also CEP:

About Chrome Verified Access

Your network service can use the Verified Access feature in the Google Admin console to communicate with both the client Chrome device and the Verified Access API. Doing so gains information about the policy compliance and (optionally) the identity of the client device from Google. To that end, a Chrome extension must be running on the device that interacts with the enterprise.platformKeys extension API, and the network service needs to talk to the Verified Access API.

How Chrome Verified Access Works

Here's the suggested implementation:



Suggested implementation.

1. The Chrome extension contacts the Verified Access API to create a challenge.
2. The Chrome extension calls the enterprise.platformKeys API to generate a challenge-response and sends the access request to the network service, including the challenge-response in the request.
3. The network service contacts the Verified Access API to verify the challenge-response.
4. In case of successful verification, the network service grants access to the device.

CEP, operating in Google Cloud, calls Google's Verified Access API, also operating in the Google Cloud, determining whether the "first host" is "clean" or "infested." If it is the latter, the "first host" is not allowed to access secure resources also stored in the Google Cloud. These functions satisfy the requirements of limitation C of claim 19.

~~53.~~107. Limitation D requires that "when it is determined that the response does not include a valid digitally signed attestation of cleanliness, quarantining the first host, including by preventing the first host from sending data to one or more other hosts associated with the protected

K.Mizra LLC, Ex. 2001, p. 170
Google LLC v. K.Mizra LLC
IPR2025-01437

network." ~~The Accused Zero Trust Network Security Instrumentalities further meet~~CEP meets these requirements by quarantining noncompliant, *i.e.*, unclean, endpoint devices, i.e. Trusted Devices, attempting to connect to ~~the~~ protected ~~network~~resources found in the Google Cloud:

> Using Context-Aware Access, admins can set up different access levels based on a user's identity and the context of the request (location, device security status, IP address). This can help provide granular access controls without the need for a VPN, and give users access to Google Workspace resources based on organizational policies. Insights and recommendations help admins improve the cybersecurity posture of their organization by proactively identifying areas that need attention, significantly reducing the need for admins to identify these risks themselves. For example, if we detect devices with outdated operating system versions accessing corporate Workspace data, we can surface this as an Insight & pair it with a recommendation to block such devices from accessing Workspace data with a few clicks.

See Ex. ~~10~~22, Context Aware Access Insights and Recommendations Are Now Generally Available, p. 3 (available at https://workspaceupdates.googleblog.com/2024/10/context-aware-access-insights-and-recommendations.html) (last accessed mid Jan. 2025 and incorporated by reference).

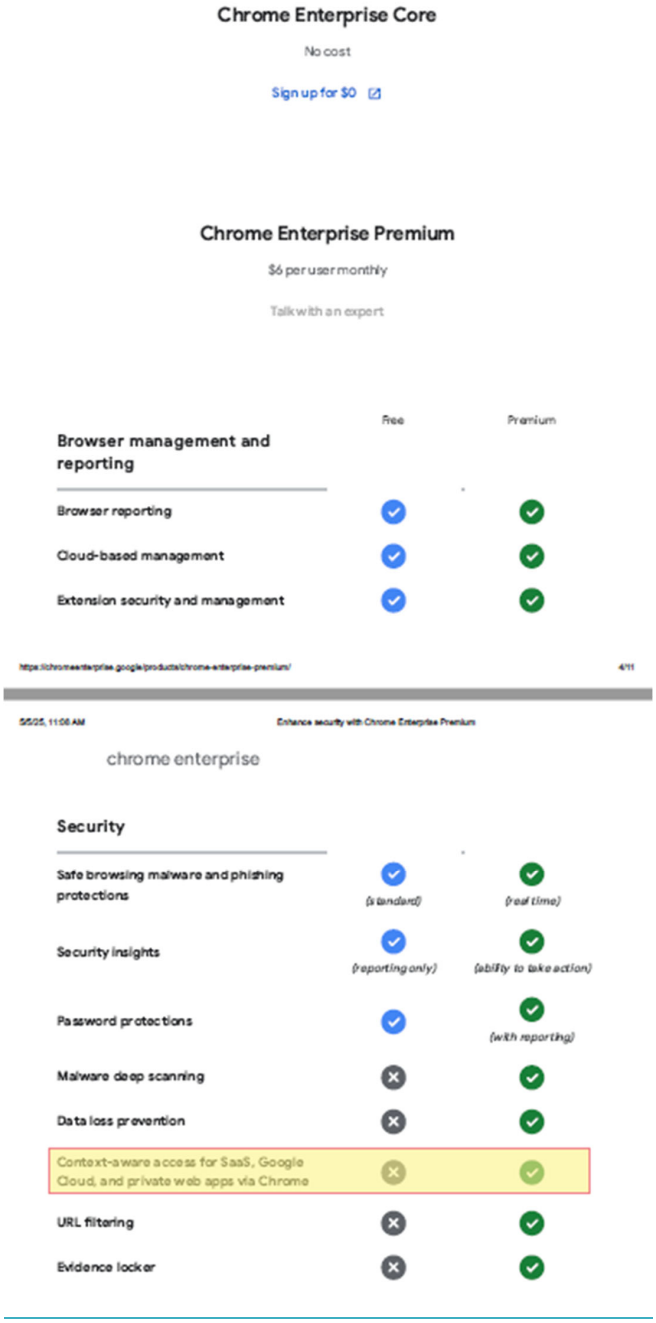
> Using context-aware access, you now have the option to automatically block access to Google Workspace data from compromised Android and iOS devices. A device may be counted as compromised if certain unusual events are detected, including devices that are jailbroken, bypassing of security controls, modification of restricted settings, and more.

See Ex. ~~11~~23, Block Compromised Mobile Devices Using Context-Aware Access, p. 2 (available at https://workspaceupdates.googleblog.com/2024/05/block-compromised-mobile-devices-using-context-aware--access.html) (last accessed mid Jan. 2025 and incorporated by reference).

K.Mizra LLC, Ex. 2001, p. 171
Google LLC v. K.Mizra LLC
IPR2025-01437

~~Accordingly, the Accused Zero Trust Network Security Instrumentalities meet~~ Context-Aware Access is another Google product that is housed in the Googler Cloud that can be used by many other Google products. One of those is CEP, which uses Context-Aware Access to facilitate quarantining of first hosts," i.e. Trusted Devices. To put a finer point in it, the following Google produced content specifically shows that Context-Aware Access is a security feature of CEP:

K.Mizra LLC, Ex. 2001, p. 172
Google LLC v. K.Mizra LLC
IPR2025-01437



CEP meets limitation D of Claim 19.

~~54.~~108. Limitation E1 requires that "preventing the first host from sending data to one or more other hosts associated with the protected network includes . . . receiving a service

request sent by the first host [and] serving a quarantine notification page to the first host when the service request comprises a web server request." ~~The Accused Zero Trust Network Security Instrumentalities meet these requirements because the Chrome Enterprise products are configured to determine~~CEP determines if an endpoint ~~device~~, i.e. Trusted Device, posture information stored in the TMP/TCB matches the profile designated for such a device ~~and if the~~. If that endpoint ~~device~~information does not match, it is quarantined and restricted from accessing the protected network~~. The quarantine prevents~~ housed on the Google Cloud, preventing the ~~user (*i.e.* first host)~~Trusted Device from sending data to or receiving data from the Google Cloud housed protected ~~network by blocking the user~~resources until ~~the user fixes the problem~~problems causing the ~~user~~Trusted Device to be blocked are remediated:

> Using remediation messages and custom messages in Context-Aware Access, you can help users unblock themselves when a policy prevents them from accessing an app. These optional (but recommended) messages can help get users back to productivity and reduce support calls for admins.
>
> For example, say that a user on a mobile device is using Gmail in the office successfully during the day, but is blocked when they try to access Gmail at home in the evening. When remediation messages are enabled, they will see guidance on how to address the reason they are blocked.
>
> Remediation and custom messages are supported for access levels created in both Basic mode and Advanced mode. Also, they are supported for both Core Services and SAML apps.

See Ex. ~~12~~24, Allow Users to Unblock Apps With Remediation Messages in Context Aware Access, p. 1 (available at https://support.google.com/a/answer/11560430?sjid=18176077754207218303-EU) (last accessed mid Jan. 2025 and incorporated by reference). A quarantine message is also delivered to the unclean endpoint device notifying its user of the quarantine.

K.Mizra LLC, Ex. 2001, p. 174
Google LLC v. K.Mizra LLC
IPR2025-01437

Use remediation messages and custom messages to help your users unblock themselves

When blocked, your users can encounter:

- **Default message**—Displays if you have not added remediation messages or custom messages. An example default message is: **Your organization's policy is blocking access to this app**.
- **Remediation messages**—Replaces the default message. The messages are system generated, and correspond to the specific policy violation that blocked the user. Remediation messages can present several remediation options to the user, which they can expand by clicking **Show more options**. In the case of several remediation options, the user needs to complete the steps for any one of the available options to unblock themselves.
- **Custom message**—Adds specific help for the user, such as additional advice on getting unblocked or a helpful link to click. You add custom messages as needed. A custom message can appear in conjunction with the default message, or with remediation messages.

See id. Accordingly, ~~the Accused Zero Trust Network Security Instrumentalities meet~~CEP meets limitation E1 of Claim 19.

~~55.~~109. Limitation E2 requires that "preventing the first host from sending data to one or more other hosts associated with the protected network includes~~" "~~…in the event the service request comprises a DNS query, providing in response an IP address of a quarantine server configured to serve the quarantine notification page if a host name that is the subject of the DNS query is not associated with a remediation host configured to provide data usable to remedy the insecure condition." ~~The Accused Zero Trust Network Security Instrumentalities also meet all the requirements of limitation E2 of Claim 19. For example, Chrome Enterprise provides the user with~~CEP meets these requirements by providing to the Trusted Device a quarantine notification page containing links, or IP address(es), with resources (*i.e.* quarantine servers) configured to resolve the issue causing the quarantine. In other words, ~~Chrome Enterprise~~CEP provides remediation information to bring the ~~device~~Trusted Device into system security compliance so that it can access ~~the~~ protected ~~network.~~resources housed in the Google Cloud:

K.Mizra LLC, Ex. 2001, p. 175
Google LLC v. K.Mizra LLC
IPR2025-01437

Use remediation messages and custom messages to help your users unblock themselves

When blocked, your users can encounter:

- **Default message**—Displays if you have not added remediation messages or custom messages. An example default message is: **Your organization's policy is blocking access to this app**.
- **Remediation messages**—Replaces the default message. The messages are system generated, and correspond to the specific policy violation that blocked the user. Remediation messages can present several remediation options to the user, which they can expand by clicking **Show more options**. In the case of several remediation options, the user needs to complete the steps for any one of the available options to unblock themselves.
- **Custom message**—Adds specific help for the user, such as additional advice on getting unblocked or a helpful link to click. You add custom messages as needed. A custom message can appear in conjunction with the default message, or with remediation messages.

See id. Accordingly, ~~the Accused Zero Trust Network Security Instrumentalities meet~~CEP meets limitation E2 of Claim 19.

~~56.~~110. Limitation F requires "permitting the first host to communicate with the remediation host." ~~The Accused Zero Trust Network Security Instrumentalities meet~~CEP meets these requirements ~~as the Chrome Enterprise products allow~~by allowing a quarantined endpoint ~~device~~, i.e. Trusted Device, to access remediation resources to help make the device complaint~~.~~ with system safety requirements. *See id.* ~~Accordingly, the Accused Zero Trust Network Security Instrumentalities meet limitation F of Claim 19.~~

~~57.~~111. Google's acts of infringement have occurred within this District and elsewhere throughout the United States.

~~58.~~112. As a result of Google's infringing conduct, K.Mizra has suffered damages. Google is liable to K.Mizra in an amount that adequately compensates K.Mizra for Google's infringement in an amount that is no less than a to-be-calculated fully paid-up, lump-sum, reasonable royalty, together with interest and costs as fixed by this Court under 25 U.S.C. § 284.

K.Mizra LLC, Ex. 2001, p. 176
Google LLC v. K.Mizra LLC
IPR2025-01437

~~VII.~~ SECOND CLAIM FOR RELIEF
~~VIII.~~VI.
(Patent Infringement Under 35 U.S.C. § 271 of the ~~'048~~'120 Patent)

~~59.~~113. K.Mizra incorporates paragraphs 1 through ~~58~~112 as though fully set forth herein.

~~60.~~114. The ~~'048~~'120 Patent includes ~~20~~16 claims.

~~61.~~115. Google has directly infringed one or more claims of the ~~'048~~'120 Patent by making, importing, using, offering for sale, and/or selling the Accused ~~Zero Trust Network Security~~Hyperparameter Tuning Instrumentalities, in violation of 35 U.S.C. § 271(a).

~~62.~~116. Based on publicly available information, the Accused ~~Zero Trust Network Security~~Hyperparameter Tuning Instrumentalities satisfy every limitation of at least Claim ~~17~~1 of the ~~'048~~'120 Patent.

~~63.~~117. Claim ~~17~~1 of the ~~'048~~'120 Patent ~~recites the following~~states:

> [preamble] A ~~computer program product, the computer program product being embodied in a non transitory computer readable medium and comprising computer instructions for:~~
>
> ~~[A] detecting an insecure condition on a first host that has connected or is attempting to connect to a protected network,~~
>
> ~~[B] wherein detecting the insecure condition includes~~
>
> > ~~[B1] contacting a trusted computing base associated with a trusted platform module within the first host,~~
> >
> > ~~[B2] receiving a response, and~~ method of determining ~~whether the response includes a valid digitally signed attestation of cleanliness,~~
>
> ~~[C] wherein the valid digitally signed attestation~~hyperparameters of ~~cleanliness includes at least one attestation selected from the group consisting~~a classifier in a machine learning system by iteratively producing an estimate of ~~an attestation that the trusted computing base has ascertained that the first host is not infested, and an attestation that the trusted computing base has ascertained the presence~~a target hyperparameter vector, each iteration comprising the steps of ~~a patch or a patch~~

K.Mizra LLC, Ex. 2001, p. 177
Google LLC v. K.Mizra LLC
IPR2025-01437

level associated with a software component on the first host;:

[D] when it is determined that the response does not include a valid digitally signed attestation of cleanliness, quarantining the first host, including by preventing the first host from sending data to one or more other hosts associated with the protected network,

[E] wherein preventing the first host from sending data to one or more other hosts associated with the protected network includes

[E1] receiving a service request sent by the first host, determining whether the service request sent by the first host is associated with a remediation request, and when it is determined that the service request sent by the first host is not associated with a remediation request, serving a quarantine notification page that provides remediation information to the first host if the service request sent by the first host comprises a web server request,

[E2] wherein serving the quarantine notification page to the first host includes re-routing by responding to the service request sent by the first host with a redirect that causes a browser on the first host to be directed to a quarantine server configured to serve the quarantine notification page; and

[F] permitting the first host to communicate with the remediation host configured to provide data usable to remedy the insecure condition.

Ex. 2 at 22:35–23:9.

64. Claim 19 of the '705 Patent and its preamble and limitations are addressed above in Paragraphs 47–56.

65. The limitations of claim 17 of the '048 Patent are identical in many, if not most, respects to the limitations of Claim 19 of the '705 Patent.

66. The preamble and limitations [A] through [D] of the '048 Patent are met for the same reasons that the preamble and limitations [A] through [D] of the '705 Patent are met.

K.Mizra LLC, Ex. 2001, p. 178
Google LLC v. K.Mizra LLC
IPR2025-01437

67. The preambles and the identical, common limitations of the Asserted Patents (e.g., limitations [A] through [D]) are addressed above, and Paragraphs 48-53 above are therefore incorporated herein by reference with respect to Claim 17 of the '048 Patent.

68. Limitations [E1] and [E2] of the '705 Patent differ in some respects from limitations [E1] and [E2] of the '048 Patent, as seen above.

69. Limitation [E1] of claim 17 the '048 Patent provides for "receiving a service request sent by the first host, determining whether the service request sent by the first host is associated with a remediation request, and when it is determined that the service request sent by the first host is not associated with a remediation request, serving a quarantine notification page that provides remediation information to the first host if the service request sent by the first host comprises a web server request[.]"

70. Limitation [E1] of claim 17 of the '048 Patent is met by the Accused Zero Trust Network Security Instrumentalities, as demonstrated (for example) by the publicly available information below:

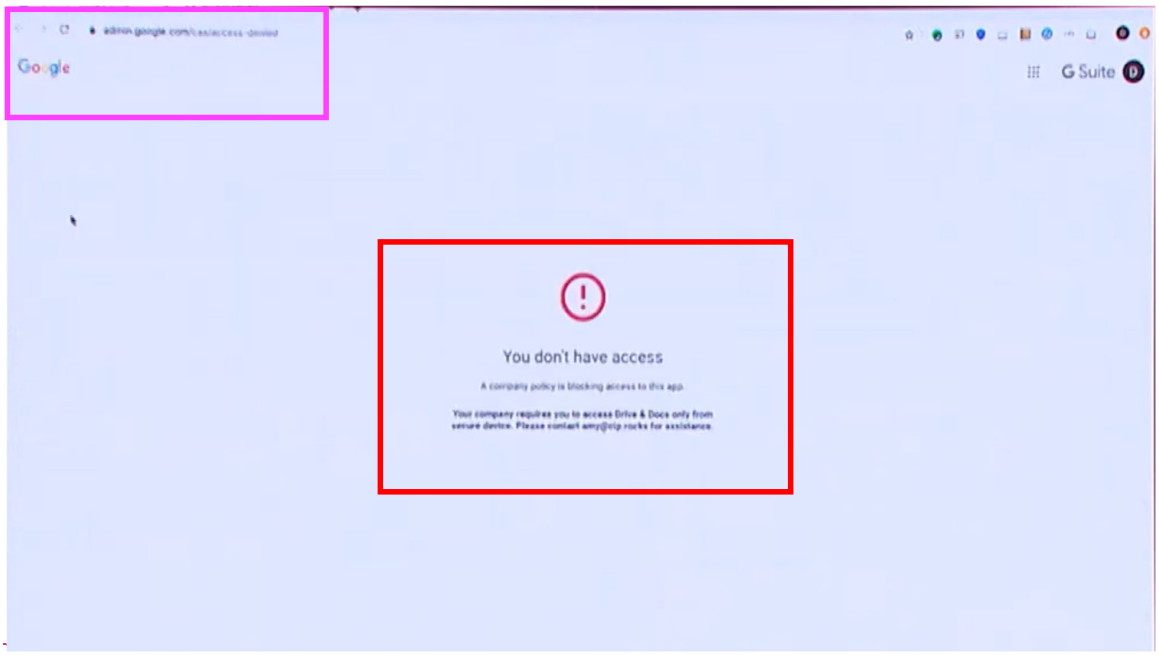
> Use remediation messages and custom messages to help your users unblock themselves
>
> When blocked, your users can encounter:
>
> - **Default message**—Displays if you have not added remediation messages or custom messages. An example default message is: **Your organization's policy is blocking access to this app**.
> - **Remediation messages**—Replaces the default message. The messages are system generated, and correspond to the specific policy violation that blocked the user. Remediation messages can present several remediation options to the user, which they can expand by clicking **Show more options**. In the case of several remediation options, the user needs to complete the steps for any one of the available options to unblock themselves.
> - **Custom message**—Adds specific help for the user, such as additional advice on getting unblocked or a helpful link to click. You add custom messages as needed. A custom message can appear in conjunction with the default message, or with remediation messages.

See Ex. 12 at 1.

K.Mizra LLC, Ex. 2001, p. 179
Google LLC v. K.Mizra LLC
IPR2025-01437

> This page shows you how to enable and use the Chrome Enterprise Premium Policy Remediator.
>
> When users attempt to access a Google Cloud resource but aren't compliant with the access policy for the resource, they are denied access and receive a general 403 error message. You can use the Policy Remediator to provide users with actionable steps that they can take to remediate their issue before reaching out to an admin for additional help. The specific remediation actions depend on the access policies, but can include things such as enabling screen lock, updating the operating system (OS) version, or accessing an app from a network allowed by your company.

See Ex. 13, Remediate Denied Access With the Policy Remediator, p. 1 (available at https://cloud.google.com/beyondcorp-enterprise/docs/policy-remediator?utm_source=chatgpt.com) (last accessed mid Jan. 2025 and incorporated by reference).



See Ex. 14, Video entitled "Enable Zero Trust Access Controls for your Web Apps, Infrastructure, and APIs (available at https://www.youtube.com/watch?v=2jz5FQpGWQ4 at approx. 13:22) (last accessed mid Jan. 2025 and incorporated by reference).

71. The Accused Zero Trust Network Security Instrumentalities determine that when a device (e.g., a first host computer) is non-compliant, the device is placed into a quarantined zone.

K.Mizra LLC, Ex. 2001, p. 180
Google LLC v. K.Mizra LLC
IPR2025-01437

~~The quarantined device may be allowed to make web service requests associated with remediation (e.g., a helpful link to click) while blocking service request associated with accessing protected network resources, and the Accused Zero Trust Network Security Instrumentalities can determine whether a request sent by the first host is associated with a remediation request.~~

~~72. Limitation [E2] of claim 17 of the '048 Patent provides that "wherein serving the quarantine notification page to the first host include re-routing by responding to the service request sent by the first host with a redirect that causes a browser on the first host to be directed to a quarantine server configured to serve the quarantine notification page[.]"~~

~~Limitation [E2] of claim 1 of the '048 Patent is met by the Accused Zero Trust Network Security Instrumentalities, as demonstrated (for example) by the publicly-~~ [A] drawing a random sample of hyperparameter vectors from a set of possible hyperparameter vectors,

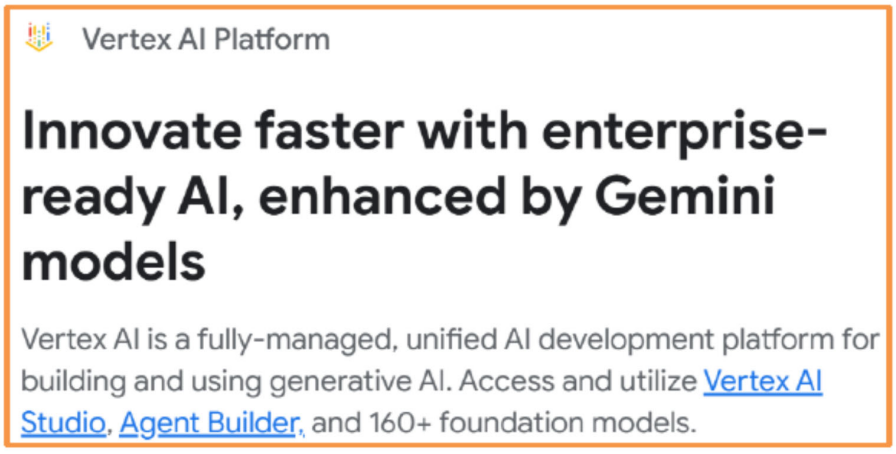
[B] updating the estimate of the target hyperparameter vector by using the random sample, and

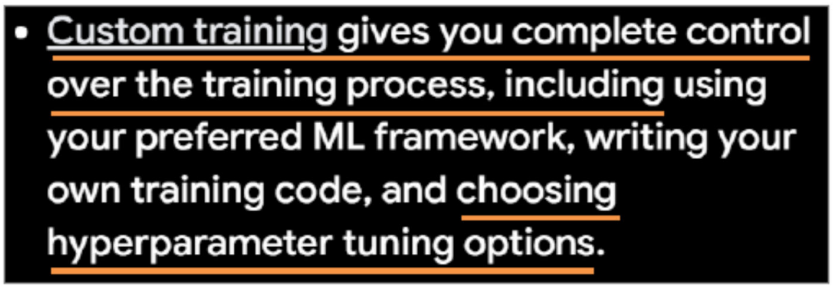
[C] Selecting, from the random sample of hyperparameter vectors, a hyperparameter vector producing a best result in the present and any previous iterations, and wherein the step of updating the estimate of the target hyperparameter vector uses said hyperparameter vector producing the best result.

Ex. 6 at 7:14–28.

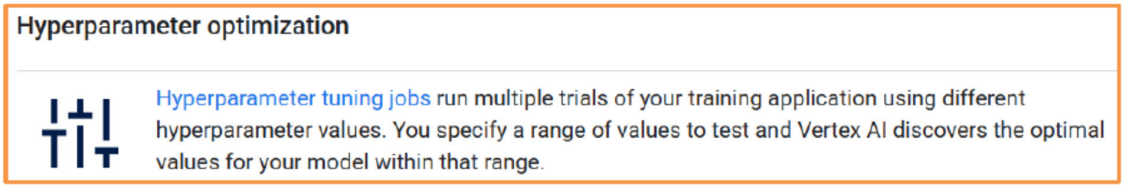
118. Regarding the preamble of Claim 1, and to the extent the preamble is determined to be limiting, the Asserted Hyperparameter Tuning Instrumentalities provide the features described in the preamble, which recites "A method of determining hyperparameters." For example, Google touts that its Vertex AI Platform is a "fully-managed, unified AI development platform for building and using generative AI."; "Custom training gives you complete control over the training process, including using your preferred ML framework, writing your own training code, and choosing hyperparameter tuning options."; and "Hyperparameter tuning jobs run multiple trials of your training application using different hyperparameter values."

K.Mizra LLC, Ex. 2001, p. 181
Google LLC v. K.Mizra LLC
IPR2025-01437





See Ex. 25, "Innovate faster with enterprise-ready AI, enhanced by Gemini models," (available at https://cloud.google.com/vertex-ai?hl=en (accessed May 21, 2025) and incorporated by reference).

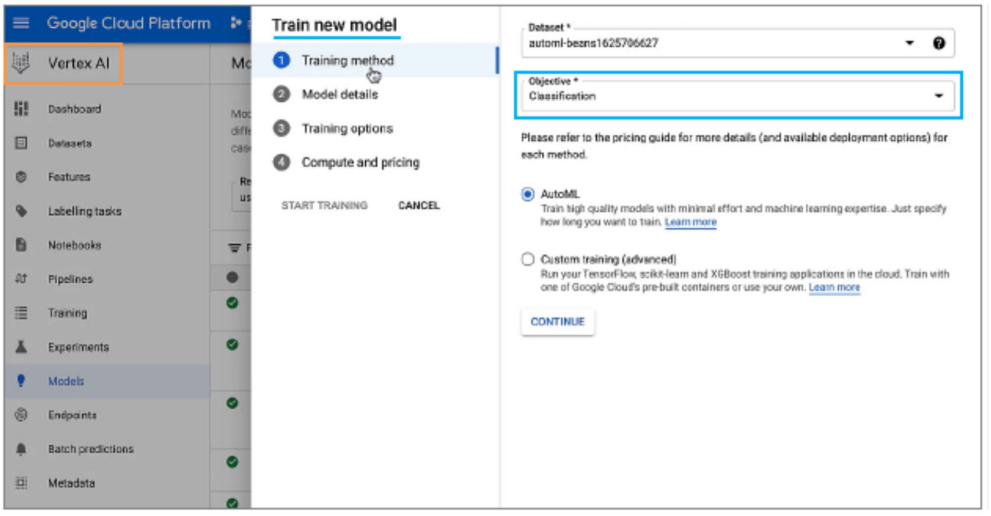


See Ex. 26, "Vertex AI custom training overview," (available at https://cloud.google.com/vertex-ai/docs/training/overview (accessed May 21, 2025) and incorporated by reference). Accordingly, and to the extent the preamble of Claim 1 is deemed limiting, the Accused Hyperparameter Tuning Instrumentalities meet the limitation.

119. The preamble also includes "of a classifier in a machine learning system".

> Vertex AI is a machine learning (ML) platform that lets you train and deploy ML models and AI applications. Vertex AI combines data engineering, data science, and ML engineering workflows, enabling team collaboration using a common toolset. Learn more.

See Ex. 27, "Vertex AI documentation," (available at https://cloud.google.com/vertex-ai/docs (accessed May 21, 2025) and incorporated by reference).



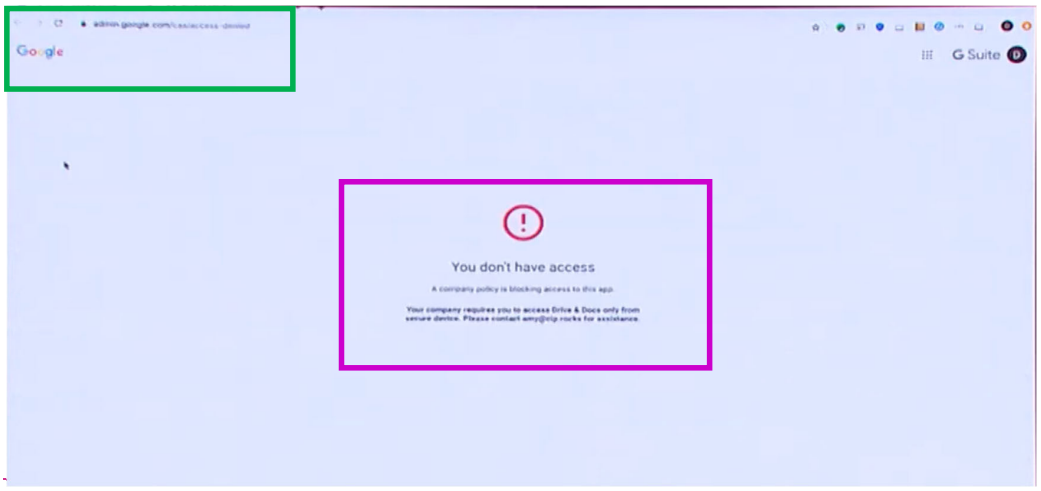
See screenshot above of page at https://www.youtube.com/watch?v=8hZ_cBwNOss (last accessed mid-May 2025) and incorporated by reference). Thus, and again only to the extent that this portion of the preamble of Claim 1 is deemed limiting, the Accused Hyperparameter Tuning Instrumentalities meet the limitation.

120. The preamble further includes "by iteratively producing an estimate of a target hyperparameter vector":

How hyperparameter tuning works

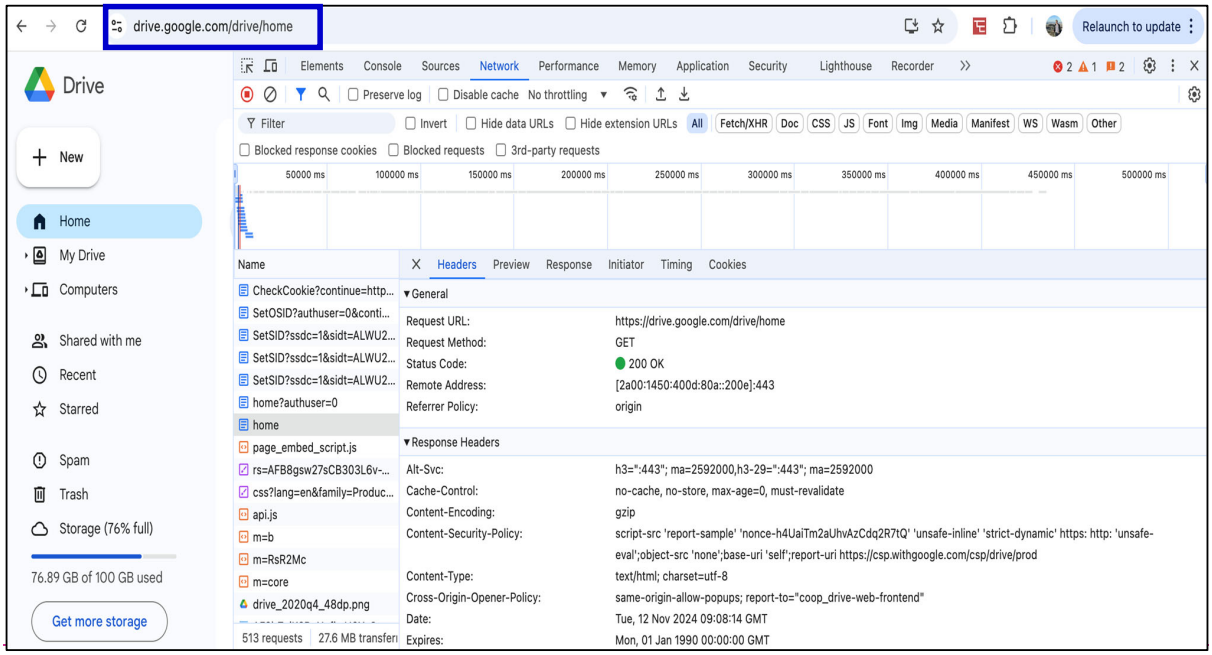
Hyperparameter tuning works by running multiple *trials* of your training application with values for your chosen hyperparameters, set within limits you specify. Vertex AI keeps track of the results of each trial and makes adjustments for subsequent trials. When the job is finished, you can get a summary of all the trials along with the most effective configuration of values according to the criteria you specify.

~~73.~~ *See* Ex. 28, "Overview of hyperparameter tuning," (available ~~information below:~~

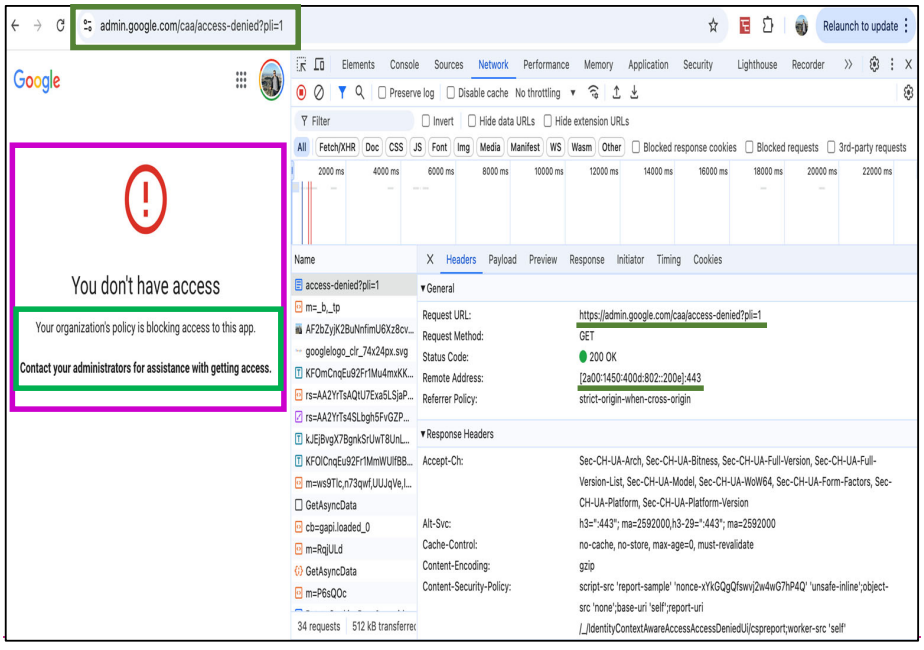


~~*See Id.*~~

K.Mizra LLC, Ex. 2001, p. 184
Google LLC v. K.Mizra LLC
IPR2025-01437



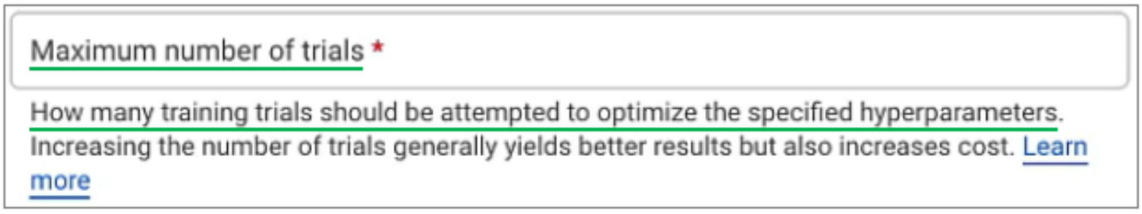
~~*See* Ex. 15, Screenshot of page at https://drive.google.com/drive/home (last accessed early Dec. 2024 and incorporated by reference).~~



~~*See* Ex. 16, Screenshot of page at https://admin.google.com/caa/access-denied?pli=1 (last accessed early Dec. 2024~~at https://cloud.google.com/vertex-ai/docs/training/hyperparameter-tuning-overview (accessed May 21, 2025) and incorporated by reference).

~~74. As seen above, a user can be blocked from accessing network resources. In such an instance, a notification page stating "You don't have access," accompanied by a remediation message, is displayed. A remediation message and guidance on how to bring the device into compliance is provided. As depicted above, serving a quarantine page to a first host computer includes re-routing by responding to a service request with a redirect that causes a browser on the first host to be directed to a quarantine server configured to serve a quarantine notification page.~~

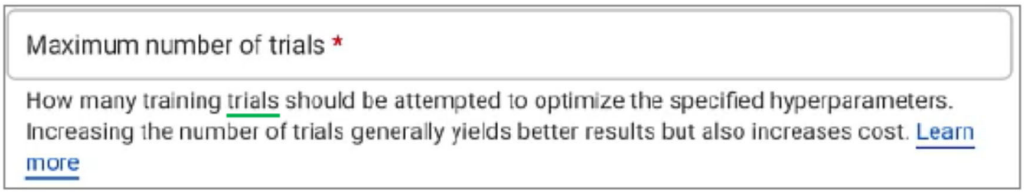
~~75. Limitation [F] of claim 17 of the '048 Patent is met for the same reasons as described in Paragraph 56 above (addressing limitation [F] of claim 19 of the '705 Patent).~~



See screenshot above of page at https://www.youtube.com/watch?v=8hZ_cBwNOss (last accessed mid-May 2025) and incorporated by reference). Accordingly, and to the extent this portion of the preamble of Claim 1 is deemed limiting, the Accused Hyperparameter Tuning Instrumentalities meet the limitation.

121. The preamble further also includes "each iteration comprising the steps of:"

K.Mizra LLC, Ex. 2001, p. 186
Google LLC v. K.Mizra LLC
IPR2025-01437



Id. Again, then, to the extent this portion of the preamble of Claim 1 is deemed limiting, the Accused Hyperparameter Tuning Instrumentalities meet the limitation.

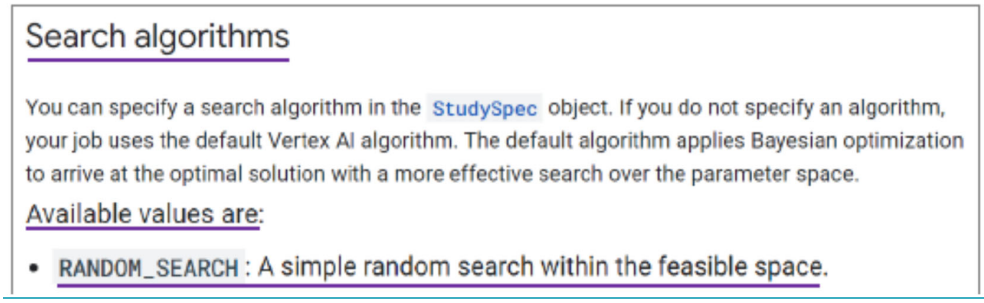
122. Accordingly, the Accused ~~Zero Trust Network Security~~Hyperparameter Tuning Instrumentalities meet ~~all~~ limitation of the ~~limitations~~preamble of Claim 1.

123. Limitations A and ~~therefore infringe~~B of Claim 1 require:

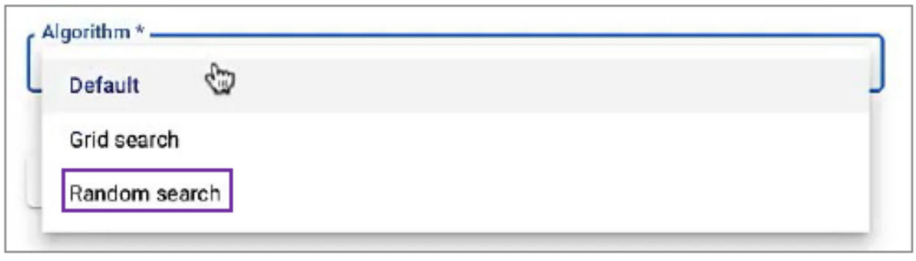
[A] drawing a random sample of hyperparameter vectors from a set of possible hyperparameter vectors,

[B] updating the estimate of the target hyperparameter vector by using the random sample, and

Ex. 6 at 7:19–22. Google has disclosed:



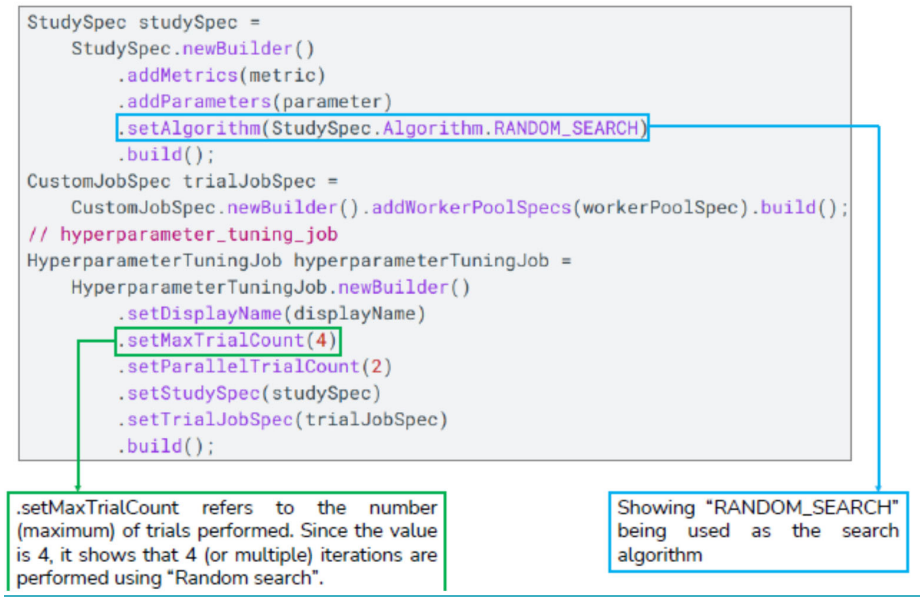
See Ex. 28.



K.Mizra LLC, Ex. 2001, p. 187
Google LLC v. K.Mizra LLC
IPR2025-01437

See screenshot above of page at ~~least Claim 17 of the '048 Patent~~https://www.youtube.com/watch?v=8hZ_cBwNOss (last accessed mid-May, 2025) and incorporated by reference).

124. Limitation A requires "drawing a random sample of hyperparameter vectors from a set of possible hyperparameter vectors" and the Google system meets this requirement:



See Ex. 29, "Create a hyperparameter tuning job for python package," (available at https://cloud.google.com/vertex-ai/docs/samples/aiplatform-create-hyperparameter-tuning-job-python-package-sample?hl=en (accessed May 21, 2025) and incorporated by reference).

125. Accordingly, the Accused Hyperparameter Tuning Instrumentalities meet Limitation A of Claim 1.

126. Limitation B requires "updating the estimate of the target hyperparameter vector by using the random sample". Google has stated:

K.Mizra LLC, Ex. 2001, p. 188
Google LLC v. K.Mizra LLC
IPR2025-01437

Random search: set up a grid of hyperparameter values and select *random* combinations to train the model and score. The number of search iterations is set based on time/resources.

Random search is surprisingly efficient compared to grid search. Although grid search will find the optimal value of hyperparameters (assuming they are in your grid) eventually, random search will usually find a "close-enough" value in far fewer iterations. This great paper explains why this is so: grid search spends too much time evaluating unpromising regions of the hyperparameter search space because it has to evaluate every single combination in the grid. Random search in contrast, does a better job of exploring the search space and therefore can usually find a good combination of hyperparameters in far fewer iterations.

See Ex. 30, excerpts of "Intro to Model Tuning: Grid and Random Search," (available at https://www.kaggle.com/code/willkoehrsen/intro-to-model-tuning-grid-and-random-search (last accessed mid mid-May, 2025) and incorporated by reference).

Randomized search is another method for hyperparameter optimization that can be more efficient than grid search in some cases. With randomized search, instead of specifying a list of values for each hyperparameter, you specify a distribution for each hyperparameter. The randomized search algorithm will then sample values for each hyperparameter from its corresponding distribution and train a model using the sampled values. This process is repeated a specified number of times, and the optimal values for the hyperparameters are chosen based on the performance of the models.

See Ex. 31, "Comparing Randomized Search and Grid Search for Hyperparameter Estimation in Scikit Learn," (available at https://www.geeksforgeeks.org/comparing-randomized-search-and-grid-search-for-hyperparameter-estimation-in-scikit-learn/ (accessed May 21, 2025) and incorporated by reference).

127. Accordingly, the Accused Hyperparameter Tuning Instrumentalities also meet Limitation B of Claim 1.

K.Mizra LLC, Ex. 2001, p. 189
Google LLC v. K.Mizra LLC
IPR2025-01437

128. Limitation C requires "selecting, from the random sample of hyperparameter vectors, a hyperparameter vector producing a best result in the present and any previous iterations, and wherein the step of updating the estimate of the target hyperparameter vector uses said hyperparameter vector producing the best result." Google describes:

> How hyperparameter tuning works
>
> Hyperparameter tuning works by running multiple *trials* of your training application with values for your chosen hyperparameters, set within limits you specify. Vertex AI keeps track of the results of each trial and makes adjustments for subsequent trials. When the job is finished, you can get a summary of all the trials along with the most effective configuration of values according to the criteria you specify.

See Ex. 28.

~~76.~~129. Accordingly, the Accused Hyperparameter Tuning Instrumentalities meet Limitation C of Claim 1.

~~77.~~130. Google's acts of infringement have occurred within this District and elsewhere throughout the United States.

~~78.~~131. As a result of Google's infringing conduct, K.Mizra has suffered damages. Google is liable to K.Mizra in an amount that adequately compensates K.Mizra for Google's infringement in an amount that is no less than a to be calculated fully paid-up, lump-sum, reasonable royalty, together with interest and costs as fixed by this Court under 25 U.S.C. § 284.

VII. THIRD CLAIM FOR RELIEF
(Patent Infringement Under 35 U.S.C. § 271 of the '717 Patent)

132. K.Mizra incorporates paragraphs 1 through 131 as though fully set forth herein.

133. The '717 Patent includes 20 claims.

134. Google has directly infringed one or more claims of the '717 Patent by making, importing, using, offering for sale, and/or selling the Accused Mesh Networking Instrumentalities, in violation of 35 U.S.C. § 271(a).

K.Mizra LLC, Ex. 2001, p. 190
Google LLC v. K.Mizra LLC
IPR2025-01437

135. Based on publicly available information, the Accused Mesh Networking Instrumentalities satisfy every limitation of at least Claim 11 of the '717 Patent.

136. Open Thread is an open-source implementation of the Thread networking protocol released by Google. A description of Open Thread is provided at www.openthread.io. On information and belief, openthread.io is owned and operated by Google.

137. On information and belief, Open Thread is an implementation of the Thread networking technology used in Google Nest products.



OpenThread released by Google is an open-source implementation of Thread® (http://threadgroup.org). Google has released OpenThread to make the networking technology used in Google Nest products more broadly available to developers, in order to accelerate the development of products for the connected home and commercial buildings.

See Ex. 32, Homepage of openthread.io (available at https://openthread.io/ (accessed May 23, 2025) and incorporated by reference).

138. Claim 11 of the '717 Patent states:

[preamble] A station for use in a wireless communication network, the station comprising:

[A] a transmitter arrangement, a receiver arrangement, and a processor circuit coupled to the transmitter arrangement and the receiver arrangement,

K.Mizra LLC, Ex. 2001, p. 191
Google LLC v. K.Mizra LLC
IPR2025-01437

[B] the processor configured to start up in a not-associated state and to cause the transmitter arrangement to transmit association request messages in the not-associated state,

[C] the processor circuit being configured to switch to an associated state upon reception at the receiver arrangement of an association grant in response to one of the association request messages,

[D] the processor circuit being configured to establish an associated route for communication with an association unit during subsequent communication of operating messages to an association unit,

[E] the associated route including a source of the association grant and, if the source of the association grant is not the association unit, the operating route between the source of the association grant and the association unit previously established for the source of the association grant in response to association request messages and/or association grants for intermediate stations,

[F] the processor circuit being configured to cause the transmitter arrangement to initiate and transmit association grants in response to association request messages, but only after switching to the associated state.

139. Regarding the preamble of Claim 11, and to the extent the preamble is determined to be limiting, the Accused Mesh Networking Instrumentalities provide the features described in the preamble, which recites "A station for use in a wireless communication network." For example, Google devices including the Nest WiFi, the Nest Hub Max, and the Nest Hub (2nd Gen) are stations configured for use in a wireless network implementing the Thread networking protocol.

K.Mizra LLC, Ex. 2001, p. 192
Google LLC v. K.Mizra LLC
IPR2025-01437



A more reliable smart home network with Thread.

Thread is the latest technology for keeping all your smart home devices connected. Nest Hub Max and Nest Hub (2nd gen) have a built-in Thread border router to connect your Thread mesh network of smart devices to your Wi-Fi network. Your Thread-enabled network will be faster, use less energy, and reach more places throughout and around your home.

Ex. 33, "Nest Hub Smart Displays for your Home – Google Store," (available at https://store.google.com/category/nest_hubs_displays (accessed May 12, 2025) and incorporated by reference).

Thread devices join users' existing home networks through a Thread Border Router. Just like a Wi-Fi router can bridge Wi-Fi and Ethernet devices into a single network, a Thread Border Router allows Thread devices to become part of users' networks.

Google devices like the Nest WiFi, Google Nest Hub Max and Google Nest Hub (2nd gen) have Thread radios built-into them and act as Thread Border Routers.

Ex. 8.

140. Similarly, the Nest Connect is a station configured for use in a Thread wireless network.

In an emailed statement, Grant Erickson, president of the Thread Group said, "Nest Detect relies on Thread's low power architecture to last for long periods of time on one battery. Nest Connect is a Thread router, which seamlessly extends both the reach and connectivity of the Thread network. Nest Guard provides seamless IP-based routing between Thread, Wi-Fi, and cellular, and their interoperability with the Yale Linus Lock is the posterchild of Thread's benefits in action."

Ex. 11.

K.Mizra LLC, Ex. 2001, p. 193
Google LLC v. K.Mizra LLC
IPR2025-01437

141. Accordingly, and to the extent the preamble of Claim 11 is deemed limiting, the Accused Mesh Networking Instrumentalities meet the limitation.

142. Limitation A of Claim 11 of the Asserted Mesh Networking Patent requires a transmitter arrangement, a receiver arrangement, and a processor circuit coupled to the transmitter arrangement and the receiver arrangement.

143. The Accused Mesh Networking Instrumentalities include a Thread radio with a transmitter arrangement and a receiver arrangement, as well as a processor circuit coupled to the transmitter arrangement and receiver arrangement that processes transmissions and receptions. For example, Google's documents confirm that each of the Nest WiFi, the Google Hub Max, and the Google Nest Hub (2nd Gen) include Thread radios.

Thread has the following key benefits:

- IPv6 based: Thread devices can join the same network as your other devices, and talk directly to each other and the cloud.

Thread devices join users' existing home networks through a Thread Border Router. Just like a Wi-Fi router can bridge Wi-Fi and Ethernet devices into a single network, a Thread Border Router allows Thread devices to become part of users' networks.

Google devices like the Nest WiFi, Google Nest Hub Max and Google Nest Hub (2nd gen) have Thread radios built-into them and act as Thread Border Routers.

Ex. 8. Similarly, a statement from the Thread Group (of which Google is a member) confirms that the Nest Connect is a Thread router.

In an emailed statement, Grant Erickson, president of the Thread Group said, "Nest Detect relies on Thread's low power architecture to last for long periods of time on one battery. Nest Connect is a Thread router, which seamlessly extends both the reach and connectivity of the Thread network. Nest Guard provides seamless IP-based routing between Thread, Wi-Fi, and cellular, and their interoperability with the Yale Linus Lock is the posterchild of Thread's benefits in action."

Ex. 11. Indeed, in order to communicate with other devices, each Thread device (including the Accused Mesh Networking Instrumentalities) must necessarily include a transmitter arrangement

K.Mizra LLC, Ex. 2001, p. 194
Google LLC v. K.Mizra LLC
IPR2025-01437

(to transmit messages from the Thread device to other devices in the Thread network) and a receiver arrangement (to receive communications, including responses to its transmitted messages, from other devices in the Thread network). Thread devices, including the Accused Mesh Networking Instrumentalities, must also include a processor coupled to both the transmitter arrangement and the receiver arrangement to, among other things, initiate transmissions, respond to messages sent by other devices in the Thread network, and process received messages. The operations of the processor are described in further detail in connection with Limitations B through F.

144. Accordingly, the Accused Mesh Instrumentalities meet Limitation A of Claim 11.

145. Limitation B of Claim 11 of the Asserted Mesh Networking Patent requires the processor is configured to start up in a not-associated state and to cause the transmitter arrangement to transmit association request messages in the not-associated state.

146. The Accused Mesh Networking Instrumentalities include this feature. For example, Thread devices, such as the Accused Mesh Networking Devices, have a Detached state when the Thread device has not yet attached to a Thread network.

Note: The Detached state is observed when Thread is enabled but the device is not attached to a Thread network. The Disabled state is observed when Thread is disabled on the device.

Ex. 34, "Developing with OpenThread APIs, (available at https://openthread.io/codelabs/openthread-apis#1 (accessed May 23, 2025) and incorporated by reference).

147. In order to attach to a Thread network, a Thread device such as the Accused Mesh Networking Instrumentalities goes through the Attach process.

K.Mizra LLC, Ex. 2001, p. 195
Google LLC v. K.Mizra LLC
IPR2025-01437

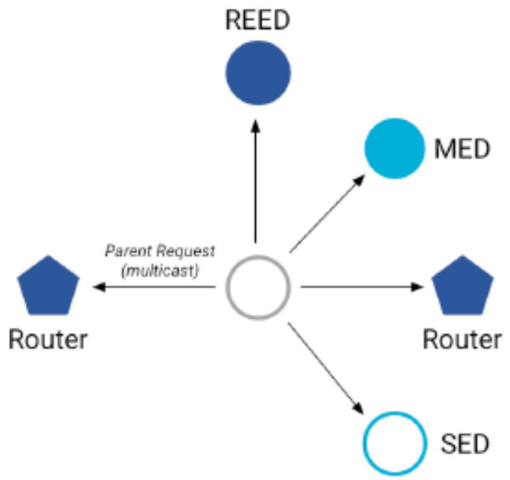
Join an existing network

If the device elects to join an existing network, it configures its Channel, PAN ID, XPAN ID, and Network Name to match that of the target network via Thread Commissioning, then goes through the MLE Attach process to attach as a Child (End Device). This process is used for Child-Parent links.

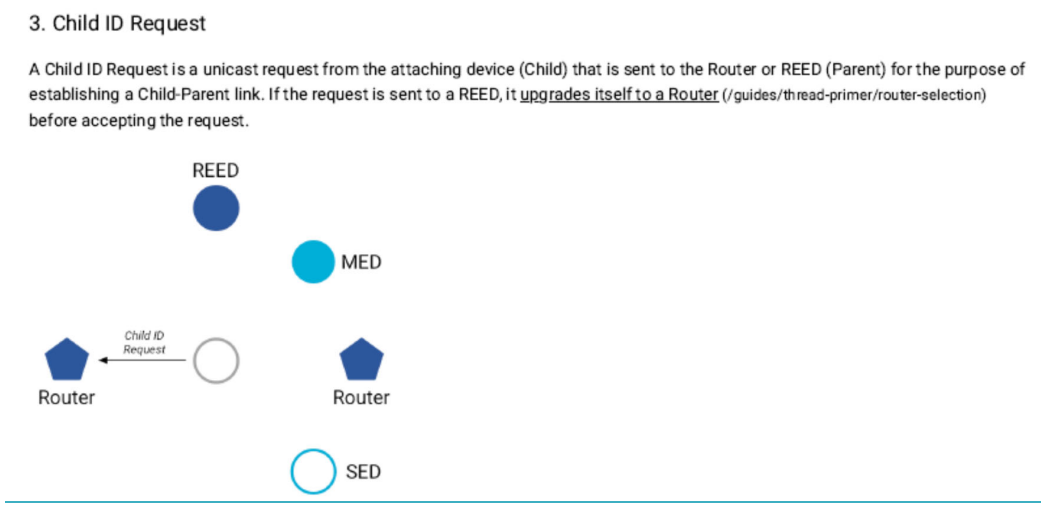
Ex. 35, "Network Discovery and Formation," (available at https://openthread.io/guides/thread-primer/network-discovery (accessed May 20, 2025) and incorporated by reference). As the first step in this process, an attaching device sends a Parent Request message to find neighboring devices that are part of the Thread network.

1. Parent Request

A Parent Request is a multicast request from the attaching device that is used to discover neighboring Routers and Router Eligible End Devices (REEDs) in the target network.



Id. Upon receiving a response, the Thread device sends a message called the Child ID Request to a particular device in the Thread network for the purpose of establishing a link with that device to join the network.



Id. The attaching device may also be referred to as the Child, and the router with which the attaching device is seeking to establish a link may be referred to as the Parent.

148. Accordingly, the Accused Mesh Networking Instrumentalities meet Limitation B of Claim 11.

149. Limitation C of Claim 11 of the Asserted Mesh Networking Patent requires the processor circuit being configured to switch to an associated state upon reception at the receiver arrangement of an association grant in response to one of the association request messages.

150. Thread devices, such as the Accused Mesh Networking Instrumentalities, become attached to a Thread network through the MLE Attach process. As described above in connection with Limitation B, the attaching device begins the process by sending a Parent Request and then exchanges further messages in order to establish a link with a device in the existing Thread network.

K.Mizra LLC, Ex. 2001, p. 197
Google LLC v. K.Mizra LLC
IPR2025-01437

Join an existing network

If the device elects to join an existing network, it configures its Channel, PAN ID, XPAN ID, and Network Name to match that of the target network via Thread Commissioning, then goes through the MLE Attach process to attach as a Child (End Device). This process is used for Child-Parent links.

Key Point: Every device, router-capable or not, initially attaches to a Thread network as a Child (End Device).

1. The Child sends a multicast Parent Request (#1_parent_request) to all neighboring Routers and REEDs in the target network.
2. All neighboring Routers and REEDs (if the Parent Request Scan Mask includes REEDs) send Parent Responses (#2_parent_response) with information about themselves.
3. The Child chooses a Parent device and sends a Child ID Request (#3_child_id_request) to it.
4. The Parent sends a Child ID Response (#4_child_id_response) to confirm link establishment.

Id. When the link has been established, the Thread device has attached to the network and thus is in an Attached state. The establishment of the link between the attaching device and the Thread device in the existing Thread network is confirmed by the Child ID Response.

4. Child ID Response

A Child ID Response is a unicast response from the Parent that is sent to the Child to confirm that a Child-Parent link has been established.

Id. Notably, Thread devices such as the Accused Mesh Networking Instrumentalities may exchange information about how long they have been in certain states or roles, suggesting that these devices track their current state/role and when each state/role is entered.

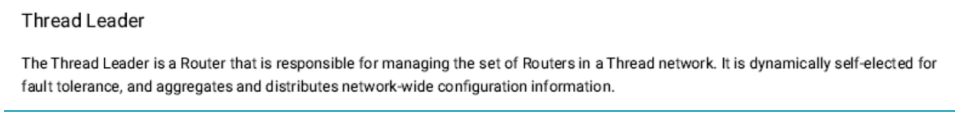
mChildRole (#structot_network_diag_mle_counters_1a33a2daf397cc144009b36dacdaa3c5de)	**uint16_t** Number of times device child role.
mChildTime (#structot_network_diag_mle_counters_1a01062bd8384fe1e360015373a45c68f1)	**uint64_t** Milliseconds device has child role.
mDetachedRole (#structot_network_diag_mle_counters_1a6d1b17ab4d3469693ff19fb596de7661)	**uint16_t** Number of times device detached role.
mDetachedTime (#structot_network_diag_mle_counters_1a75fde6d44ca4db581050d6a71a31bbee)	**uint64_t** Milliseconds device has detached role.

Ex. 36, "otNetworkDiagMleCounters," (available at https://openthread.io/reference/struct/ot-network-diag-mle-counters (accessed May 23, 2025) and incorporated by reference).

151. Accordingly, the Accused Mesh Networking Instrumentalities meet Limitation C of Claim 11.

152. Limitation D of Claim 11 of the Accused Mesh Networking Instrumentalities requires the processor circuit being configured to establish an associated route for communication with an association unit during subsequent communication of operating messages to an association unit.

153. Each of the Accused Mesh Networking Instrumentalities is configured to operate in a Thread network. A Thread network includes a device designated as the Leader.


Thread Leader

The Thread Leader is a Router that is responsible for managing the set of Routers in a Thread network. It is dynamically self-elected for fault tolerance, and aggregates and distributes network-wide configuration information.

Ex. 37, "Node Roles and Types," (available at https://openthread.io/guides/thread-primer/node-roles-and-types (accessed May 20, 2025) and incorporated by reference). The processors of the Accused Mesh Networking Devices are configured to store information about the route between the attaching device and the Leader as further discussed below in connection with Limitation E of Claim 11.

154. Accordingly, the Accused Mesh Networking Instrumentalities meet Limitation D of Claim 11.

155. Limitation E of Claim 11 requires the associated route includes a source of the association grant and, if the source of the association grant is not the association unit, the operating route between the source of the association grant and the association unit previously established

K.Mizra LLC, Ex. 2001, p. 199
Google LLC v. K.Mizra LLC
IPR2025-01437

for the source of the association grant in response to association request messages and/or association grants for intermediate stations.

156. Upon joining a Thread network, Thread devices such as the Accused Mesh Networking Devices receive a Child ID Response which provides information about the route between the Leader and the Thread device. For example, the Child ID Response includes the Source Address (i.e., the address for the Parent Thread device), the Leader RLOC (i.e., the address for the Leader of the Thread network), Network Data such as "more-specific routes," and Route information.

Child ID Response Message Contents

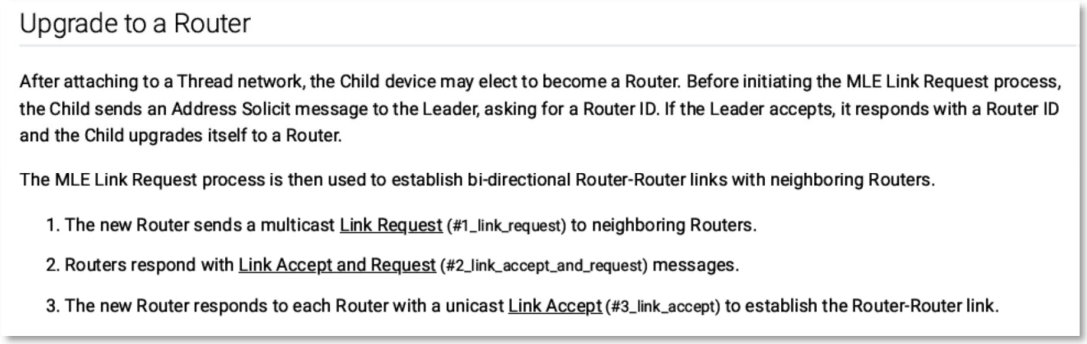
Source Address	Parent's RLOC16
Address16	Child's RLOC16
Leader Data	Information about the Parent's Leader (RLOC, Partition ID, Partition weight)
Network Data	Information about the Thread network (on-mesh prefixes, address autoconfiguration, more-specific routes)
Route (REED only)	Route propagation
Timeout	Inactivity duration before the Parent removes the Child
Address Registration (MEDs and SEDs only)	Confirm registered addresses

Ex. 35. K.Mizra further expects that discovery, including review of source code, will provide further information and confirm this understanding.

157. Accordingly, the Accused Mesh Networking Instrumentalities meet Limitation D of Claim 11.

158. Limitation E of Claim 11 of the Asserted Mesh Networking Patent requires the processor circuit being configured to cause the transmitter arrangement to initiate and transmit association grants in response to association request messages, but only after switching to the associated state.

159. Thread devices such as the Accused Mesh Networking Instrumentalities can send Parent Response and Child ID Response communications once attached to the Thread network, but will not send such responses if the device is disconnected from the network. For example, upon joining a Thread network, Router Eligible End Devices (or REEDs) can upgrade to become routers.



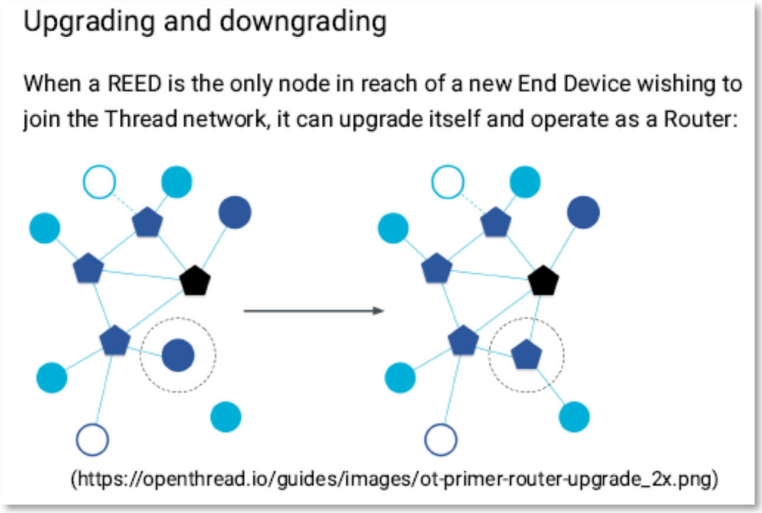
Upgrade to a Router

After attaching to a Thread network, the Child device may elect to become a Router. Before initiating the MLE Link Request process, the Child sends an Address Solicit message to the Leader, asking for a Router ID. If the Leader accepts, it responds with a Router ID and the Child upgrades itself to a Router.

The MLE Link Request process is then used to establish bi-directional Router-Router links with neighboring Routers.

1. The new Router sends a multicast Link Request (#1_link_request) to neighboring Routers.
2. Routers respond with Link Accept and Request (#2_link_accept_and_request) messages.
3. The new Router responds to each Router with a unicast Link Accept (#3_link_accept) to establish the Router-Router link.

Ex. 38, "Router Selection," (available at https://openthread.io/guides/thread-primer/router-selection (accessed May 27, 2025 and incorporated by reference).



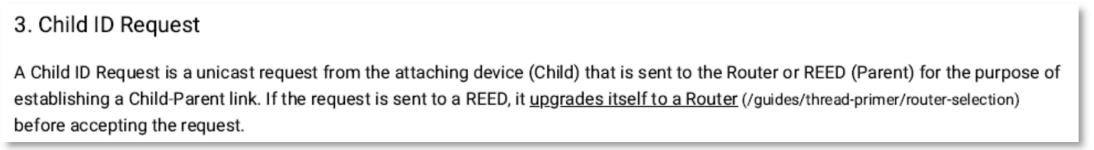
Upgrading and downgrading

When a REED is the only node in reach of a new End Device wishing to join the Thread network, it can upgrade itself and operate as a Router:

(https://openthread.io/guides/images/ot-primer-router-upgrade_2x.png)

Ex. 37. Upon upgrading to a router, the Thread device is capable of accepting a request from an attaching device to join the Thread network.

K.Mizra LLC, Ex. 2001, p. 201
Google LLC v. K.Mizra LLC
IPR2025-01437


3. Child ID Request

A Child ID Request is a unicast request from the attaching device (Child) that is sent to the Router or REED (Parent) for the purpose of establishing a Child-Parent link. If the request is sent to a REED, it upgrades itself to a Router (/guides/thread-primer/router-selection) before accepting the request.

Ex. 35.

160. Accordingly, the Accused Mesh Networking Instrumentalities meet limitation of the Limitation E of Claim 11.

161. Google's acts of infringement have occurred within this District and elsewhere throughout the United States.

162. As a result of Google's infringing conduct, K.Mizra has suffered damages. Google is liable to K.Mizra in an amount that adequately compensates K.Mizra for Google's infringement in an amount that is no less than a to-be-calculated fully paid-up, lump-sum, reasonable royalty, together with interest and costs as fixed by this Court under 25 U.S.C. § 284.

~~IX.~~VIII. REQUEST FOR RELIEF

WHEREFORE, K.Mizra respectfully requests the Court find in its favor and against Google, and that the Court grant K.Mizra at least the following relief:

A. Judgment that Google has infringed, literally and/or under the doctrine of equivalents, one or more claims of the Asserted Patents;

B. Awarding damages to K.Mizra in an amount to be proven at trial and in the form of a fully paid-up, lump sum, reasonable royalty that ~~takes into account~~considers and runs through expiration of all the Asserted Patents;

C. Awarding enhanced damages, as appropriate, under 35 U.S.C. § 284;

K.Mizra LLC, Ex. 2001, p. 202
Google LLC v. K.Mizra LLC
IPR2025-01437

D. Awarding K.Mizra's costs (including internal and external costs and disbursements) and declaring this an exceptional case and awarding K.Mizra its attorneys' fees in accordance with 35 U.S.C. § 285;

E. Pre-judgment and post-judgment interest at the maximum rate permitted by law on the damages caused by reason of Google's infringing activities and other conduct complained of herein; and

F. Awarding such other and further relief as this Court deems just and proper under the circumstances.

~~X.~~IX. DEMAND FOR JURY TRIAL

Pursuant to Fed. R. Civ. P. 38(b), K.Mizra hereby demands a trial by jury on all issues so triable.

Dated: ~~February 18, 2025~~

Respectfully submitted,

By: /s/ ~~*Michael C. Smith*~~DRAFT

Michael C. Smith
Texas Bar No. 18650410
michael.smith@solidcounsel.com
Scheef & Stone, LLP
113 E. Austin Street
Marshall, TX 75670
(903) 938-8900

Robert R. Brunelli~~*~~ (admitted *pro hac vice*)
CO State Bar No. 20070
rbrunelli@sheridanross.com
~~Scott R. Bialecki*~~
~~CO State Bar No. 23103~~
~~sbialecki@sheridanross.com~~
Bart A. Starr~~*~~ (admitted *pro hac vice*)
CO State Bar No. 50446
bstarr@sheridanross.com
Brian S. Boerman~~*~~ (admitted *pro hac vice*)
CO State Bar No. 50834
bboerman@sheridanross.com

K.Mizra LLC, Ex. 2001, p. 203
Google LLC v. K.Mizra LLC
IPR2025-01437

Gene Volchenko* (admitted *pro hac vice*)
IL State Bar No. 6342818
gvolchenko@sheridanross.com
SHERIDAN ROSS P.C.
1560 Broadway, Suite 1200
Denver, CO 80202
Telephone: 303-863-9700
Facsimile: 303-863-0223
litigation@sheridanross.com

Of Counsel:
Claire A. Henry
Texas Bar No. 24053063
Miller Fair Henry PLLC
1507 Bill Owens Parkway
Longview, TX 75604
Telephone: (903) 757-6400
Facsimile: (903) 757-2323
claire@millerfairhenry.com

~~*Pro hac vice pending~~
Attorneys for Plaintiff K.Mizra LLC

K.Mizra LLC, Ex. 2001, p. 204
Google LLC v. K.Mizra LLC
IPR2025-01437

CERTIFICATE OF SERVICE

I hereby certify that a true and correct copy of the above and foregoing document has been served on _____ upon all counsel of record who are deemed to have consented to electronic service via the Court's CM/ECF system.

/s/ DRAFT
Lori R. Brown
lbrown@sheridanross.com
SHERIDAN ROSS P.C.
1560 Broadway, Suite 1200
Denver, CO 80202
Telephone: 303-863-9700
Facsimile: 303-863-0223

K.Mizra LLC, Ex. 2001, p. 205
Google LLC v. K.Mizra LLC
IPR2025-01437

EXHIBIT 6

[redacted]

From: Michael Smith <Michael.Smith@Solidcounsel.com>
Sent: Monday, June 9, 2025 9:46 AM
To: Shaun Hassett
Cc: Mike Jones
Subject: Re: Mizra - amended pleadings

Shaun,

Update. K. Mizra has decided against amending their complaint at this time - so we can go forward with getting the schedule on file under the extended schedule. Let me know if you have any questions or need anything else on the existing schedule proposal.

Michael C. Smith PARTNER

SCHEEF & STONE

Michael.Smith@solidcounsel.com | 903.938.8900

Marshall office: 113 East Austin Street, Marshall, Texas 75670
Dallas office: 500 N. Akard Street, Suite 2700, Dallas, Texas 75201
Frisco office: 2600 Network Boulevard, Suite 400, Frisco, Texas 75034

Eastern District Texas Blog - www.EDTexweblog.com
Texas IP Law Blog - www.TexIPLawBlog.com
Texas Business Courts Blog - www.TexasBusinessCourtsBlog.com

ABOTA FOUNDATION SENIOR LIFE FELLOW ★ ★ ★

TEXAS BAR FOUNDATION LIFE FELLOW

Important: This email and any attached files contain information intended for the exclusive use of the person(s) to whom it is addressed and may contain information that is confidential. If you are neither the intended recipient nor responsible for delivering this email to the recipient, any dissemination or copying of this communication is strictly prohibited. Please notify the sender of any unintended recipients and delete the original message without making any copies.

From: Michael Smith <Michael.Smith@Solidcounsel.com>
Sent: Friday, June 6, 2025 6:31 PM
To: Shaun Hassett <shaunhassett@potterminton.com>
Cc: Mike Jones <mikejones@potterminton.com>
Subject: Re: Mizra - amended pleadings

Shaun,

K.Mizra LLC, Ex. 2001, p. 207
Google LLC v. K.Mizra LLC
IPR2025-01437