

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

MICROSOFT CORPORATION,
Petitioner,

v.

QOMPLX LLC,
Patent Owner.

Case IPR2026-00182
Patent 12,218,934

PATENT OWNER'S RESPONSE

TABLE OF CONTENTS

	Page
I. INTRODUCTION	1
II. PETITIONER’S GROUNDS FAIL (ALL CLAIMS, ALL GROUNDS).....	2
A. Petitioner Fails To Demonstrate That Its Grounds Disclose Or Render Obvious A “Time-Series Database” (All Claims, All Grounds).....	2
1. A “Time-Series Database” Is A Specialized Database Optimized Or Especially Suited For Time-Series Data.....	5
2. Petitioner Does Not Support Its Implicit Construction Of A “Time-Series Database.”	13
3. Petitioner Has Failed To Establish That Kirti’s Analytics And Threat Intelligence Repository Is A “Time-Series Database.”	18
B. Petitioner Fails To Demonstrate That Its Grounds Disclose Or Render Obvious “Determining Whether The Additional Verification Is Required To Grant Access” After “Receiv[ing] A Request To Authenticate A Client” (All Claims, All Grounds).	26
1. The Claims Require That The “Determining Whether An Additional Verification” Step Occurs After “Receiv[ing]” And As Part Of Evaluating “A Request To Authenticate A Client.”	27
2. Kirti’s Historical Threat Analysis Is Performed Independently Of The Current Authentication Request.	35
3. Kirti Does Not Determine That Additional Verification Is Required To Grant Access To The Current Authentication Request.....	42

C.	Petitioner Fails To Demonstrate Selection Of An Additional Verification Method “From A Plurality Of Verification Methods” (All Claims, All Grounds).....	47
1.	Kirti Selects Among Remedial Actions, Not Among A Plurality Of Verification Methods.	48
2.	Coffin Does Not Cure Kirti’s Failure To Select From A Plurality Of Verification Methods.	50
D.	Petitioner Fails To Demonstrate That Kirti “Determine[s] That A Probable Cyberattack Is Detected” “Based On The Additional Verification Being Required To Grant Access” (Claims 5, 15, 21, 29; All Grounds).....	54
III.	CONCLUSION	60

TABLE OF AUTHORITIES

Page(s)

COURT DECISIONS

<i>Amgen Inc. v. Sandoz Inc.</i> , 923 F.3d 1023 (Fed. Cir. 2019)	31
<i>Intelligent Bio-Sys., Inc. v. Illumina Cambridge, Ltd.</i> , 821 F.3d 1359 (Fed. Cir. 2016)	17, 43
<i>KSR Int’l Co. v. Teleflex Inc.</i> , 550 U.S. 398 (2007).....	53
<i>In re Magnum Oil Tools Int’l, Ltd.</i> , 829 F.3d 1364 (Fed. Cir. 2016)	53
<i>Mformation Techs., Inc. v. Rsch. in Motion Ltd.</i> , 764 F.3d 1392 (Fed. Cir. 2014)	30
<i>PAR Pharm., Inc. v. TWI Pharms., Inc.</i> , 773 F.3d 1186 (Fed. Cir. 2014)	19
<i>Persion Pharms. LLC v. Alvogen Malta Operations Ltd.</i> , 945 F.3d 1184 (Fed. Cir. 2019)	19, 21, 25
<i>Sound View Innovations, LLC v. Hulu, LLC</i> , 166 F.4th 958 (Fed. Cir. 2026)	31
<i>Southwire Co. v. Cerro Wire LLC</i> , 870 F.3d 1306 (Fed. Cir. 2017)	19, 21
<i>Wi-LAN, Inc. v. Apple Inc.</i> , 811 F.3d 455 (Fed. Cir. 2016)	31

AGENCY DECISIONS

<i>Xerox Corp. v. Bytemark, Inc.</i> , IPR2022-00624, Paper 9 (Aug. 24, 2022) (precedential).....	15
--	----

EXHIBIT LIST	
2001	Declaration of Nathan Lowenstein in Support of Notice of Intent
2002	Declaration of Colette Woo in Support of Notice of Intent
2003	Amended Scheduling Order, <i>Qomplx LLC v. Microsoft Corp.</i> , No. 1:25-cv-01383 (W.D. Tex.) (Jan. 5, 2026) [Amended Scheduling Order]
2004	Defendant Microsoft Corporation’s Preliminary Invalidity Contentions, <i>Qomplx LLC v. Microsoft Corp.</i> , No. 1:25-cv-01383, (W.D. Tex.) (served Jan. 26, 2026) [Invalidity Contentions]
2005	Ex. A-10 to Defendant Microsoft Corporation’s Preliminary Invalidity Contentions, <i>Qomplx LLC v. Microsoft Corp.</i> , No. 1:25-cv-01383 (W.D. Tex.) (served Jan. 26, 2026) [Invalidity Contentions – Ex. A-10]
2006	Ex. A-11 to Defendant Microsoft Corporation’s Preliminary Invalidity Contentions, <i>Qomplx LLC v. Microsoft Corp.</i> , No. 1:25-cv-01383, (W.D. Tex.) (served Jan. 26, 2026) [Invalidity Contentions – Ex. A-11]
2007	Katie Roof, <i>Risk Analytics Firm Qomplx to Go Public Via Casper CEO SPAC</i> , BLOOMBERG (Mar. 1, 2021, 11:16 PM UTC), https://www.bloomberg.com/news/articles/2021-03-01/risk-analytics-firm-qomplx-to-go-public-through-casper-ceo-spac (archived at http://archive.ph/3bpbk2) [Bloomberg]
2008	<i>On Point: EP. 3 Taking Qomplx Public via SPAC with Jason Crabtree '08</i> , https://www.oldgradclub.com/on-point/blog-post-title-two-ffmmh [On Point Podcast]
2009	<i>West Point First Captains</i> , WEST POINT ASSOCIATION OF GRADUATES, https://www.westpointaog.org/about/history/west-point-first-captains/ [West Point]
2010	Compilation of Awards Given to Jason Crabtree [Crabtree Awards]

2011	<i>Jason Crabtree</i> , NEW AMERICA, https://www.newamerica.org/our-people/jason-crabtree/ [New America]
2012	<i>Meet the Team</i> , THE DAKOTA FOUNDATION, https://www.dakotafoundation.org/team [Dakota Foundation]
2013	Joe Panettieri, <i>SPAC Cybersecurity Merger: Tailwind, QOMPLX Tackle Microsoft Active Directory Security</i> , MSSPALERT (Mar. 2, 2021), https://www.msspalert.com/news/spac-cybersecurity-merger-tailwind-qomplx-tackle-microsoft-active-directory-security [MSSPALert]
2014	<i>Architect's Corner: Qomplx leverages automation to run stateful services in containers for the financial and cybersecurity industries</i> , PORTWORX (Sept. 20, 2018), https://portworx.com/blog/devops-realized-qomplx-leverages-automation-run-stateful-services-containers-massive-scale/ [Portworx]
2015	QOMPLX, Inc., <i>QOMPLX CEO Jason Crabtree Wins Ernst & Young Entrepreneur Of The Year® 2020 Mid-Atlantic Award</i> (Aug. 10, 2021), https://finance.yahoo.com/news/qomplx-ceo-jason-crabtree-wins-135600025.html [Yahoo! Finance]
2016	Sydney Lake, <i>Tyson's analytics firm Qomplx plans to go public</i> , VIRGINIA BUSINESS (Mar. 2, 2021), https://virginiabusiness.com/tysons-analytics-firm-qomplx-plans-to-go-public/ [Virginia Business]
2017	Tailwind Acquisition Corp., <i>QOMPLX, Inc., Frequently Asked Questions about Tailwind Transaction</i> , https://www.sec.gov/Archives/edgar/data/1814215/000110465921033365/tm219025d2_425.htm [SEC FAQ]
2018	<i>QOMPLX and Tailwind Acquisition Corp. Mutually Agree To End Business Combination Due to Market Conditions</i> , BUSINESSWIRE (Aug. 17, 2021, 8:21 AM EDT), https://www.businesswire.com/news/home/20210817005565/en/QOMPLX-and-Tailwind-Acquisition-Corp.-Mutually-Agree-To-End-Business-Combination-Due-to-Market-Conditions [BusinessWire]

2019	<i>Entrepreneurship Essentials for the Military Community</i> , USPTO, https://www.uspto.gov/about-us/events/entrepreneurship-essentials-military-community [Entrepreneurship Essentials for the Military Community]
2020	<i>2024 Veterans Innovation and Entrepreneurship Program</i> , USPTO, https://www.uspto.gov/about-us/events/2024-veterans-innovation-and-entrepreneurship-program [Veterans Innovation and Entrepreneurship Program]
2021	<i>Ready, set, compete! How we're helping veterans and military family members innovate and start new businesses</i> , USPTO (May 25, 2023) https://www.uspto.gov/blog/ready-set-compete-how-we [Ready, Set, Compete]
2022	Time to Trial Statistics for J. Alan Albright (generated via DocketNavigator) [Albright Time To Trial]
2023	Declaration of Sam Malek, Ph.D.
2024	<i>Time series database explained</i> , INFLUXDATA, https://www.influxdata.com/time-series-database/ (accessed Mar. 5, 2026) [InfluxData]
2025	Marie Fayard, <i>Time Series Database (TSDB): A Guide With Examples</i> , DATACAMP (updated Feb. 21, 2025), https://www.datacamp.com/blog/time-series-database [Datacamp]
2026	Piotr Grzesik & Dariusz Mrozek, <i>Comparative Analysis of Time Series Databases in the Context of Edge Computing for Low Power Sensor Networks</i> , COMPUTATIONAL SCIENCE – ICCS 2020, 371. [Grzesik]
2027	Niels de Waal, <i>Literature Study: Timeseries Databases</i> (Dec. 16, 2022) [Literature Study: Timeseries Databases]

2028	Keith D. Foote, <i>A Guide to Time Series Databases</i> , DATAVERSITY (Sept. 15, 2022), https://www.dataversity.net/articles/a-guide-to-time-series-databases/ [A Guide to Time Series Databases]
2029	<i>The ultimate guide to time series databases</i> , KX, https://kx.com/time-series-database/ (accessed Apr. 7, 2026) [The ultimate guide to time series databases]
2030	Alex Vondrak, <i>How Time Series Databases Work-and Where They Don't</i> , HONEYCOMB (Jan. 6, 2026), https://www.honeycomb.io/blog/time-series-database [Honeycomb]
2031	<i>What Is a Time Series Database? How It Works + Use Cases</i> , Timeplus (Feb. 2, 2024), https://www.timeplus.com/post/time-series-database [What is a Time Series Database?]
2032	U.S. Pat. No. 10,204,147 ['147]
2033	Alexandre Gaillard, <i>Top 5 User Identity Verification Methods for 2025</i> , INVESTGLASS (updated July 3, 2025), https://www.investglass.com/top-5-user-identity-verification-methods-for-2025/ [InvestGlass]
2034	DOUGLAS DOWNING ET AL., <i>DICTIONARY OF COMPUTER AND INTERNET TERMS</i> (BARRON'S BUSINESS GUIDES), 530 (12 th ed. 2017) [Barron's Dictionary of Computer and Internet Terms]
2035	Declaration of Colette Woo [SERVED, NOT FILED]
2036	<i>Why Time Series Matters for Metrics, Real-Time and Sensor Data</i> , INFLUXDATA, https://get.influxdata.com/rs/972-GDU-533/images/why%20time%20series.pdf [SERVED, NOT FILED]
2037	Second Declaration of Sam Malek, Ph.D. [Malek-2 nd -Decl.]
2038	Deposition of John R. Black, Ph.D. (August 7, 2026) [Black-Tr.]

2039	Søren K. Jensen et al., <i>Time Series Management Systems: A Survey</i> , 29 IEEE Transactions on Knowledge & Data Eng'g 2581 (2017) [Jensen]
2040	Goetz Graefe, <i>Storage Manager</i> , in <i>Encyclopedia of Database Systems</i> 3736 (Ling Liu & M. Tamer Özsu eds., 2d ed. 2018) [Graefe]
2041	Sofwan A. Lawal, <i>Database Engines: Overview: Considerations and Importance in Database Engineering</i> (July 25, 2023), https://blog.sofwancoder.com/database-engines-overview (accessed Aug. 31, 2026) [Database-Engines-Overview]
2042	<i>Database Design in DBMS</i> , GeeksforGeeks (updated May 1, 2026) https://www.geeksforgeeks.org/dbms/database-design-in-dbms/ (accessed Aug. 31, 2026) [GeeksforGeeks]
2043	Sean Rhea et al., <i>LittleTable: A Time-Series Database and Its Uses</i> , in <i>Proc. of the 2017 ACM Int'l Conf. on Mgmt. of Data</i> 125 (2017) [LittleTable]

I. INTRODUCTION

Petitioner fails to prove that any of the challenged claims are unpatentable for at least the reasons below.

First, Petitioner fails to demonstrate that its grounds disclose or render obvious the claimed “time-series database.” *See* Section II.A. The intrinsic record, technical literature, and Dr. Black’s testimony establish that a time-series database is a specialized database optimized or especially suited for time-series data. Petitioner instead treats Kirti’s repository as a time-series database based on characteristics of the data stored within it. And Kirti never identifies repository 211 as a time-series database or discloses the design or organization that would make it one. *See* Sections II.A.1-.3.

Second, Petitioner fails to demonstrate that Kirti performs the claimed determination whether additional verification is required to grant access after receiving, and as part of evaluating, the current authentication request. *See* Section II.B. Kirti’s historical threat analysis occurs independently of any particular authentication request. And even assuming that analysis can occur in real time, Kirti does not determine that additional authentication is required to grant access to the request being evaluated. *See* Sections II.B.1-.3.

Third, Petitioner fails to demonstrate selection of an additional verification method “from a plurality of verification methods.” *See* Section II.C. Kirti selects

among remedial actions, only one of which Petitioner characterizes as additional verification. Coffin, meanwhile, selects a delivery channel for a passcode, not among the plurality of authentication techniques on which Petitioner relies. Neither establishes the claimed selection. *See* Sections II.C.1-.2.

Finally, Petitioner fails to establish the additional limitations of claims 5, 15, 21, and 29. Those claims require determining that a probable cyberattack is detected “based on” additional verification being required to grant access. Kirti teaches the opposite relationship: it detects a threat and then selects or recommends remediation, potentially including additional authentication. *See* Section II.D.

II. PETITIONER’S GROUNDS FAIL (ALL CLAIMS, ALL GROUNDS).

A. Petitioner Fails To Demonstrate That Its Grounds Disclose Or Render Obvious A “Time-Series Database” (All Claims, All Grounds).

Each of the claims requires the use of a “multidimensional *time-series database*.” Petitioner maps that limitation to Kirti’s “analytics and threat intelligence repository database.” Pet., 31. But Kirti never identifies that repository as a time-series database, and Petitioner never demonstrates that it is one. Instead, Petitioner identifies characteristics of the *data* stored in the repository—activity data having multiple attributes and associated with multiple points in time—and concludes from those characteristics that the *database* containing the data “is” a time-series database. *Id.*, 31-33. The developed record

establishes that this inference is exactly backwards: a “time-series database” is a specialized database especially suited or optimized for time-series data, not merely any database that happens to contain time-stamped information. *See* Section II.A.1.

Critically, Petitioner’s own expert confirms that distinction. Dr. Black testified that a time-series database may use a general-purpose database engine only if the database is “design[ed] and arrang[ed] ... appropriately,” and that a general-purpose database qualifies as a time-series database only “if it’s organized in the correct way.” Ex. 2038 [Black-Tr.] 27:15-28:6, 30:13-31:19. He likewise confirmed that merely storing timestamped information does not establish that the database itself is a time-series database; that depends upon how the database is implemented and organized. *Id.*, 85:15-86:10, 91:1-21.

Petitioner tries to mask this problem by avoiding explicitly setting forth and supporting the construction of “time-series database” that its analysis actually applies. Petitioner reasons that “at least because” repository 211 stores the same kinds of activity data at multiple points in time, the repository itself “is” a time-series database. Pet., 31-32. But that conclusion assumes that merely storing time-series data is enough to convert the database containing it into a time-series database. Petitioner cites no intrinsic or extrinsic evidence supporting that assumption, and Dr. Black’s declaration merely repeats it without technical

reasoning. His deposition testimony confirms the opposite: whether a database containing timestamped information is a time-series database depends on how the database itself is designed and organized. Petitioner therefore has not supported the understanding of “time-series database” necessary for its mapping to work. Moreover, Petitioner’s use of the phrase “*at least* because” suggests that Petitioner expressly refused to provide a binding construction. *See* Section II.A.2.

Independently, Petitioner fails to prove that Kirti’s repository actually satisfies the proper understanding of a time-series database. Because Kirti never identifies repository 211 as one, and Petitioner does not propose modifying or specially configuring it to create one, Petitioner necessarily asks the Board to infer that repository 211 possesses that undisclosed characteristic. But inherency requires necessity, not possibility. And Kirti affirmatively defeats any such showing: it states that repository 211 may be “*any database* or data repository with query capability,” while its exemplary Cassandra engine is, as Dr. Black agreed, a general-purpose engine not directed to time-series data. Ex. 1004 [Kirti] 10:61-11:2; Ex. 2038 [Black-Tr.] 75:7-19, 82:17-83:12. Kirti nowhere discloses how repository 211 is designed or organized in a manner that makes it a time-series database. Nor do Kirti’s time-based reports or analytics fill that gap: they concern what Kirti does *with data obtained from* repository 211, not how the repository itself is designed or organized. *See* Section II.A.3.

Because Petitioner fails to prove that Kirti discloses the claimed “time-series database,” much less that such a database is necessarily present, the claims should be confirmed.

1. A “Time-Series Database” Is A Specialized Database
Optimized Or Especially Suited For Time-Series Data.

The intrinsic record, contemporaneous technical literature, and even Dr. Black’s testimony all establish that a “time-series database” is a specialized database optimized or especially suited for time-series data, not merely a generic database that happens to contain timestamped information.¹

The claim language itself reflects that importance of the “time-series *database*,” not merely the data stored within it. As Dr. Malek explains:

The claims recite “stor[ing] ... information” in and “retrieving ... historical information” from a multidimensional “time-series *database*.” Ex. 1001 [’934] cls. 1, 11, 18, 26. Notably the claims do not recite merely storing “time-series data” in any kind of database of the user’s choosing. Rather, the claims specify the use of a “time-series

¹ This is precisely the same claim construction Patent Owner proposed in the POPR. *See, e.g.*, Paper 12, 9-13 (“both the technical literature and the intrinsic record describe a ‘time-series database’ as a specialized database optimized or especially suited for time-series data, not a generic database.”).

database.” “Time-series,” thus, describes the database, not simply the data that a user may store in the database.

Ex. 2037 [Malek-2nd-Decl.] ¶ 36.

The specification confirms this understanding. As Dr. Malek explains:

Moreover, the '934 patent incorporates by reference one of its parent applications, which resulted in the '147 patent, titled “System for capture, analysis and storage of time series data from sensors with heterogeneous report interval profiles.” The '147 patent teaches that “the sensor data is passed without transformation to the data management engine 120, where it is aggregated and organized for storage *in a specific type of data store 125 designed to handle the multidimensional time series data* resultant from sensor data.” Ex. 2032 ['147] 6:34-39; *see id.*, 7:20-24. It further explains that, due to the volume and continuous nature of such time series data, it cannot be stored arbitrarily, but instead must be organized using timestamps and defined sampling intervals—*e.g.*, storing data “every 10 seconds, using the timestamp as the key.” *Id.*, 6:51-56.

The “specific type of data store,” thus, is “designed to handle the multidimensional time series data” and utilizes, for example, timestamp keys, to efficiently process time-series workloads. Accordingly, the intrinsic record describes a specialized database optimized or especially suited for time-series data, rather than a general-purpose database that merely stores data over time.

Ex. 2037 [Malek-2nd-Decl.] ¶¶ 37-38.

The extrinsic evidence confirms the same understanding. The Petition employs a priority date of October 19, 2017. The contemporaneous literature consistently refers to time-series databases as specialized databases that are optimized or especially suited for time-series data, rather than databases that contain time-associated data. As Dr. Malek explains:

Consistent with the intrinsic evidence, the extrinsic evidence also confirms that a time-series database is a specialized database optimized or especially suited for time-series data. For example, a 2017 survey of time-series databases explains that although general Database Management Systems (“DBMSs”), including relational DBMSs, “have been successfully deployed in many situations,” they were “unsuitable to handle the velocity and volume of the time series produced by the large scale sensor networks deployed today.” Ex. 2039 [Jensen] 2. To address the increasing need to efficiently store and analyze time-series data, “specialized Time Series Management Systems (TSMSs) have been developed to overcome the limitations of general purpose Database Management Systems (DBMSs) for time[] series management.” *Id.* Jensen notes that “Time Series Database” is another common name for these systems. *Id.*, 2 n.1. The authors define a TSMS as a system “developed or extended for storing and querying data in the form of time series.” *Id.*, 2. The survey further noted systems using different architectures—including systems with internal data stores, systems built upon external DBMSs, and systems extending existing relational databases—and evaluated, among other things, the

workload for which each system was “designed and optimized.” *Id.*, 3-5.

Jensen thus reflects the same distinction described in the intrinsic record. A database does not become a time-series database merely because a user stores records collected at different times in it. Rather, a time-series database is specialized, optimized, or otherwise designed for management of time-series data. Jensen’s contrast is particularly instructive: a general-purpose DBMS may contain time-series data, yet Jensen nevertheless distinguishes such systems from the specialized time-series systems developed to efficiently store and query that data.

Ex. 2037 [Malek-2nd-Decl.] ¶¶ 39-40.

Dr. Black’s testimony independently confirms this distinction. A database may be specialized or optimized through either its database engine or the design and organization of its underlying store. As Dr. Malek explains:

A database system generally includes software that manages the database, commonly referred to as a database-management system or database engine, as well as the underlying database store in which the data and associated structures are maintained. *See* Ex. 2040 [Graefe] 1 (explaining that the “storage manager is a software layer within a database management system” that exposes records in “storage structures” and manages access to data in permanent storage); Ex. 2038 [Black-Tr.] 19:22-21:18. Either—or both—of these components may allow a database system to be optimized or especially suited for time-series data.

For example, a database engine may be optimized for a particular type of data or workload by incorporating functionality tailored to that data, such as specialized storage structures, indexing, query-processing techniques, compression, or other features intended to improve the storage, retrieval, or analysis of that data. *See* Ex. 2041 [Database-Engines-Overview] 1-8; Ex. 2038 [Black-Tr.] 31:5-19 (explaining that adding “specific features to the engine itself” can make it “specialized” so that particular databases “work better,” are “faster,” or have other “desirable features”); 77:1-78:3 (specialized engine may be “particularly tuned to provide features” for a particular type of data); 83:14-84:3 (describing optimization through sorting, indexing, and “efficient algorithms”). Time-series database engines, therefore, may include features specifically directed to storing and querying time-series data more efficiently than a general-purpose database engine. *See* Ex. 2038 [Black-Tr.] 54:15-55:22 (specialized time-series engines may implement roll-ups to summarize historical data and conserve storage); 59:12-21 (roll-ups are one of a “list of different features” “specific to time series”); 82:17-83:12 (general-purpose engines are not “particularly directed toward” time-series data and lack features for a particular data type).

Ex. 2037 [Malek-2nd-Decl.] ¶¶ 41-42.

Alternatively or in addition to a specialized database engine, the database store itself may instead be designed and arranged for time-series data. As Dr. Malek explains:

Another way a time-series database can be optimized for time-series data is through the design of the database store. Database stores can be designed to “organiz[e] and structur[e] data in a way that ensures efficiency, accuracy, and usability.” Ex. 2042 [GeeksforGeeks] 1. Dr. Black similarly explained that designing a database includes determining “how to arrange all the data” in the data store, including its metadata, indexes, relationships, and schema. Ex. 2038 [Black-Tr.] 19:22-21:18. He further testified that this design process may involve determining “how we’re going to lay the data out on the disk such that the engine can manage it in the way that we want.” *Id.*, 29:1-14. A contemporaneous example of such a time-series database is the “LittleTable” system, which is a “relational database optimized for time-series data.” Ex. 2043 [LittleTable] 1. The LittleTable system was specifically designed around characteristics of time-series data, partitioning rows by timestamp, clustering data along a time dimension, and selecting its storage layout so that time-based information could be retrieved efficiently. *Id.*, 1-3.

Indeed, Dr. Black testified that a time-series database can be built using general-purpose database engines such as Cassandra, Redis, or even a relational database engine, but only by “designing and arranging the database appropriately.” Ex. 2038 [Black-Tr.] 27:15-28:6. He similarly testified that his understanding of a time-series database before the 2015-2020 time-range was that “the data store was arranged in a certain way using a generic general-purpose database engine,” and that a general-purpose database could be used as a time-series database only “if it’s organized in the correct way.” *Id.*, 30:13-31:19. By

Dr. Black’s own reasoning, merely storing time-series data in Cassandra, Redis, or a relational database engine, without taking specific steps to design and arrange the database appropriately, would not turn that database into a “time-series database.”

Ex. 2037 [Malek-2nd-Decl.] ¶¶ 43-44.

The remaining technical literature further supports this understanding of time-series databases. As Dr. Malek explains, sources since 2017 have continued to describe time-series databases as specialized databases optimized or especially suited for time-series data:

The term “time-series database” has also been used in numerous sources and primers to refer to specialized databases optimized for time-series data since at least 2017. For example, InfluxData explains that a “time series database (TSDB) is a database optimized for time-stamped or time series data” and is “built specifically for handling metrics and events ... that are time-stamped.” Ex. 2024 [InfluxData] 4. Similarly, DataCamp explains that time-series databases are “specialized databases designed to manage data that is organized and indexed by time,” and distinguishes them from “traditional databases ... optimized for general-purpose data storage.” Ex. 2025 [Datacamp] 1.

Other technical and academic sources confirm the same understanding. *See, e.g.*, Ex. 2026 [Grzesik] 3 (“Time series database (TSDB) is a database type designed and optimized to handle timestamped or time-series data”); Ex. 2027 [Literature Study:

Timeseries Databases] 1 (describing time-series databases as “specialized” systems “optimize[d]” for time-indexed data); Ex. 2028 [A Guide to Time Series Databases] 1 (time-series databases “have been optimized for processing time series data”); Ex. 2030 [Honeycomb] 3 (time series database is “a specialized database that efficiently stores and retrieves time-stamped data”); Ex. 2031 [What is a Time Series Database?] 1-2 (explaining that time-series databases are “specialized” systems “specifically designed” to handle time-stamped data).

These sources consistently describe time-series databases as databases specialized for time-indexed data, not general-purpose databases that happen to store time-stamped data. While, as Dr. Black testified, time-series databases become more powerful and sophisticated around the 2015-2020 time-range (Ex. 2038 [Black-Tr.] 30:5-31:1), the fundamental requirement that they be optimized or especially suited for storing and processing time-series data has not changed.

Ex. 2037 [Malek-2nd-Decl.] ¶¶ 45-47.

Thus, the intrinsic record, contemporaneous literature, and Dr. Black’s own testimony all establish the relevant distinction. A database need not use any particular engine, schema, or implementation to qualify as a time-series database. But the database must—whether through its engine, the design and organization of its store, or both—be optimized or especially suited for time-series data. The mere

presence of timestamped data in an otherwise ordinary database does not make that database a time-series database.

2. Petitioner Does Not Support Its Implicit Construction Of A “Time-Series Database.”

As established above, a “time-series database” is a specialized database optimized or especially suited for time-series data, not merely any database that happens to contain timestamped information. *See* Section II.A.1. Petitioner’s mapping, however, necessarily applies a different, implicit, and unsupported understanding of a time series database. Although Petitioner purports to apply the “plain and ordinary meaning of each claim term” (Pet., 16), its analysis necessarily assumes that a database is a “time-series database” merely because it contains data associated with multiple points in time.

Petitioner’s own explanation makes that implicit construction clear. Petitioner argues that “Kirti’s analytics and threat intelligence repository database ... *is a time-series database at least because it includes the same activity data* (i.e., values for the same parameters/attributes) *for multiple time points.*”²

² Petitioner separately contends that Kirti’s data are “multidimensional” because each time point contains multiple attributes. Pet., 31-32. Patent Owner need not dispute that proposition here. The dispute concerns Petitioner’s separate

Pet., 31-32. Petitioner, thus, derives its conclusion entirely from characteristics of the *data* stored within the *database*. Because repository 211 contains activity data associated with multiple points in time, Petitioner concludes that repository 211 “is a time-series database.” *Id.*, 31-32. But even assuming that “the same activity data (i.e., values for the same parameters/attributes) for multiple time points” qualifies as “time-series data,” Petitioner identifies nothing about the database itself that makes it a time-series database. Its analysis therefore assumes that storing time-series data is sufficient to make the database containing that data a time-series database—the very assumption that Section II.A.1 and Dr. Black’s deposition testimony show is incorrect.

That assumption disregards the claim language itself. The limitation concerns a “time-series *database*” not merely “time-series data,” but Petitioner’s implicit construction requires nothing of the database itself. Its construction therefore eliminates the very distinction established in Section II.A.1 between a time-series database and an ordinary database that happens to contain time-stamped information. Under Petitioner’s reasoning, for example, an ordinary database of an individual’s checking-account transactions could qualify as a time-

conclusion that the repository containing those data is a “time-series database.”

series database merely because it stores transactions occurring at different points in time.

But Petitioner provides no evidence that a POSITA would understand “time-series database” in that expansive manner. The Petition cites no intrinsic or extrinsic evidence supporting its implicit construction and does not demonstrate that such an understanding was known in the art. *See* Pet., 31.

Its only purported support is Dr. Black’s declaration, but Dr. Black merely repeats the Petition’s conclusion that Kirti’s repository “is” a time-series database because it contains activity data at multiple points in time. Ex. 1003 [Black-Decl.]

¶ 87. He, like the Petition, provides no evidence or technical reasoning explaining why that characteristic of the stored data establishes that the database itself is a time-series database. In *Xerox Corp. v. Bytemark, Inc.*, the Board gave little weight to expert testimony that “merely repeats, *verbatim*, the conclusory assertion for which it is offered to support” without “cit[ing] any additional supporting evidence or provid[ing] any technical reasoning to support his statement.”

IPR2022-00624, Paper 9, 15 (Aug. 24, 2022) (precedential) (emphasis in original, citations omitted). Dr. Black’s declaration does the same thing here. It supplies no evidence or reasoning supporting the premise that time-series data makes the database containing it a time-series database, and therefore cannot carry Petitioner’s burden to establish the limitation.

And, contrary to Petitioner’s position, characteristics of the stored data alone do not establish that the database is a time-series database. As Dr. Malek explains:

The presence of time-stamped data in a store does not itself make that store into a time-series database. Indeed, Dr. Black agrees with me. Critically, Dr. Black testified that although a time-series database can be built using a “nonspecialized engine,” doing so requires “*designing and arranging the database appropriately.*” Ex. 2038 [Black-Tr.] 27:15-28:6. Consistent with that distinction, when asked whether an Outlook inbox containing regularly received, timestamped sensor data would be a time-series database, Dr. Black testified that “it depends” on implementation details, including whether it “prioritize[s] timestamp as a primary key and is organized around time.” *Id.*, 85:15-86:10. Similarly, when asked whether browser history stored in a SQL database constitutes a time-series database, Dr. Black again testified that “[i]t could be, depending on certain details,” including whether “the database is organized around the timestamp as a first-class object and index.” *Id.*, 91:1-21.

Thus, in my opinion, Dr. Black’s deposition testimony confirms that merely storing information associated with multiple points in time does not make a database a time-series database. Rather, Dr. Black testified that whether a database is a time-series database depends upon the design and organization of the database itself.

Ex. 2037 [Malek-2nd-Decl.] ¶¶ 52-53. Dr. Black’s testimony thus leaves no evidentiary support for the inference made in paragraph 87 of his declaration.

Moreover, Dr. Black’s own cross-examination testimony undermines the premise of his declaration. Dr. Black testified that a general-purpose database may serve as a time-series database only “if it’s organized in the correct way.” Ex. 2038 [Black-Tr.] 30:13-31:19. Similarly, a time-series database may use a nonspecialized engine, but only by “designing and arranging the database appropriately.” *Id.*, 27:15-28:6; *see* Section II.A.1. His testimony thus contradicts the premise underlying his declaration that the nature of the stored data alone establishes that Kirti’s repository is a time-series database.

Nor can Petitioner cure that failure by replacing the Petition’s theory—that Kirti’s repository already “is” a “time-series database” because of the data it contains—with a new theory that a POSITA could have specially configured Kirti’s repository to create one. The Federal Circuit has made clear that “the expedited nature of IPRs bring with it an obligation for petitioners to ***make their case in their petition to institute.***” *Intelligent Bio-Sys., Inc. v. Illumina Cambridge, Ltd.*, 821 F.3d 1359, 1369 (Fed. Cir. 2016). The Petition neither identifies such a configuration nor proposes modifying Kirti’s repository in that manner.

Petitioner therefore fails to substantiate the implicit construction underlying its mapping of the time-series-database limitation. The Petition simply assumes that characteristics of the stored data establish the character of the database, and

Dr. Black's declaration provides no reasoned support for that assumption. His deposition testimony instead confirms that the assumption is wrong. And, as discussed next, Petitioner does not demonstrate that Kirti's repository satisfies the proper understanding of a time-series database. *See* Section II.A.3.

3. Petitioner Has Failed To Establish That Kirti's Analytics And Threat Intelligence Repository Is A "Time-Series Database."

As established in Section II.A.1, a "time-series database" is a specialized database optimized or especially suited for time-series data, not merely any database that happens to contain timestamped information. In addition to failing to support its implicit and incorrect construction (*see* Section II.A.2), Petitioner has also failed to prove that repository 211 actually satisfies the proper construction of a "time-series database." Kirti never identifies repository 211 as a time-series database. Yet Petitioner does not propose modifying or specially configuring repository 211 to create one; it asserts that Kirti's existing repository already "*is* a time-series database." Pet., 31-32. Because that asserted characteristic is nowhere disclosed by Kirti, Petitioner necessarily seeks to infer it from other disclosed features—"at least" that repository 211 "includes the same activity data (i.e., values for the same parameters/attributes) for multiple time points" and that repository 211 "stores weeks' worth of activity data that 'can be utilized to generate reports'" Pet., 31-33.

Petitioner’s theory is thus one of inherency: although Kirti never says repository 211 is a time-series database, Petitioner contends that the disclosed repository necessarily possesses that unstated characteristic. That theory faces a demanding standard. Inherency in an obviousness analysis is a “high standard” that must be “carefully circumscribed.” *Persion Pharms. LLC v. Alvogen Malta Operations Ltd.*, 945 F.3d 1184, 1191 (Fed. Cir. 2019) (quoting *PAR Pharm., Inc. v. TWI Pharms., Inc.*, 773 F.3d 1186, 1195 (Fed. Cir. 2014)). It “may not be established by probabilities or possibilities”; the missing limitation must be “necessarily present” or “the natural result” of what the prior art teaches. *Id.* The Federal Circuit has likewise rejected an obviousness determination based on inherency where the record did not establish that the missing limitation “necessarily would” result from the prior-art disclosure. *Southwire Co. v. Cerro Wire LLC*, 870 F.3d 1306, 1311-12 (Fed. Cir. 2017).

Petitioner does not come close to making that showing. The developed record instead demonstrates that whether a general-purpose database is a time-series database depends upon implementation choices that Kirti does not specify. Dr. Black testified that selecting a database engine does not determine the nature of the resulting database: the designer must still determine how the data are arranged, what metadata and indexes are used, what relationships are built, and how the schema is structured. Ex. 2038 [Black-Tr.] 20:1-21:18. The same generic engine

may be used in “very different ways” depending upon its purpose. *Id.*, 22:1-23:2.

And specifically as to time-series databases, Dr. Black testified that a nonspecialized engine may be used to build one only by “designing and arranging the database appropriately,” and that a general-purpose database may serve as a time-series database only “if it’s organized in the correct way.” *Id.*, 27:15-28:6, 30:13-31:19.

Petitioner identifies no such design or organization for repository 211. Patent Owner does not contend that every time-series database must employ some particular schema, index, key structure, partitioning scheme, or other implementation feature. *See* Section II.A.1. The defect is more basic: Petitioner bears the burden of proving that the particular database it mapped is a time-series database, yet identifies nothing about the design or organization of repository 211 establishing that it is specialized or optimized for storing and processing time-series data. And as Dr. Malek explains, the significance of Kirti’s failure to specify the implementation of repository 211 is particularly important here:

The design and configuration choices used to make a general-purpose database optimized or especially suited for time-series data are not implementation-independent. They depend, among other things, on the database technology and storage architecture selected. For example, a relational database system and a system implemented using Cassandra provide different data models, storage mechanisms, and

mechanisms for organizing and retrieving data, and therefore involve different design and configuration choices when used for a time-series workload. Thus, one cannot infer merely from the data being stored that every database implementation capable of storing that data necessarily has been designed or organized as a time-series database.

Ex. 2037 [Malek-2nd-Decl.] ¶ 57.

The very Kirti passage Petitioner cites affirmatively defeats any inference of necessity. *See* Pet., 31-32 (citing, *inter alia*, Ex. 1004 [Kirti] 10:60-11:4). Kirti expressly leaves implementation choices open and states that repository 211 “may be **any database** or data repository with query capability” and that “any data warehouse infrastructure may be utilized as appropriate for the application.” Ex. 1004 [Kirti] 10:61-11:2. A disclosure expressly permitting different database implementations cannot establish that the undisclosed characteristics of a time-series database are **necessarily** present in repository 211. At most, it leaves open the possibility that repository 211 could be implemented as one. Inherency requires necessity, not possibility. *Persion*, 945 F.3d at 1191; *Southwire*, 870 F.3d at 1311-12.

Kirti’s exemplary reference to Cassandra only proves possibility, not inherency. Kirti states that, in several embodiments, repository 211 may use “a NoSQL based infrastructure such as Cassandra or other distributed data processing system.” Ex. 1004 [Kirti] 10:63-67. But Petitioner provides no analysis

explaining why Cassandra makes repository 211 a time-series database. And Dr. Black testified that Cassandra is a “general-purpose database engine,” Ex. 2038 [Black-Tr.] 75:7-19, and that a general-purpose engine is “not particularly directed toward ... time-series data” and “doesn’t have features for any particular type of data.” *Id.*, 82:17-83:12; Ex. 2037 [Malek-2nd-Decl.] ¶¶ 58-59. And as Dr. Black further explained, using a nonspecialized engine to build a time-series database requires “designing and arranging the database appropriately.” *Id.*, 27:15-28:6. Petitioner identifies no such design or arrangement for repository 211. Thus, Kirti’s exemplary Cassandra implementation does not establish that repository 211 necessarily is a time-series database; Dr. Black’s testimony establishes that additional, undisclosed implementation choices would determine whether it is one. Ex. 2037 [Malek-2nd-Decl.] ¶ 59.

Nor can Petitioner find the missing database characteristic in Kirti’s references to “reformatting,” “categorizing,” or “structuring.” Petitioner does not rely on those disclosures to establish a time-series database, and they would not do so in any event. Kirti describes reformatting and structuring the *data* received from different cloud applications into a “common format” for storage and retrieval. Ex. 1004 [Kirti] 11:3-15. That says nothing about whether repository 211 itself has the specialized or optimized database design the limitation requires.

Petitioner instead attempts to bolster its inference with Kirti’s time-based

reports. Petitioner asserts that repository 211 stores weeks of activity data and that Figure 8C displays events over four- and twelve-week periods, which supposedly “confirm[s]” the database’s asserted “time-series nature.” Pet., 33. But this evidence fares no better under either the proper construction or the inherency standard. The ability to generate a time-based report does not establish that the database from which the information was obtained is necessarily a time-series database. Indeed, Petitioner’s reasoning is circular: it begins by assuming the database’s “time-series nature” and then invokes reports generated from its data to “confirm[]” that assumed nature. *Id.*; Ex. 2037 [Malek-2nd-Decl.] ¶ 61.

Figure 8C illustrates the evidentiary gap. As Dr. Malek explains:

Kirti’s Figure 8C does not depict repository 211 or any aspect of its design or organization. Kirti describes Figure 8C simply as a user-interface chart displaying “a count of events by date” in categories such as failed logins, after-hours downloads, and activity at unusual times. Ex. 1004 [Kirti] 15:4-12. The figure shows aggregate event counts over selected time periods. It does not show the underlying database records, much less how repository 211 stores, indexes, partitions, keys, or otherwise organizes them. Nothing in Figure 8C establishes—necessarily or otherwise—that repository 211 is a time-series database.

Kirti’s own architecture further defeats Petitioner’s attempt to derive the nature of repository 211 from the analytics performed on its data. Kirti separately identifies repository 211, descriptive analytics

application 207, and threat detection and prediction analytics application 212, and expressly states that “[a]nalytics may be performed **using data stored** in the analytics and threat intelligence repository 211.” Ex. 1004 [Kirti] 5:38-54. Kirti likewise attributes analytics and report generation to MapReduce, Spark, Apache Hive, Hadoop, Mahout, R, scheduled jobs, and other analytics processes. *Id.*, 11:18-12:57. Thus, Petitioner points to functions performed **on data obtained from** repository 211 and asks the Board to infer from those functions an unstated characteristic of the repository itself. Kirti does not support that inference. The fact that separate analytics components can generate time-based reports from data stored in repository 211 says nothing about whether repository 211 itself is designed as a time-series database.

Ex. 2037 [Malek-2nd-Decl.] ¶¶ 62-63.

Dr. Black’s testimony confirms why it fails. When asked whether browser-history information stored in an SQL database could make that database a time-series database, Dr. Black testified that “[i]t could be, depending on certain details,” including whether “the database is organized around the timestamp as a first-class object and index.” Ex. 2038 [Black-Tr.] 90:25-91:21. Although he also identified reports over time as potentially relevant, his answer expressly depended on additional characteristics of the database itself. *Id.* Petitioner identifies no corresponding characteristic of repository 211. Dr. Black’s “it could be” testimony is, moreover, the antithesis of inherency: a characteristic that **could** be present,

depending upon unspecified implementation details, is not necessarily present. *See Persion*, 945 F.3d at 1191.

Kirti's separate reference to "time series" does not rescue Petitioner either. Kirti identifies "time series" alongside "[d]ecision tree" and "naive Bayes analysis" as examples of machine-learning techniques that may be used "to generate predictions." Ex. 1004 [Kirti] 12:58-64. That disclosure concerns an analytical technique applied to data, not the type of database in which the data are stored. Indeed, it reinforces Kirti's distinction between repository 211 and the separate analytics performed using information obtained from it.

Petitioner's Preliminary Reply merely relabels the same inference. It asserts that Kirti's reports and use of access-request timing to calculate login velocity "confirm[] it is structured for time-based security analytics." Paper 16, 2-3. But "structured for time-based security analytics" is not the claimed limitation. The claims require a "time-series *database*," and Petitioner maps that database specifically to repository 211. Pet., 31-32. The Preliminary Reply again identifies what Kirti's analytics do with time-associated data; it still identifies no characteristic of repository 211 establishing that repository 211 necessarily is a time-series database.

Petitioner therefore has failed to prove the limitation independently of the claim-construction failure addressed in Section II.A.2. Kirti does not expressly

identify repository 211 as a time-series database. Petitioner’s attempt to infer that undisclosed characteristic from the data stored in repository 211 and the analytics performed on those data cannot satisfy inherency: Kirti expressly leaves the repository implementation open, and Dr. Black confirms that whether a general-purpose database becomes a time-series database depends upon additional design and organization choices. Petitioner proves, at most, that repository 211 *could* be implemented as a time-series database. It does not prove that repository 211 *is* one, much less that a time-series database is necessarily present in Kirti. Petitioner therefore has failed to carry its burden on the claimed “time-series database” limitation.

B. Petitioner Fails To Demonstrate That Its Grounds Disclose Or Render Obvious “Determining Whether The Additional Verification Is Required To Grant Access” After “Receiv[ing] A Request To Authenticate A Client” (All Claims, All Grounds).

The claims require that when the computer system “receive[s] a request to authenticate a client,” the system then “determine[s] whether additional verification is required to grant access.” Ex. 1001 [’934] cl. 1; *see also id.*, cls. 11, 18 & 26 (similar). That determination “comprises” retrieving historical information from the MDTSDb. *See* Section II.B.1. Petitioner, however, fails to demonstrate that Kirti performs this required determination after “receiving” the current “request to authenticate a client.”

Petitioner's theory instead pieces together separate aspects of Kirti's distinct and asynchronous data-collection, threat-analysis, and remediation systems. Kirti's threat-analysis processes are not tied to the current authentication request. *See* Section II.B.2. Even assuming that Kirti analyzes a current authentication event in real time, it still does not determine that additional verification is required to grant access to that request. *See* Section II.B.3.

1. The Claims Require That The “Determining Whether An Additional Verification” Step Occurs After “Receiv[ing]” And As Part Of Evaluating “A Request To Authenticate A Client.”

The claims set out a clear authentication process. A client submits an identifier and password; the system checks if the credentials are valid, and decides whether the client must complete additional verification before access is granted. In making that decision, the system retrieves historical information about earlier access requests. Claim 1, in relevant part, provides:

[1.1] A computer system configured to execute software instructions stored on nontransitory machine-readable storage media, wherein the software instructions comprise instructions that:

[1.2] *receive a request to authenticate a client*, wherein the request comprises *an identifier and a password*,

[1.3] store, in a multidimensional time-series database, information about *the request*,

[1.4] determine whether *the password* corresponds to a user account identified by *the identifier*,

[1.5] *determine whether an additional verification is required to grant access*, wherein determining whether the additional verification is required to grant access comprises:

[1.6] *retrieving, from the multidimensional time-series database, historical information about previous access requests* associated with *the user account*, and

...

[1.8] based on the additional verification being required to grant access:

[1.9] select an additional verification method from a plurality of verification methods,

[1.10] cause the client to be prompted to complete the additional verification method, and

Ex. 1001 [’934] cl. 1. The other independent claims have substantially similar limitations. *Id.*, cls. 11, 18, 26.

As can be seen above, the claims do not recite a collection of independent capabilities. Rather, they recite a series of operations performed after receiving and as part of evaluating an authentication request. As Dr. Malek explains:

The claim begins with limitation [1.2], which requires the system to “receive a request to authenticate a client, wherein the request

comprises an identifier and a password.” The subsequent limitations repeatedly refer back to and operate upon that request and its components. For example, limitation [1.3] stores information about “*the* request,” and limitation [1.4] evaluates “*the* password” and “*the* identifier” provided in that request. Limitations [1.5]-[1.7] then determine whether “an *additional* verification is required to grant access,” including by “retrieving ... historical information about *previous* access requests associated with the user account.” Critically, Dr. Black testified and I agree that “the additional verification is additional relative to the verification of the ... identifier and password recited in [limitation [1.2]].” Ex. 2038 [Black-Tr.] 106:23-107:4. Dr. Black likewise testified that “previous” is temporally relative to the “current” authentication event recited in limitation [1.2]. *Id.*, 107:6-108:10 (the term “previous” “means previous to the current first limitation where a client is trying to be authenticated. This is an access request that was made previously to that in time. In other words, it happened before the current access request.”).

Thus, the logic and grammar of the claim tie the determination in limitations [1.5]-[1.7] to the current authentication request received in limitation [1.2]. The historical information may already exist—and may have been analyzed or otherwise processed before the current request—but the claim requires that determining whether additional verification is required for that current request itself comprise retrieving the historical information from the MDTSDb. The claimed retrieval therefore is not a freestanding historical analysis that may occur independently of the current authentication request.

Ex. 2037 [Malek-2nd-Decl.] ¶¶ 69-70.

Thus, the system receives the current authentication request and then determines whether the client must complete verification additional to the identifier-and-password verification before access is granted. The claims specify how that determination is made: it “comprises” retrieving historical information and determining whether the user account is associated with the recited anomalous previous access request. Ex. 1001 [’934] cl. 1. The historical information itself need not be newly created or first analyzed after the current request. But an analysis performed independently of the current request is not itself the claimed determination whether additional verification is required to grant access for that request. The claim requires that the “retrieve” and “determine” steps happen after receiving the request to authenticate.

Federal Circuit law gives effect to these express relationships among claim limitations. An ordering is required where “the claim language, as a matter of logic or grammar,” requires it. *Mformation Techs., Inc. v. Rsch. in Motion Ltd.*, 764 F.3d 1392, 1398-99 (Fed. Cir. 2014). Recent Federal Circuit precedent illustrates two circumstances particularly relevant here. First, later language that refers back to an earlier-recited object may require the earlier operation to have occurred. Most recently, *Sound View Innovations, LLC v. Hulu, LLC* applied that principle to claim language requiring “receiving a request” followed by an

operation on the “requested” object. 166 F.4th 958, 965 (Fed. Cir. 2026). The Federal Circuit explained that “‘requested’ is not only a grammatical descriptor, but also is a status indicator reflecting a completed action—the receiving of a request.” *Id.* Because the later limitation referred to the “said requested” object, it “necessarily depend[ed] on the first limitation having been performed.” *Id.*

The Federal Circuit likewise requires an order where the later operation logically presupposes the earlier one. *Amgen Inc. v. Sandoz Inc.*, 923 F.3d 1023, 1028 (Fed. Cir. 2019) (“the claim language logically requires that the process steps, lettered (a) through (g), be performed in sequence. For example, expressing the protein in a non-mammalian cell (limitation (a)) obviously must occur before the step of lysing that cell (limitation (b)).”); *Wi-LAN, Inc. v. Apple Inc.*, 811 F.3d 455, 462 (Fed. Cir. 2016) (“The text of the asserted claims requires randomizing the modulated data symbols before combining them.”).

That reasoning applies here. The claims first introduce the current authentication request and its identifier and password. The ensuing limitations repeatedly operate on that same request and those same credentials. And the system then determines whether verification “additional” to that identifier-and-password verification is required “to grant access.” Dr. Black agreed that “additional” refers to the identifier-and-password verification recited in limitation [1.2], and that the “previous” access request is previous to the current

authentication request. Ex. 2038 [Black-Tr.] 106:23-108:10. The determination whether additional verification is required therefore necessarily follows, and is made in response to, the current authentication request.

The specification confirms the same request-driven sequence. In Figure 6, a user requests access and provides credentials before the server determines what verification methods are suitable and ultimately grants access. As Dr. Malek explains:

The '934 patent's Figure 6 depicts a process in which a **user first requests access** (steps 603 and 606) before **the server determines the required verification score** (step 615) and **verifies the user** (step 618):

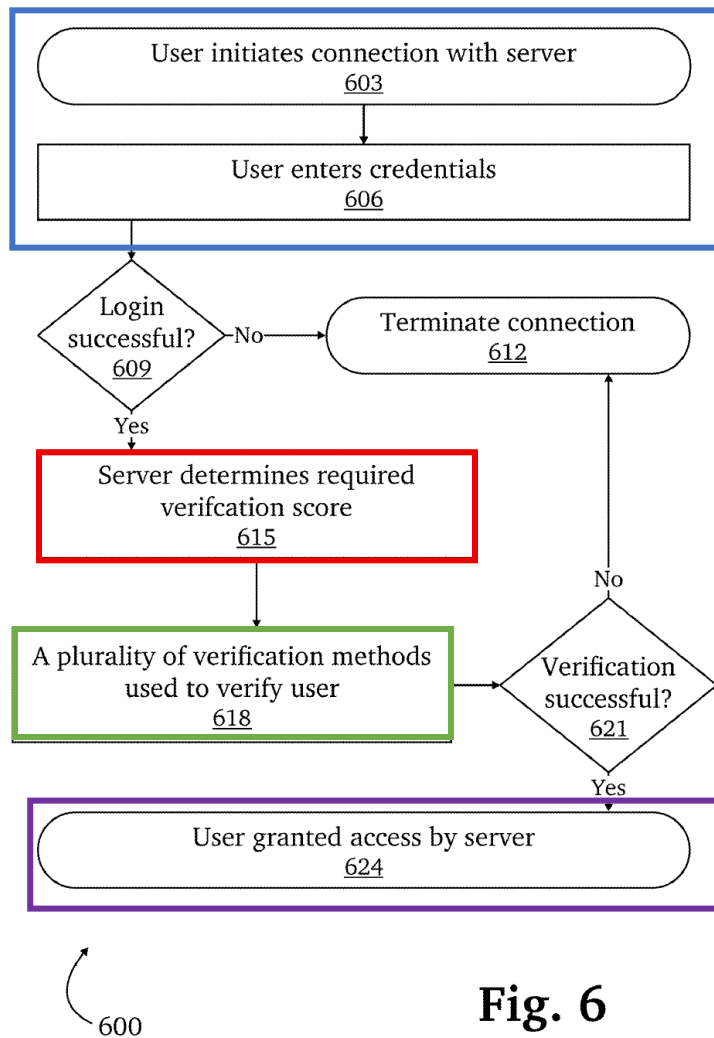


Fig. 6

Ex. 1001 ['934] Fig. 6 (annotated). If the verification is successful, **the user is granted access by the server** (step 624).

The '934 patent's corresponding written description also confirms this order. The '934 patent teaches that "[a]t an initial step 603, a user requests access from a server." *Id.*, 11:63-64. Then, "[a]t step 606, the server requests login credentials from the user." *Id.*, 11:65-66. *"If the login is successful at step 609, the server dynamically determines a required verification score required before the user can access the server at step 615."* *Id.*, 12:2-4. Then, "[a]t step

618, a plurality of verification methods may be used to verify the user.”
Id., 12:9-10. “If the verification is successful at step 621, the user is
granted access at step 624.” *Id.*, 12:18-19.

While Figure 6 does not separately illustrate every sub-operation
recited in limitations [1.6]-[1.7], it nevertheless reflects the request-
driven authentication architecture required by the claims. The server
receives the user’s current access request and credentials, then
determines the verification required for that access attempt, performs
that verification, and grants access if verification succeeds. The claims
further specify that the determination of whether additional verification
is required “comprises” the historical-information retrieval recited in
limitation [1.6].

Ex. 2037 [Malek-2nd-Decl.] ¶¶ 73-75.

The specification’s general statement concerning the ordering of method
steps does not alter this claim-specific analysis. The specification states that
“[w]here a **method** is claimed below, the steps of the method are intended to be
construed without any order unless it is impossible to perform a given step before
another.” Ex. 1001 [’934] 18:5-7. Patent Owner does not contend that every
recited operation must occur in the precise order written. Nor does Patent Owner
contend that historical information cannot exist or be analyzed before the current
authentication request. But the claim language itself makes it “impossible to
perform a given step before another”: the system receives the current

authentication request and then determines whether verification additional to the identifier/password authentication is required to grant access, and that determination expressly “comprises” retrieving information about access requests before the current request. The specification’s generic statement concerning order therefore does not sever the relationships the claim language itself creates. Moreover, by its terms, that statement concerns “[w]here a method is claimed”; it therefore does not alter the limitations of system claims 1 and 18. *See id.*, cls. 1, 18.

Accordingly, the claims require a request-driven determination: after receiving the current request to authenticate a client, the system determines whether additional verification is required to grant access for that request. That determination must comprise the recited retrieval and use of historical information. It is not enough that a system separately analyzes historical activity for threats and, at some other time or in some other context, may impose additional authentication.

2. Kirti’s Historical Threat Analysis Is Performed Independently Of The Current Authentication Request.

Petitioner fails to demonstrate that Kirti performs the claimed determination—including retrieving historical information from repository 211—in response to receiving the current authentication request. Instead, Petitioner

relies on Kirti's threat-detection processes, which collect and analyze activity information independently of any particular, current authentication request.

For limitation [1.5], Petitioner argues that Kirti "analyzes a user's historic login activity to identify potential threats and determine whether to apply 'remedial measures, such as adding additional steps to authentication.'" Pet., 34. And for limitations [1.6]-[1.7], Petitioner relies on Kirti's "analytics and security intelligence processes," which allegedly "retriev[e] historic user login data from Kirti's analytics and threat repository database" and use that information to detect anomalous behavior relative to baseline user profiles. *Id.*, 38-41. Nothing in those processes, however, identifies receiving the current authentication request as constituting the trigger for determining whether additional verification is required or for retrieving historical information.

To the contrary, Kirti describes collecting activity information from cloud applications as an independent data-gathering process. As Dr. Malek explains:

Petitioner relies upon Kirti's Figure 4 as demonstrating storing and retrieving activity data in repository 211. *See, e.g.*, Pet., 43-44. In Kirti's Figure 4 process, a cloud data loader connects to cloud applications and collects activity data. Ex. 1004 [Kirti] 10:2-17, Fig. 4. Kirti explains that "[i]n several embodiments," that collection "is scheduled to occur periodically (e.g., every 4 hours or every 6 hours)." *Id.*, 10:15-17. Dr. Black likewise characterized Figure 4 as a "burst of

activity that occurs every four to six hours,” explaining that once its “collection, storage analysis, and report generation” are complete, “that task is complete” and the system waits until the next time the process runs. Ex. 2038 [Black-Tr.] 43:9-19. Kirti’s Figure 4 process is not initiated by receipt of a request to authenticate a client.

Significantly, Petitioner relies on Kirti’s disclosed batch processing. Petitioner argues that Kirti’s anomalous login attempts qualify as “previous” access requests because they come from “past activity data that is collected in batches,” such as a batch containing the most recent 24-hour period. Pet., 45-46. Petitioner even explains that if Kirti identifies a “current login attempt as anomalous,” that login will become “previous” “by the time of the next login attempt where the system would require additional verification.” *Id.*, 46. That theory confirms that Kirti detects and stores activity independently, and Petitioner relies on that previously performed analysis at some later authentication. It does not identify the claimed retrieval from repository 211 as part of determining whether additional verification is required for the current request.

Ex. 2037 [Malek-2nd-Decl.] ¶¶ 78-79.

As Dr. Malek further explains, Kirti’s batch processing is different from the process recited in the claims:

In contrast to Kirti, the claims require that, as part of determining whether additional verification is required for the current authentication request, the system retrieves historical information from the MDTSDB.

Kirti's Figure 4 process instead collects activity information from external cloud applications and places that information into repository 211. Nothing in Figure 4 teaches retrieving historical information from repository 211 because a current authentication request has been received.

Ex. 2037 [Malek-2nd-Decl.] ¶ 80.

Moreover, as Dr. Malek explains, Kirti's Figure 5B does not show Kirti's threat analysis as connected with, let alone performed in response to, a current authentication request:

Petitioner relies heavily upon Kirti's Figure 5B as allegedly “show[ing] an exemplary process in which activity data is retrieved and used to generate threat models (step 506) that ‘include baseline user profiles over various periods of time.’” Pet., 42. But Figure 5B is not initiated in response to any authentication request. Indeed, no such request appears in the figure or Kirti's corresponding description:

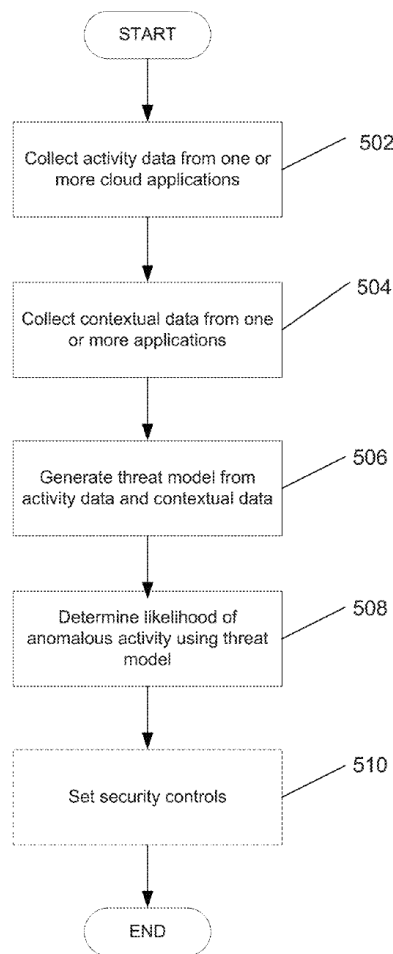


FIG. 5B

Ex. 1004 [Kirti] Fig. 5B.

Rather, as with Figure 4, Kirti's system generates a threat model and sets appropriate security controls in steps 506-510 when data is collected from the cloud applications in steps 502-504. Figure 5B begins by collecting activity and contextual data (steps 502-504), generates a threat model from that data (step 506), determines the likelihood of anomalous activity using the threat model (step 508), and then sets security controls (step 510). Ex. 1004 [Kirti] Fig. 5B, 24:8-

50. Nothing in Figure 5B or its description says an authentication request triggers it.

Ex. 2037 [Malek-2nd-Decl.] ¶¶ 81-82.

Petitioner also relies on Kirti's Figure 5, arguing that threat detection and prediction analytics application 212 retrieves historical activity data from repository 211 to "[d]iscover threats and risks." Pet., 44-45. But Figure 5 does not connect that retrieval to a current authentication request either. It shows repository 211 supplying data for threat analysis; it does not teach that the data is retrieved because a request to authenticate a client has been received or as part of determining whether additional verification is required to grant access to that request. Thus, Figure 5 supplies the retrieval from repository 211 that Petitioner identifies, but not the request-driven relationship the claims require.

Petitioner itself relies upon Kirti's batch architecture, arguing that the anomalous login attempts are "previous" because they come from "past activity data that is collected in batches." Pet., 45-46. Petitioner's cited evidence establishes at most that Kirti periodically collects and analyzes activity concerning multiple users independently of any particular authentication request. It does not establish that, upon receiving a current authentication request, Kirti retrieves historical information from repository 211 as part of deciding whether additional verification is required for that request.

Thus, Petitioner identifies two different kinds of retrieval in Kirti, neither of which establishes the claimed operation. Figure 4 periodically retrieves activity data from the cloud applications and stores that data in repository 211. Ex. 1004 [Kirti] 10:2-24, 10:57-61, Fig. 4. Figures 5 and 5B, in turn, use collected activity data for threat analysis, including retrieval of historical activity from repository 211. Pet., 42-45. But Petitioner identifies no disclosure tying that retrieval from repository 211 to receipt of a current authentication request or showing that it occurs as part of determining whether additional verification is required to grant access to that request. *Id.* Kirti's periodic collection and independent threat analysis therefore do not establish the request-driven determination the claims require.

Petitioner cannot bridge the gap by treating Kirti's threat model or baseline profile as pre-analyzed historical information that is simply available when a later authentication occurs, as it attempted to do in its Preliminary Reply. Paper 16, 3. As Dr. Malek explains:

Even assuming that pre-analyzing historical information is permissible, the claims still require the current determination to comprise retrieving historical information from the MDTSDDB. And Dr. Black testified that Kirti does not even say what happens to its threat model after Figure 5B. He suggested that the model could be stored to avoid repeating computations, but acknowledged that Kirti does not say

whether it is stored, and whether it would be placed “into the repository 211 or a separate place” would be “a design decision.” Ex. 2038 [Black-Tr.] 57:17-58:24. Thus, even a theory based on pre-analyzed historical information does not establish that, upon receiving the current authentication request, Kirti retrieves the claimed historical information from repository 211 as part of deciding whether additional verification is required.

Ex. 2037 [Malek-2nd-Decl.] ¶ 86.

None of Kirti’s disclosures of various ways of collecting and analyzing user activity establishes the claimed request-driven operation: receiving the current authentication request and, as part of determining whether additional verification is required to grant access to that request, retrieving historical information from the MDTSDB.

3. Kirti Does Not Determine That Additional Verification Is Required To Grant Access To The Current Authentication Request.

Although Petitioner does not rely upon these disclosures in its Petition, its Preliminary Reply invoked Kirti’s “real time” disclosures. Paper 16, 4. Kirti states that “[i]n further embodiments, data is collected and retrieved in real time as events occur,” and that activity associated with certain “high risk events” may be retrieved “near real-time outside scheduled retrieval.” Ex. 1004 [Kirti] 10:17-24.

Such post-petition reliance upon a new theory is improper. *Intelligent Bio-Sys., Inc. v. Illumina Cambridge, Ltd.*, 821 F.3d 1359, 1369 (Fed. Cir. 2016).

Even if considered, however, Kirti’s “real-time” disclosures do not cure the deficiency. First, they concern only the timing of activity-data collection and retrieval; they do not tie Kirti’s threat analysis to receipt of a current authentication request. Second, even assuming that Kirti performs its threat analysis in real time as part of evaluating a current authentication request, Kirti still does not determine that additional verification is required to grant access to that request.

As to the first point, Dr. Malek explains:

Kirti’s “real-time” disclosures are only connected to collecting and retrieving activity data, not to threat analysis. Ex. 1004 [Kirti] 10:19-24, 10:57-59. Kirti states that activity data may be “collected and retrieved in real time as events occur” and that certain high-risk activity may be retrieved “near real-time outside scheduled retrieval.” *Id.* Nothing in those passages states that Kirti analyzes the retrieved data in real time, much less that such analysis occurs as part of evaluating a current authentication request or determining whether additional authentication is required to grant access. Indeed, Kirti states that the retrieved data is subsequently stored in repository 211. *Id.*, 10:57-61.

Ex. 2037 [Malek-2nd-Decl.] ¶ 89.

Nor, even assuming real-time threat analysis of a current authentication event, does Figure 5B teach the claimed determination. As Dr. Malek explains:

Figure 5B determines the likelihood of anomalous activity and then generically “[s]et[s] security controls” at step 510. Ex. 1004 [Kirti] Fig. 5B. Kirti explains that the resulting risk scores may be used to “prioritize remediation actions,” giving as examples “resetting passwords, restricting access from foreign countries, and/or suspending accounts.” *Id.*, 24:39-50. Thus, even if Figure 5B were performed in real time, it would establish only that Kirti detects anomalous activity and selects some security response—not that Kirti determines that additional authentication is required. Kirti mentions “additional steps to authentication” and “elevated authentication or OTP validation” elsewhere, *id.*, 5:25-37, 28:31-34, but Petitioner identifies no disclosure tying those particular responses to the Figure 5B determination.

Ex. 2037 [Malek-2nd-Decl.] ¶ 90.

Petitioner’s reliance on Kirti’s specific additional-authentication disclosures fares no better. Petitioner argues that Kirti’s selection of a remediation method that may include “additional steps to authentication” teaches the claimed additional verification, and further contends that where the selected action is a one-time password, the client necessarily would be prompted to complete it. Pet., 49-50. But even assuming that Kirti determines whether additional verification is required as part of evaluating a current authentication request, Kirti does not apply its

disclosed “additional steps to authentication” in deciding whether to grant access to that request. Those disclosures arise in a different context: *after* a threat is detected and in connection with *another* cloud application.

As Dr. Malek explains:

Kirti discusses additional verification only twice. Pet., 23 (citing Ex. 1004 [Kirti] 28:33 and 5:34). Both disclosures concern a first cloud application detecting a threat and alerting a second cloud application of the threat, which may thereafter impose additional authentication requirements. Kirti explains:

One cloud application can *proactively warn another cloud application of a potential threat*. ... When a threat is identified in a first cloud (e.g., a query from a blocked IP address), a cloud security monitoring and control system in accordance with embodiments of the invention can automatically *notify a second cloud* that is part of the business process. The notification can include a request or recommendation for a higher level of security controls, such as elevated authentication or OTP validation, in the business process.

Ex. 1004 [Kirti] 28:24-34.

Kirti’s disclosure of single sign-on (“SSO”) does not alter this distinction. Kirti states only that profile-linking algorithms may identify a single user across multiple clouds using a common identifier or an “SSO authentication mechanism,” thereby allowing activities

associated with that user to be correlated across applications. Ex. 1004 [Kirti] 13:47-51. But identifying the same user across clouds does not make separate authentication events a single authentication request. Kirti still describes a threat detected in one cloud followed by a notification to a second cloud, where heightened authentication may thereafter be imposed. *Id.*, 28:24-34. Nor does Kirti teach that, upon receiving an authentication request at the second cloud, the system retrieves historical information from repository 211 as part of determining whether additional verification is required to grant that request. At most, the second cloud may impose additional verification based on the notification from the first cloud—not based on the claimed retrieval from repository 211.

Thus, Kirti's additional verification is not used to determine whether to grant access to the authentication request being evaluated. It is instead a potential security response imposed by a different cloud after a threat has already been detected.

Ex. 2037 [Malek-2nd-Decl.] ¶¶ 92-94.

Petitioner therefore fails to identify any disclosure in Kirti where, after receiving and as part of evaluating an authentication request, Kirti determines whether additional verification is required to grant access to that request—much less does so by retrieving historical information and prompting the client to complete the additional verification, as required by the claims.

C. Petitioner Fails To Demonstrate Selection Of An Additional Verification Method “From A Plurality Of Verification Methods” (All Claims, All Grounds).

The claims require more than performing some form of additional verification. Once additional verification is required, the system must “select an additional verification method *from a plurality of verification methods*,” then prompt the client to complete “the additional verification method” selected and determine whether that same method “has been completed correctly.” Ex. 1001 [’934] cls. 1, 11, 18, 26. Petitioner’s theories do not establish the claimed selection. For Kirti, Petitioner identifies a selection from a plurality of *remedial* actions, only one of which it characterizes as involving additional authentication. Pet., 49-50. One verification method is not a “*plurality* of verification methods.” *Id.*; see Section II.C.1. For Coffin, Petitioner identifies a general list of authentication techniques, but the only actual selection rule on which it relies chooses among communication channels for delivery of a passcode. Pet., 50-52. Selecting where to deliver the same passcode is not selecting among a plurality of verification methods. See Section II.C.2. In neither case does Petitioner identify selection of a verification method from a plurality of verification methods.

1. Kirti Selects Among Remedial Actions, Not Among A Plurality Of Verification Methods.

Petitioner first argues that Kirti alone “teaches” “select[ing] an additional **verification** method from a plurality of verification methods” through its disclosure of “recommending and performing a **remedial action** when a threat is detected.”

Pet., 49. Specifically, Petitioner argues that:

Kirti teaches recommending and performing a **remedial action** when a threat is detected and mentions that one such remedial action is “adding additional steps to authentication,” i.e., requiring completion of an additional verification step (e.g., enter a one-time password). *E.g.*, EX1004, 5:27-37, 28:24-36. EX1003, ¶112.

Kirti discloses manually or automatically **selecting which one of a plurality of remedial actions** to use. ...

Id.

Petitioner’s Kirti theory does not identify the selection the claims require. Even accepting Petitioner’s characterization of “adding additional steps to authentication” as a verification method, Petitioner identifies **only one verification method** within a broader set of remediation actions. Selecting that one action from a menu that also includes changing passwords, blocking IP addresses or senders, locking or deactivating accounts, and other security responses is not selecting a verification method “from **a plurality** of verification methods.”

Moreover, Kirti's disclosures confirm the distinction. As Dr. Malek explains, Kirti discloses several remediation measures, but only one is even arguably tied to further authentication:

Kirti's selection among remedial actions is different from the selection required by the claims. Kirti describes a variety of responses that may be taken after a threat is identified. Those responses include, for example, blocking an IP address, resetting a password, suspending or locking an account, and adding additional authentication. Ex. 1004 [Kirti] 5:27-37, 25:45-49. These measures may all remediate a security threat, but they do so in different ways. Blocking an IP address or locking an account, for example, does not verify that a client is who the client claims to be.

Petitioner identifies only one member of Kirti's remediation menu—"adding additional steps to authentication"—as an additional verification method. Pet., 49. Thus, even accepting Petitioner's characterization of that action, Kirti discloses selecting one possible verification-related action from a plurality of remediation actions. It does not disclose a plurality of verification methods from which an additional verification method is selected.

Ex. 2037 [Malek-2nd-Decl.] ¶¶ 100-101.

The surrounding claim language reinforces this distinction. The selected "additional verification method" is something the client is thereafter "prompted to complete," and the system determines whether that method "has been completed

correctly.” Ex. 1001 [’934] cl. 1. Kirti’s other remedial actions—blocking an IP address, deactivating an account, resetting a password, or locking an account—are actions taken by the system to address a threat. They are not verification methods that the client is prompted to complete and that the system then determines were completed correctly. Nor is it sufficient that only one of Kirti’s remedial actions might qualify as a verification method. Petitioner has not identified even a second verification method in the set from which Kirti makes its alleged selection.

Kirti therefore does not disclose “select[ing] an additional verification method from a plurality of verification methods.”

2. Coffin Does Not Cure Kirti’s Failure To Select From A Plurality Of Verification Methods.

Petitioner’s Coffin theory also fails. To identify a “plurality of verification methods,” Petitioner first points to Coffin’s general discussion of numerous authentication techniques, including PINs, CAPTCHAs, personal questions, biometrics, and passcodes. Pet., 50-51. But Petitioner identifies no disclosure in which Coffin’s system selects among that plurality. When Petitioner turns to the claimed “select[ing],” it relies instead on a different disclosure: Coffin’s rule for sending a passcode to a pager and cell phone if available, and otherwise to email. *Id.*, 51-52. Thus, the plurality Petitioner identifies and the set from which Coffin actually selects are different.

Petitioner's own treatment of the ensuing limitations demonstrates the problem. Immediately after characterizing Coffin's choice of pager, cell phone, or email as selection of the "verification method," Petitioner explains that, "[o]nce the additional verification method is selected," Coffin sends the passcode to the selected device, after which the user is prompted to enter the passcode and the system determines whether the entered code is correct. Pet., 52. In other words, the thing the client actually completes and the system validates is the passcode verification, while pager, cell phone, or email determines how that passcode reaches the user. Petitioner never explains why selecting a channel through which the same verification challenge is delivered constitutes selecting a different "verification method" for purposes of limitation [1.9].

Indeed, as Dr. Malek explains, Coffin never teaches selecting among the different authentication techniques it lists and only concerns where to send a passcode:

Although Coffin lays out a list of the "many things that are being done in computer security to attempt to achieve" authentication, that list is disembodied and not tethered to any particular system. Ex. 1005 [Coffin] 177. In other words, Coffin's list of authentication methods is not a "plurality of verification methods" from which a particular system will select an additional verification method as required by the claims. Rather, it is a list of methods being attempted in computer security.

Unsurprisingly, Coffin never mentions or teaches how a system would “select an additional verification method” from Coffin’s general list of authentication methods.

Instead, the only selection Coffin discusses concerns where to send a passcode. If a pager or cell phone is available, Coffin sends the code there; otherwise it sends the code by email. Ex. 1005 [Coffin] 200. Regardless of that choice, the authentication process remains the same in the relevant respect: a passcode is generated and delivered to the user, the user supplies that code, and Coffin determines whether the supplied code is correct. *Id.*, 183, 194, 200-01, 206-07. Thus, Coffin’s selection determines the channel or mechanism used to deliver the passcode. It does not disclose a system selecting, for example, between a passcode, biometric authentication, a CAPTCHA, and a personal question—the different authentication techniques Petitioner relies upon as its alleged “plurality of verification methods.”

Ex. 2037 [Malek-2nd-Decl.] ¶¶ 106-107.

The ’934 patent further supports that distinction. As Dr. Malek explains, the patent expressly distinguishes the one-time-code verification from the mechanism used to deliver that code:

The specification states that “one-time-use codes 415g” may be sent “through a text message or email,” generated on-demand on a mobile device, or take the form of a uniquely generated hyperlink, and expressly characterizes these alternatives as “[v]arious implementations of the one-time-use code.” Ex. 1001 [’934] 11:21-27.

Similarly, the Background discusses “one-time use codes sent to a user’s mobile device or email” and then identifies a problem with “over-reliance on a single method of *delivery*.” *Id.*, 2:23-34. Consequently, the ’934 patent distinguishes between the “one-time-use code” method and the “method of delivery.”

Ex. 2037 [Malek-2nd-Decl.] ¶ 110.

Petitioner’s generalized reliance on what was allegedly “well known” does not fill the gap in its asserted combination. The Petition separately states that selecting among different authentication methods based on risk, availability, cost, or user preference was known in the art. Pet., 8-9, 51-52. But demonstrating that other systems were known to select among authentication techniques is not the same as demonstrating that Petitioner’s asserted Kirti-Coffin combination does so. The Petition does not explain how those unasserted systems alter Kirti’s remediation architecture or Coffin’s passcode-delivery implementation so that the resulting system selects among a plurality of verification methods. Nor does it identify which plurality of verification methods is configured for selection in the proposed combination and how that selection occurs. A petitioner must provide “some articulated reasoning with some rational underpinning” for the claimed arrangement, not merely establish that its constituent concepts were independently known. *KSR Int’l Co. v. Teleflex Inc.*, 550 U.S. 398, 418 (2007); *see In re Magnum Oil Tools Int’l, Ltd.*, 829 F.3d 1364, 1380 (Fed. Cir. 2016).

Accordingly, Kirti does not disclose selecting from a plurality of verification methods; it discloses selecting among a plurality of remedial actions, only one of which Petitioner identifies as additional verification. Coffin does not cure that deficiency. Although Coffin generally describes multiple authentication techniques, Petitioner identifies no selection among them and instead relies on Coffin's selection of a delivery channel for a passcode. Petitioner therefore has not proven that any asserted ground teaches or renders obvious "select[ing] an additional verification method from a plurality of verification methods."

D. Petitioner Fails To Demonstrate That Kirti "Determine[s] That A Probable Cyberattack Is Detected" "Based On The Additional Verification Being Required To Grant Access" (Claims 5, 15, 21, 29; All Grounds).

Claims 5, 15, 21, and 29 require that the determination that a probable cyberattack is detected be "*based on* the additional verification being required to grant access." Ex. 1001 ['934] cls. 5, 15, 21, 29. It is not enough that the same suspicious activity independently causes both additional verification to be required and a cyberattack to be detected. The cyberattack determination must itself be "based on" the additional verification. Ex. 2037 [Malek-2nd-Decl.] ¶ 115.

Petitioner does not identify a disclosure in Kirti expressly establishing the claimed relationship. Instead, it argues that a POSITA would have understood or found it obvious that, if an anomalous login caused Kirti to require additional

verification, Kirti would also detect a probable cyberattack if the same account exhibited other anomalies, such as repeated failed login attempts. Pet., 58-59.

Kirti, however, teaches the opposite relationship as required by the claims. The claims require the cyberattack determination to be “based on” the determination that additional verification is required. Kirti instead first detects a threat or anomaly and only then determines what response to take.³ Its heightened-authentication disclosure follows that same sequence. As Dr. Malek explains:

Kirti’s threat-remediation process begins by “**identifying (602) a threat**” and thereafter in **steps 604-608, presenting an alert and identifying and performing an appropriate remediation action:**

³ As discussed in Section II.C.1, Petitioner identifies only one of Kirti’s remedial actions—“adding additional steps to authentication”—as an additional verification method. Pet., 49-50. Kirti’s other remedial actions are not verification methods.

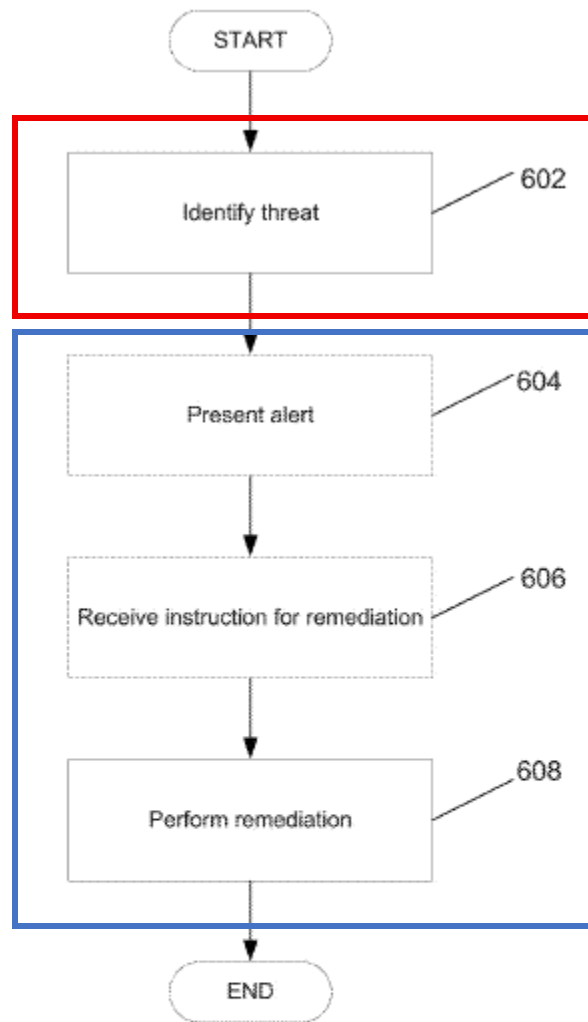


FIG. 6

Ex. 1004 [Kirti] 25:3-20, Fig. 6. Figure 5B likewise first “[g]enerate[s]
a) threat model” and “[d]etermine[s] likelihood of anomalous

activity using threat model” at steps 506-508 and then “[s]et[s]
security controls” at step 510:

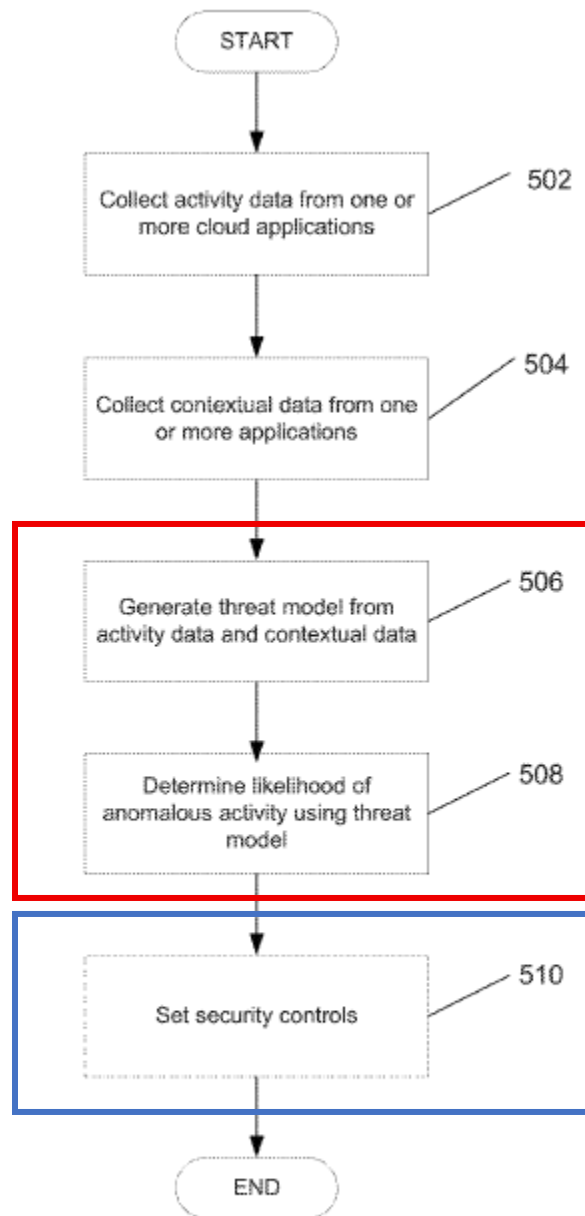


FIG. 5B

Id., Fig. 5B, 24:33-50. Both disclosed processes therefore proceed in the same direction: Kirti first identifies a threat or anomaly and then determines an appropriate response. Neither discloses determining that a probable cyberattack exists because the system has already determined that additional verification is required. And the additional-authentication disclosure on which Petitioner relies follows the same sequence: after a threat has been identified, the system may recommend heightened authentication as a security response. *Id.*, 28:24-34. In Kirti, therefore, threat detection supplies the reason for remediation; a determination requiring additional authentication does not supply the reason for detecting the threat.

Ex. 2037 [Malek-2nd-Decl.] ¶ 117; *see also id.*, ¶ 119; Section II.B.3; Ex. 2038 [Black-Tr.] 56:16-57:16.

Petitioner's claim 5 analysis does not identify a contrary disclosure. Again, Petitioner posits that, if a login attempt from a suspicious IP address causes additional verification to be required, Kirti "would also determine that a probable cyberattack is detected if that same user account is associated with other anomalies, such as repeated failed login attempts." Pet., 58-59. Petitioner then asserts that a brute-force attack would be detected based on the IP address "being required to provide additional verification" and having failed multiple login attempts. *Id.*, 59.

But Petitioner’s own explanation shows that the alleged cyberattack determination is based on the underlying suspicious activity—the suspicious IP address, repeated failed logins, and other anomalies—not on the determination that additional verification is required. The fact that the same suspicious facts might lead Kirti both to require additional verification and to detect a threat does not establish the relationship the claims recite. A common cause for two decisions is not the same as making the second decision “based on” the first. As Dr. Malek explains:

Kirti analyzes observed activity to determine whether a potential security threat exists. After a threat or anomaly is detected, Kirti may determine an appropriate security response. Thus, the same underlying facts could potentially influence both Kirti’s threat assessment and its remediation. But that does not mean Kirti detects the threat because it previously determined that additional authentication is required. To the contrary, Kirti teaches the opposite direction: the threat or anomalous behavior is identified first and supplies the reason for the subsequent remediation. Petitioner does not identify any disclosure in Kirti in which a decision requiring additional verification supplies the basis for determining that a probable cyberattack has been detected.

Ex. 2037 [Malek-2nd-Decl.] ¶ 118.

Petitioner thus identifies no disclosure or reasoned obviousness theory in which Kirti’s determination that additional verification is required supplies a basis

for determining that a probable cyberattack has been detected. Petitioner therefore fails to demonstrate that its grounds teach claims 5, 15, 21, and 29.

III. CONCLUSION

Petitioner fails to demonstrate by a preponderance of the evidence that any of the claims are unpatentable.

Respectfully submitted,

/ Kenneth J. Weatherwax /
Kenneth J. Weatherwax (Reg. No. 54,528)
Nathan Lowenstein, *pro hac vice*
Colette Woo, *pro hac vice*
LOWENSTEIN & WEATHERWAX LLP

Date: September 1, 2026

CERTIFICATE OF COMPLIANCE WITH TYPE-VOLUME LIMITS

This Patent Owner Response (the “POR”) consists of 11,742 words, excluding table of contents, table of authorities, certificate of service, this certificate, or table of exhibits. The POR complies with the type-volume limitation of 14,000 words as mandated in 37 C.F.R. § 42.24. In preparing this certificate, counsel has relied on the word count of the word-processing system used to prepare the paper (Microsoft Word).

Respectfully submitted,

/ Michael Cosley /

Date: September 1, 2026

CERTIFICATE OF SERVICE

The undersigned hereby certifies that the following documents were served
by electronic service, by agreement between the parties, on the date below:

PATENT OWNER'S RESPONSE

EXHIBITS 2037-2043

The names and addresses of the parties being served are as follows:

Andrew M. Mason	andrew.mason@klarquist.com
Frank Morton-Park	frank.morton-park@klarquist.com
Todd M. Siegel	todd.siegel@klarquist.com
Samuel B. Thacker	samuel.thacker@klarquist.com
	Msft-Qomplx@klarquist.com

Respectfully submitted,

/ Michael Cosley /

Date: September 1, 2026