

3UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

MICROSOFT CORPORATION,
Petitioner,

v.

QOMPLX LLC,
Patent Owner.

IPR2026-00326
U.S. Patent No. 12,301,628
Issued: May 13, 2025
Application No. 18/892,295
Filed: September 20, 2024

Title: CORRELATING NETWORK EVENT
ANOMALIES USING ACTIVE AND PASSIVE EXTERNAL
RECONNAISSANCE TO IDENTIFY ATTACK INFORMATION

PETITION FOR
INTER PARTES REVIEW OF U.S. PATENT NO. 12,301,628

TABLE OF CONTENTS

	Page
LIST OF EXHIBITS	v
MANDATORY NOTICES UNDER 37 C.F.R. § 42.8	vii
1. Real Party-In-Interest.....	vii
2. Related Matters	vii
3. Lead And Back-Up Counsel, And Service Information.....	vii
CLAIMS LISTING.....	ix
I. INTRODUCTION	1
II. GROUNDS FOR STANDING PER 37 C.F.R. § 42.104(a)	2
III. IDENTIFICATION OF CHALLENGE	2
IV. LEVEL OF SKILL IN THE ART, AND STATE OF THE ART	2
A. Person Of Ordinary Skill In The Art.....	2
B. State Of The Art	3
V. THE '628 PATENT.....	3
A. The '628 Patent Disclosures In Common With Crabtree	4
B. The Prosecution History	6
C. The Challenged Claims' Earliest Possible Effective Filing Date Is April 22, 2021	6
VI. CLAIM CONSTRUCTION	8
VII. GROUND 1: CLAIMS 1, 4-6, 9-11, 13-15, 17, 20, AND 22 ARE OBVIOUS OVER BREZINSKI (EX1004) AND CRABTREE (EX1005).....	8
A. Brezinski.....	8
B. Crabtree	12

C.	Combining Brezinski And Crabtree.....	12
D.	Claim 1	17
1.	Element [1.1].....	17
2.	Element [1.2].....	17
3.	Element [1.3].....	19
4.	Element [1.4].....	22
5.	Element [1.5].....	26
6.	Element [1.6].....	28
7.	Element [1.7].....	33
8.	Element [1.8].....	34
9.	Element [1.9].....	37
10.	Element [1.10].....	39
a)	[1.10.a].....	39
b)	[1.10.b].....	45
c)	[1.10.c].....	46
d)	[1.10.d].....	47
11.	Element [1.11].....	50
12.	Element [1.12].....	54
13.	Element [1.13].....	58
E.	Claim 4	59
F.	Claim 5	60
G.	Claim 6	63
H.	Claim 9	65
I.	Claim 10	66

J.	Claim 11	68
1.	[11.1]-[11.5], [11.8], [11.11] And [11.15]-[11.17].....	68
2.	[11.6]	68
3.	[11.7]	71
4.	[11.9]	72
5.	[11.10]	73
6.	[11.12]	73
7.	[11.13]	74
8.	[11.14]	74
K.	Claim 13	75
L.	Claim 14	76
M.	Claims 15 & 17.....	76
N.	Claim 20	77
O.	Claim 22	77
VIII.	NO OBJECTIVE INDICIA OF NON-OBVIOUSNESS.....	78
IX.	CONCLUSION.....	78
	CERTIFICATE OF COMPLIANCE.....	79
	CERTIFICATE OF SERVICE	80

TABLE OF AUTHORITIES

Page(s)

Cases

Dynamic Drinkware, LLC v. Nat’l Graphics, Inc.,
800 F.3d 1375 (Fed. Cir. 2015) 7

In re NTP, Inc.,
654 F.3d 1268 (Fed. Cir. 2011) 7

Board Decisions

Penumbra, Inc., v. Rapidpulses, Inc.,
IPR2021-01466, Paper 34 (PTAB Mar. 10, 2023).....7

Statutes

35 U.S.C. § 103 2

Rules

37 C.F.R. § 42.104 2

37 C.F.R. § 42.24 79

37 C.F.R. § 42.6 80

LIST OF EXHIBITS

No.	Description
1001	U.S. Patent No. 12,301,628 (“ 628 patent ”)
1002	File History of U.S. Patent No. 12,301,628
1003	Declaration of John Black, dated April 3, 2026 (“ Black Decl. ” or “Black”)
1004	U.S. Patent No. 9,225,730 (“ Brezinski ”)
1005	U.S. Patent App. Pub. No. 2018/0219919A1 (“ Crabtree ”)
1006	U.S. Patent App. Pub. No. 2022/0014560A1
1007	U.S. Patent App. Pub. No. 2020/0296137A1
1008	William Eberle & Lawrence Holder, <i>Incremental Anomaly Detection in Graphs</i> , IEEE 13 th Int’l Conference on Data Mining Workshops (IEEE 2013)
1009	Leman Akoglu <i>et al.</i> , <i>Graph Based Anomaly Detection Description: A Survey</i> , Data Mining and Knowledge Discovery, Vol. 29, Issue 3 (Springer 2014)
1010	Listing and Comparison of ’628 Patent Challenged Claims
1011	Plaintiff Qomplx’s Preliminary Infringement Contentions, <i>Qomplx LLC v. Microsoft Corp.</i> , Case No. 25-CV-01383-ADA (W.D. Tex. Nov. 24, 2025)
1012	Defendant’s Preliminary Claim Constructions, <i>Qomplx LLC v. Microsoft Corp.</i> , Case No. 25-CV-01383-ADA (W.D. Tex. Mar. 2, 2026)
1013	Plaintiff Qomplx LLC’s Preliminary Disclosure of Proposed Claim Constructions, <i>Qomplx LLC v. Microsoft Corp.</i> , Case No. 25-CV-01383-ADA (W.D. Tex. Mar. 2, 2026)
1014	Microsoft Computer Dictionary, 5 th Ed. (2002)
1015	U.S. Patent App. Pub. No. 2019/0081876A1

No.	Description
1016	U.S. Patent App. Pub. No. 2018/0069885A1
1017	U.S. Patent No. 10,909,241
1018	U.S. Patent App. Pub. No. 2018/0367548A1
1019	Thomas H. Cormen, <i>et al.</i> , <i>Introduction to Algorithms</i> , 3d Ed. (MIT Press 2009) (excerpt)
1020	Redline comparison of specification of the '628 patent (EX1001) to specification of the Crabtree prior art (EX1005)
1021	Complaint For Patent Infringement, <i>Qomplx LLC v. Microsoft Corp.</i> , Case No. 25-CV-01383-ADA (W.D. Tex. Aug. 28, 2025)
1022	U.S. Patent App. Pub. No. 2020/0177617A1
1023	<i>Breadth-First Search</i> , Carnegie Mellon Univ. School of Computer Science (Apr. 29, 2015), available at: https://www.cs.cmu.edu/afs/cs/academic/class/15210-s15/www/lectures/bfs-notes.pdf

MANDATORY NOTICES UNDER 37 C.F.R. § 42.8

1. Real Party-In-Interest

Microsoft Corporation is the sole real party-in-interest.

2. Related Matters

The '628 patent (EX1001) has been asserted against Microsoft Corporation in the following district court litigation (the “district court case”):

- *Qomplx LLC v. Microsoft Corporation*, Case No. 1-25-cv-01383 (W.D. Tex.), filed August 28, 2025.

3. Lead And Back-Up Counsel, And Service Information

Lead Counsel	Back-up Counsel
Andrew M. Mason, Reg. No. 64,034 andrew.mason@klarquist.com	Cameron D. Clawson, Reg. No. 73,509 cameron.clawson@klarquist.com Caitlin C. Potter, Reg. No. 72,477 caitlin.potter@klarquist.com Frank Morton-Park, Reg. No. 80,750 frank.morton-park@klarquist.com Todd M. Siegel, Reg. No. 73,232 todd.siegel@klarquist.com Samuel B. Thacker, Reg. No. 78,633 samuel.thacker@klarquist.com
KLARQUIST SPARKMAN, LLP 121 SW Salmon Street, Suite 1600 Portland, Oregon, 97204 503-595-5300 (phone) 503-595-5301 (fax)	

Petitioner consents to service via email at the above email addresses and the email address of Msft-Qomplx@klarquist.com.

Pursuant to 37 C.F.R. § 42.10(b), concurrently filed with this Petition is a Power of Attorney executed by Petitioner and appointing the above counsel.

Petitioner authorizes Account No. 02-4550 to be charged for any fees, including those enumerated in 37 C.F.R. § 42.15.

CLAIMS LISTING

Challenged claim 1 is representative and listed here, with element numbering added in the lefthand column. A full claims listing is submitted as Exhibit 1010.

[1.1]	A computer system comprising:
[1.2]	a hardware memory, wherein the computer system is configured to execute software instructions stored on nontransitory machine-readable storage media comprising software instructions that:
[1.3]	store in the hardware memory a representation of a first graph, wherein the representation of the first graph comprises representations of a first plurality of nodes corresponding to a first plurality of entities and further comprises representations of a first plurality of edges, wherein the first graph is a directed graph,
[1.4]	wherein the first plurality of entities comprises a plurality of accounts and a plurality of resources,
[1.5]	wherein each edge of the first plurality of edges corresponds to a respective relationship between a respective pair of entities of the first plurality of entities;
[1.6]	receive streaming data comprising time-stamped data about events relating to one or more entities of the first plurality of entities;
[1.7]	based on a first portion of the streaming data, identify a first entity that does not correspond to any of the first plurality of nodes, wherein the first entity is not of the first plurality of entities;
[1.8]	based on a second portion of the streaming data, wherein the second portion is not identical to the first portion, identify a first relationship between a pair of entities of the first plurality of entities that does not correspond to any of the first plurality of edges;
[1.9]	modify, in the hardware memory, the representation of the first graph to generate a modified representation of the first graph, wherein the modified representation of the first graph comprises a representation of a first node corresponding to the first entity and a representation of a first edge corresponding to the first relationship, wherein the first node is not of the first plurality of nodes and the first edge is not of the first plurality of edges;

[1.10]	for an anomalous event associated with a node in the modified representation of the first graph, perform a first correlation using the modified representation of the first graph to identify a first plurality of correlated nodes, wherein each of the first plurality of correlated nodes corresponds to a respective event or resource, wherein each respective event or resource is associated with the anomalous event, and wherein each of the first plurality of correlated nodes is connected by a respective edge of a second plurality of edges to the node associated with the anomalous event in the modified representation of the first graph;
[1.11]	for one or more of the first plurality of correlated nodes, perform a further correlation using the modified representation of the first graph to identify a second plurality of correlated nodes, wherein each of the second plurality of correlated nodes is connected through a respective edge of a third plurality of edges to the respective node of the first plurality of correlated nodes in the modified representation of the first graph;
[1.12]	generate a representation of a second graph comprising representations of one or more of the first plurality of correlated nodes, representations of one or more of the second plurality of correlated nodes, representations of one or more of the second plurality of edges, and representations of one or more of the third plurality of edges, wherein one or more of the second plurality of edges together with one or more of the third plurality of edges represent one or more event flows that could be involved in a cybersecurity attack; and
[1.13]	generate a report comprising information associated with the one or more event flows.

A full claims listing is submitted as Exhibit 1010.

I. INTRODUCTION

Microsoft Corporation (“Petitioner”) respectfully requests *inter partes* review (“IPR”) of claims 1, 4-6, 9-11, 13-15, 17, 20, and 22 of U.S. Patent 12,301,628 (“’628 patent,” EX1001), allegedly assigned to QOMPLX LLC (“Patent Owner”). As explained below, these claims should be found unpatentable and cancelled.

The challenged claims relate to a computer system storing and updating a graph structure comprising nodes and edges that represent, respectively, entities in a computer network and their relationships. The computer system uses this graph to identify pluralities of correlated nodes and edges associated with an anomalous event and then generate a second graph reflecting the correlated nodes and edges.

Years before the challenged claims’ effective filing date, Brezinski (EX1004) disclosed the same concept of generating a network graph, modifying the graph to add nodes and edges based on new events, and then analyzing the graph to identify graph nodes and edges connected to an anomalous node. Brezinski describes detailed processes for “determining a graph that correlates a plurality of events detected on host devices, and for identifying anomalous activity by traversing the graph” (EX1004, 2:9-12), where “each event may correspond to an interaction between two entities in a computing environment” (*id.*, 2:17-19). *See also id.*, FIGS. 10-11.

Brezinski was not previously considered by the Office and discloses the same features relied on during prosecution to allegedly distinguish the challenged claims

from the prior art. The combination of Brezinski and Crabtree, which includes many disclosures identical to the '628 patent, renders each challenged claim obvious.

II. GROUNDS FOR STANDING PER 37 C.F.R. § 42.104(a)

Petitioner certifies that the '628 patent is available for IPR and that Petitioner is not barred or estopped from requesting IPR on the grounds presented herein.

III. IDENTIFICATION OF CHALLENGE

Petitioner requests review of claims 1, 4-6, 9-11, 13-15, 17, 20, and 22 (the “Challenged Claims”) of the '628 patent, on the following statutory ground:

	Reference(s)	Basis	Claims
Ground 1	Brezinski (EX1004) and Crabtree (EX1005)	§ 103	1, 4-6, 9-11, 13-15, 17, 20, and 22

The petition presents evidence of unpatentability and establishes a reasonable likelihood that Petitioner will prevail in establishing that each Challenged Claim is unpatentable.

IV. LEVEL OF SKILL IN THE ART, AND STATE OF THE ART

A. Person Of Ordinary Skill In The Art

The person of ordinary skill in the art in April 2021 (“POSITA”) would have had a bachelor’s degree in computer science, or equivalent, with two to four years’ experience designing, implementing, or otherwise working with cybersecurity technologies. More education, e.g., a master’s degree or Ph.D., would compensate

for less work experience and vice versa. EX1003, ¶36. The state of the art, immediately below, identifies certain technologies within the knowledge and skill of a POSITA. *Id.*

B. State Of The Art

Graph structures have long been used to represent and analyze data, including entities and their behaviors. *See* EX1008, 522. Such techniques were applied to computer networks more than a decade before the '628 patent's filing date. *Id.*, 521. These types of graphical network representations have been commonly used to identify anomalies in a computer network, such as by identifying deviations from normative behavior patterns. *See id.* (“In this work, we describe methods for graph-based anomaly detection”). *See also* EX1009, 673-74 (2014 survey of graph based methods for identifying “[a]nomalies in computer networks”). EX1003, ¶37.

V. THE '628 PATENT

The '628 patent issued May 13, 2025, from Application No. 18/892,295, filed September 20, 2024, which is a continuation of Application No. 17/237,346 (the “346 application”), filed April 22, 2021. The '628 patent alleges priority through a series of continuation-in-part applications to an application filed October 28, 2015.

The '628 patent describes a “method for identifying correlations between an anomaly and other network events and resources.” *Id.*, 21:9-11. It explains that “when an anomaly is identified 3010..., it may be correlated with other events,

network resources, behavior patterns, or any other available information that may be contextually linked to the anomaly.” *Id.*, 21:11-16. Further, “correlations may be identified by first comparing resources involved in the anomaly 3020, for example any machines that participated in the anomalous event or any databases or other resources that were accessed. This information may then be used to further correlate with other events involving those resources 3030.” *Id.*, 21:16-25.

A. The ’628 Patent Disclosures In Common With Crabtree

The ’628 patent and the Crabtree prior art (EX1005) share numerous disclosures. *See* EX1020 (redline comparison showing differences); EX1003, ¶52. Crabtree lists as inventors two of the ’628 patent’s three inventors and Crabtree issued from Application No. 15/823,363, which is an ancestor to the ’628 patent through several continuation-in-part applications (EX1001, 1:9-24).

The ’628 patent and Crabtree both disclose “mapping a cyber-physical system graph (CPG)” comprising “a visualization of hierarchies and relationships between devices and resources in a security infrastructure, contextualizing security information with physical device relationships that are easily understandable for security personnel and users.” *Id.*, 23:58-65, EX1005, [0076]. Both describe such a graph as a “representation of information and relationships, where each primary unit of information makes up a ‘node’ or ‘vertex’ of the graph and the relationship

between two nodes makes up an edge of the graph.” EX1001, 7:47-51; EX1005, [0044].

The ’628 patent and Crabtree disclose receiving time-series information and using it “to produce [] a graph visualization of users, servers, devices, and other resources correlating physical relationships ... with logical relationships (such as access privileges or database connections), to produce a meaningful and contextualized visualization of a security infrastructure that reflects the current state of the internal relationships present in the infrastructure.” EX1001, 24:8-16; EX1005, [0076]. Both also describe “updating CPG information as changes occur and events are logged.” EX1001, 24:5-8; EX1005, [0076]. Time-series data “may be received continuously ... for use in producing time-series graphs [] of states and changes over time”, thereby “providing live monitoring” of the state of the network. EX1001, 23:43-55; EX1005, [0075].

The ’628 patent and Crabtree disclose the graph including information for evaluating and identifying security risks in the network, such as “behavior analytics information” for recognizing deviations from established patterns of network behavior. EX1001, 22:46-23:16, 23:65-24:1; EX1005, [0072], [0076]. The graph may also contain “impact assessment scores” reflecting access capabilities of user accounts in a network, “representing the potential risk should that account be compromised.” EX1001, 23:17-30, 24:1-5; EX1005, [0074], [0076].

B. The Prosecution History

A first Office action issued November 15, 2024 (EX1002, 166-82/279), rejecting pending independent claims as obvious over Neil and Gilbert. *Id.*, 174/279. In response, the Applicant substantially amended the pending independent claims on February 15, 2025. *Id.*, 205-212/279. In doing so, the Applicant argued that the cited art failed to teach or disclose three newly-added limitations: [1] using a modified representation of a first graph to perform a “correlation” identifying a “first plurality of correlated nodes”; [2] using the modified graph representation to identifying a “second plurality of correlated nodes”; and [3] generating a “representation of a second graph” including the first and second pluralities of correlated nodes with edges representing “possible event flows of a cybersecurity attack.” *Id.*, 221-222/279. Thereafter, the claims were allowed, subject to an Examiner’s amendment (*id.*, 256-270/279), and the ’628 patent issued May 13, 2025 (EX1001, Cover).

C. The Challenged Claims’ Earliest Possible Effective Filing Date Is April 22, 2021

In its Complaint and preliminary infringement contentions in district court, Patent Owner has identified the claims’ priority date as April 22, 2021. EX1011, 9; EX1021, ¶160.

Patent Owner cannot show entitlement to an effective filing date before April 22, 2021, because the earlier applications do not provide written description

support for the challenged claims, including at least the following limitations, which were relied on in distinguishing the prior art during prosecution (*supra* Section V.B):

- “for an anomalous event associated with a node in the modified representation of the first graph, perform a first correlation...”;
- “for one or more of the first plurality of correlated nodes, perform a further correlation...”;
- “generate a representation of a second graph...”

When adding these limitations in prosecution, Applicant identified alleged support that consisted solely of paragraphs and figures first included in the '346 application and not found in any earlier application identified by the '628 patent. *Compare* EX1002, 214/279 (identifying “¶¶ [84]-[85] and FIGS. 30-31 of the as-filed application”), 55-56/279, 86-88/279 *with* EX1006, [0100]-[0101], FIGS. 30-31.

If Patent Owner contends otherwise, it “must demonstrate entitlement to a priority date” that would render Crabtree not prior art. *See Penumbra, Inc., v. Rapidpulses, Inc.*, IPR2021-01466, Paper 34 at 18 (PTAB Mar. 10, 2023) (Precedential as to § II.E.3) (noting “the patentee must demonstrate that the claims meet the requirements” of Section 120, including Section 112) (quoting *In re NTP, Inc.*, 654 F.3d 1268, 1276–77 (Fed. Cir. 2011)); *see also Dynamic Drinkware, LLC v. Nat’l Graphics, Inc.*, 800 F.3d 1375, 1379-80 (Fed. Cir. 2015) (discussing burdens in *inter partes* review to show entitlement to earlier filing dates).

VI. CLAIM CONSTRUCTION

Unless otherwise expressly discussed below, Petitioner applies the plain and ordinary meaning of all claim terms. EX1003, ¶44.

VII. GROUND 1: CLAIMS 1, 4-6, 9-11, 13-15, 17, 20, AND 22 ARE OBVIOUS OVER BREZINSKI (EX1004) AND CRABTREE (EX1005)

As explained below, claims 1, 4-6, 9-11, 13-15, 17, 20, and 22 are obvious over Brezinski and Crabtree.

A. Brezinski

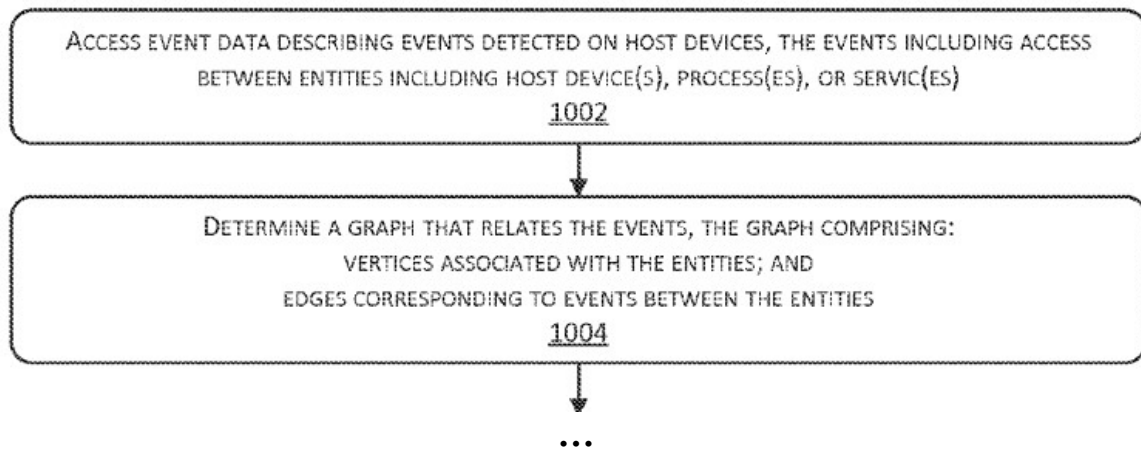
Brezinski (EX1004) issued December 29, 2015, from an application filed March 19, 2014, and is prior art under at least 35 U.S.C. § 102(a)(1). EX1004, Cover. Brezinski was not cited during prosecution and is not referenced by the '628 patent.

Brezinski describes processes “for generating a graph that correlates multiple events” between network entities (EX1004, 1:54-60) and “analyzing a graph to identify anomalous activity that includes chains of related events” (*id.*, 1:62-63). “By correlating event data in a graph, and traversing the graph according to risk and rarity of events, implementations may enable the detection of an attack.” *Id.*, 4:34-38.

The Brezinski environment includes a plurality of host computing devices and an analysis device. *Id.*, FIG. 1, 4:39-48. Using event data received over time, the analysis device “generate[s] a graph that interrelates or correlates a plurality of events.” *Id.*, 2:31-32; *id.*, 17:23-29. The event data describes network events, such

as connections between host devices, processes, services, and other entities. *Id.*, FIG. 1, 1:18-22; Abstract. The graph includes vertices (i.e., nodes) and edges, with “each vertex corresponding to an entity (e.g., a host device, process, or service)” and “each edge connecting two vertices to indicate an event.” EX1004, 2:45-53.¹

Figure 10 of Brezinski depicts “a process for generating a graph that correlates multiple events present in the event data” (*id.*, 1:54-56) as shown in this excerpt:



¹In the art, “node” and “vertex” were used interchangeably. *Infra* p. 20; EX1003, ¶67.

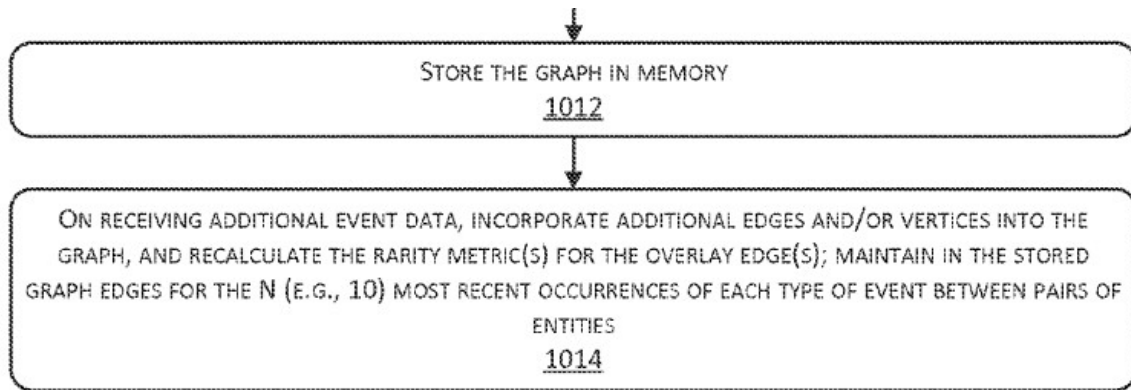


FIG. 10

Id., FIG. 10 (excerpted). As shown in Step 1014, “on receiving additional event data,” the graph is modified to incorporate additional edges and nodes corresponding to newly-detected events and entities. *Id.*, 17:23-29.

Brezinski Figure 4 shows an exemplary graph, with nodes/vertices 402(1-10) connected by directed edges 404(1) representing events, such as connections or interactions between the vertices. *Id.*, FIG. 4, 1:27-31, 8:29-62.

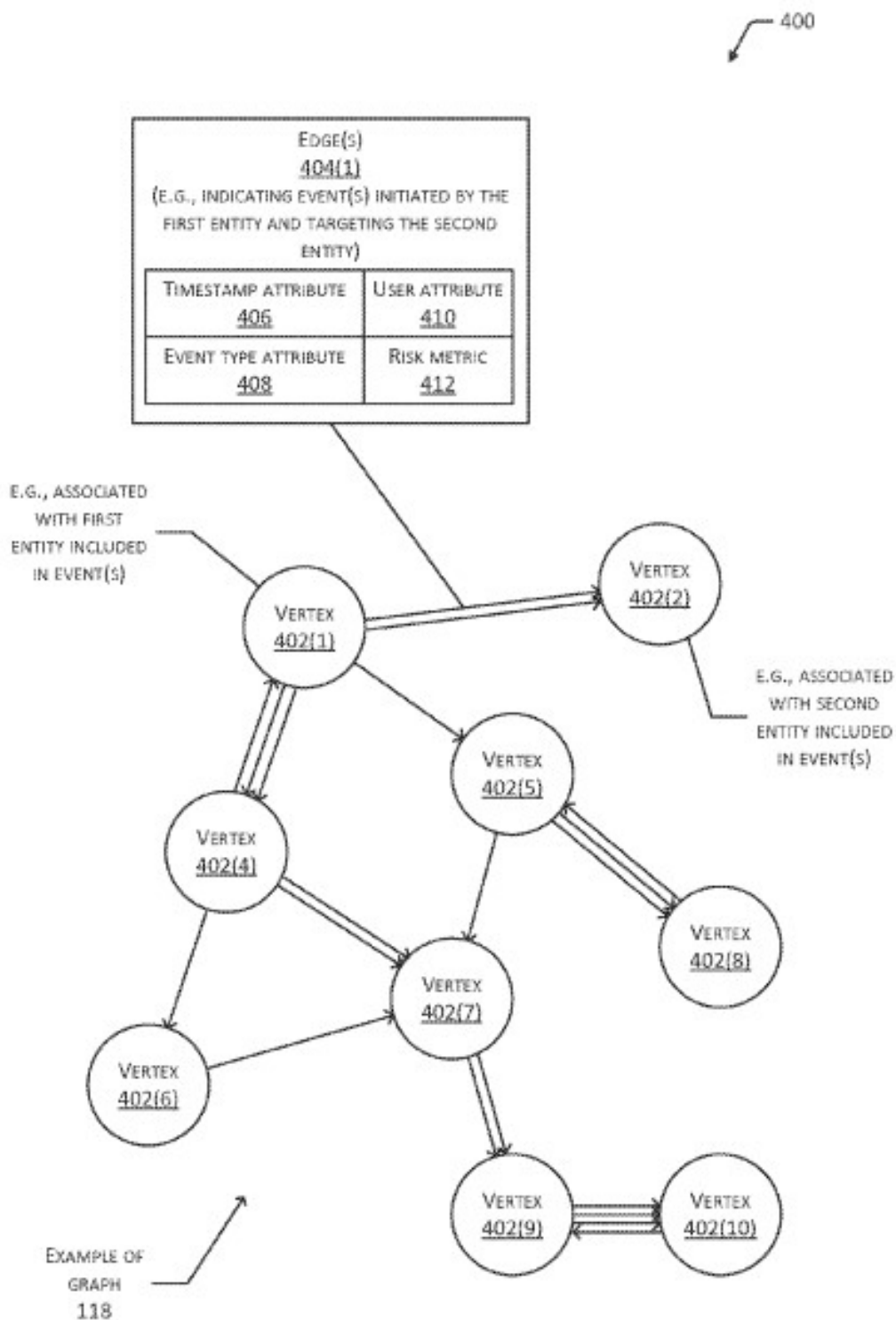


FIG. 4

Id., FIG. 4.

Brezinski discloses using its graph structure “to identify anomalous activity data.” EX1004, 3:52-58. Starting at one vertex/node, it traverses “at least a portion of the edges in the graph to identify those edges” that satisfy certain rarity and risk thresholds. *Id.* Brezinski also teaches identifying edges or vertices/nodes as anomalous if connected to potentially malicious entities, e.g., an edge that “connects a vertex 402 that is associated with a particular host device 102 or other entity” such as a designated honey pot. *Id.* 12:42-47; 12:35-38. Identified edges and vertices/nodes are stored in a subset graph as “anomalous activity data” for subsequent analysis. *Id.*, 11:39-43; 12:12-27; 18:29-38; FIG. 11.

B. Crabtree

Crabtree (EX1005) published August 2, 2018, from an application filed November 27, 2017 (EX1005, Cover) and is prior art under at least Section 102(a)(1) because Patent Owner cannot meet its burden in showing an effective filing date any earlier than April 22, 2021. *Supra* Section V.C.

Crabtree shares many of the same disclosures as the '628 patent. *Supra* Section V.A.

C. Combining Brezinski And Crabtree

As explained below, a POSITA would have been motivated to implement a Brezinski-like system—including its network graph generation and graph-based anomaly detection—with Crabtree’s teachings on receiving and analyzing streaming

data to build and modify network graphs and would have had a reasonable expectation of success in doing so.

A POSITA would have been motivated to combine Brezinski and Crabtree for several reasons. *First*, the references address closely related cybersecurity problems using similar graph-based analysis and provide complementary teachings. Brezinski describes processing event data from a networked computing environment, generating a graph with nodes/vertices representing entities and edges representing events/relationships, and traversing that graph to identify anomalous activity. *See supra* Section VII.A; EX1004, Abstract, 2:21-25, 2:45-48, 2:60-64, 3:32-33, 3:52-58, 8:29-37, FIGS. 4-7, 10-11. Crabtree likewise describes generating and updating a graph of node-represented entities and edge-represented relationships in a network environment for the purpose of identifying cybersecurity risks and anomalous behavior. *See supra* Section VII.B; EX1005, [0072]-[0076]; EX1003, ¶56. Thus, Crabtree's teachings are directed to the same type of graph-based cybersecurity system already disclosed in Brezinski.

Second, a POSITA implementing Brezinski would have recognized the benefit of Crabtree's teachings on receiving and processing event data in a streaming manner. Crabtree notes known benefits, such as allowing "live assessment of attacks as they occur" and "proactive and high-speed reactive defense capabilities against a variety of cyberattack threats." EX1005, [0073], [0079]. Background references

likewise reflect the known benefits of promptly detecting anomalous behavior indicative of cybersecurity attacks. See EX1015, [0003] (discussing the criticality of quickly identifying “anomalous data”); EX1003, ¶57. It was known that real-time streaming data analysis allowed for quicker anomaly identification. EX1015, [0003] (“beneficial” to capture and analyze “streaming data”); EX1016, [0004]-[0005] (“real-time” extraction provides “effective and timely detection of cyber threats”); EX1017, 2:39-41 (importance of “streaming data” when “finding relevant security events”); EX1003, ¶57. A POSITA thus would have been motivated to incorporate Crabtree’s live, continuous, streaming analysis teachings into Brezinski’s system to improve the timeliness and effectiveness of anomaly detection and response. EX1003, ¶57.

Third, Crabtree’s streaming and real-time analysis teachings are directly applicable to Brezinski’s graph-generation and graph-analysis processes. Both references model cybersecurity-relevant environments as graphs of nodes/entities and edges/relationships (*e.g.*, EX1004, Abstract, 2:21-25; EX1005, [0076]); both analyze those graphs to identify anomalous or attack-related behavior (*e.g.*, EX1004, Abstract, 3:52-53; EX1005, [0072]); and both are directed to helping security personnel understand and respond to suspicious network activity (*e.g.*, EX1004, 12:24-27; EX1005, [0074], [0076]). Crabtree’s streaming teachings thus would have

complemented Brezinski's graph-based anomaly-detection approach by improving the manner and timing of event-data receipt and analysis. EX1003, ¶58.

A POSITA would have had a reasonable expectation of success in combining Brezinski and Crabtree. The two references operate in the same technical field of graph-based cybersecurity analysis and rely on compatible inputs, structures, and objectives. *E.g.*, EX1004, Abstract; EX1005, [0072]-[0074], [0076]. Brezinski collects event data, generates a directed graph of entities and events, and analyzes that graph to identify anomalous activity. *See* EX1004, Abstract, 2:21-25, 2:45-48, 2:60-64, 3:32-33, 3:52-58, 8:29-37. Crabtree similarly uses continuously collected time-series event information to generate and analyze a graph of users, devices, servers, and other resources for cybersecurity monitoring and risk assessment. EX1005, [0075]-[0076]. A POSITA thus would have recognized that Crabtree's streaming techniques applied to, and were compatible with, the highly-similar graph-based security system of Brezinski. EX1003, ¶60. Incorporating Crabtree's teachings into Brezinski involved the predictable use of known techniques to improve event data ingestion and analysis, without requiring any fundamental change to Brezinski's architecture. *Id.*

Brezinski already contemplates ongoing graph modification "on receiving additional event data," including incorporation of additional edges and/or vertices and recalculation of rarity metrics using "recently detected events." *Id.*, 17:23-38.

And Crabtree requires no specific hardware, structure, or algorithm for its streaming data, explaining that such functionality is performed using modular components and “standardized protocols for converting streams of information into graph representations of that data.” EX1005, [0050]. These disclosures confirm that a POSITA possessed the skill and knowledge necessary to implement Crabtree’s streaming event-ingestion and graphing functionality in the Brezinski devices and environment, without technological incompatibility or undue experimentation. EX1003, ¶61. That expectation of success is confirmed by the common usage of streaming data analysis in similar cybersecurity systems. EX1015, [0003], [0015]; EX1016, [0004]-[0005]; EX1017, 3:20-38; EX1018, [0023], [0038]. EX1003, ¶61.

The resulting Brezinski-Crabtree system would include the computing environment of Brezinski, including its analysis and host devices, with the analysis device receiving and processing event data in a continuous or streaming manner as taught by Crabtree. EX1003, ¶62.

Further motivations to combine Brezinski and Crabtree are discussed for individual claim elements below.

D. Claim 1

1. Element [1.1]²

[1.1] A computer system comprising:

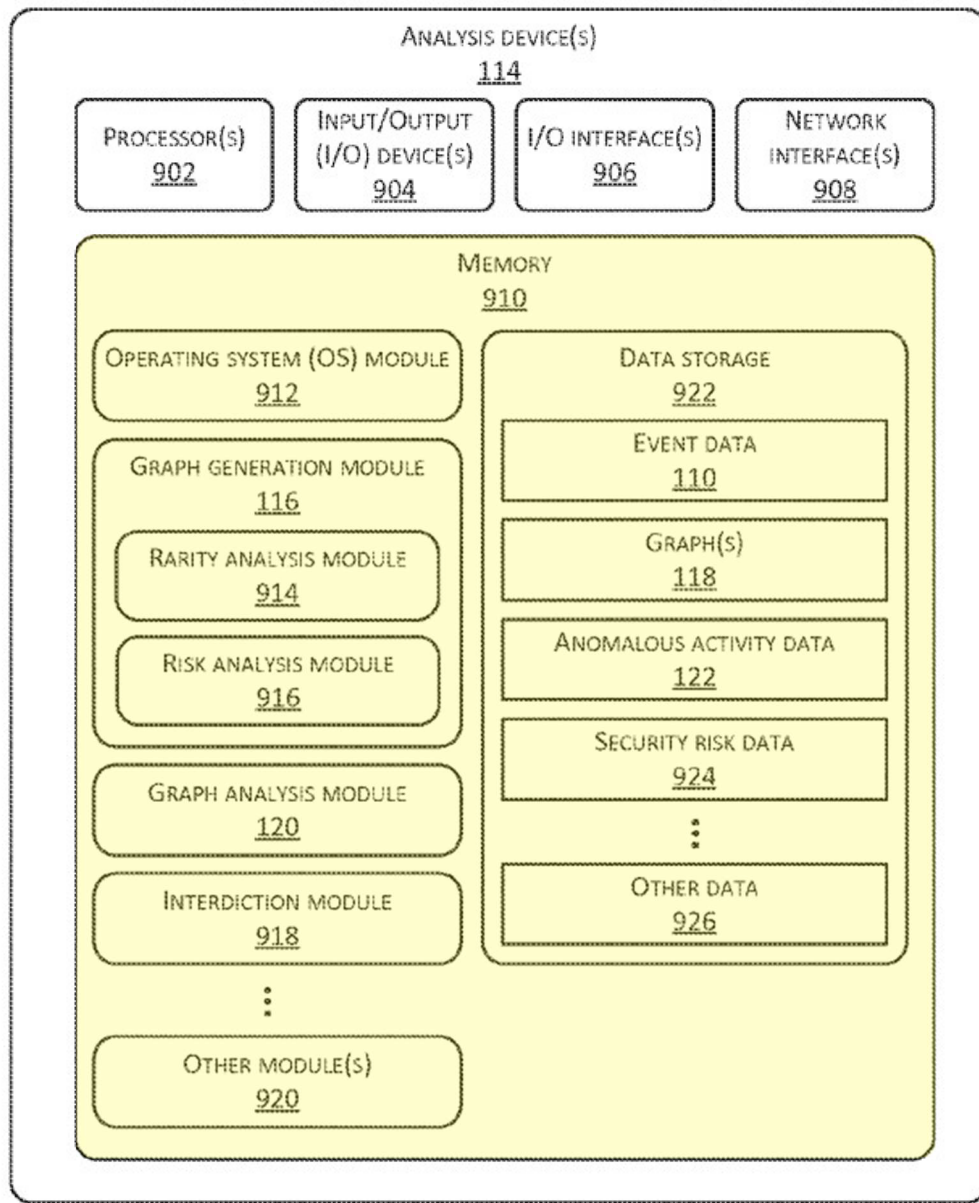
Brezinski discloses one or more “analysis devices” (*computer system*) that “may comprise any type of computing device.” *E.g.*, EX1004, 1:18-22, 4:40-41, 8:6-7, FIGS. 1, 9; *compare with* EX1001, 34:10-42 (describing “computer system”). EX1003, ¶64.

2. Element [1.2]

[1.2] a hardware memory, wherein the computer system is configured to execute software instructions stored on nontransitory machine-readable storage media comprising software instructions that:

Brezinski discloses that each analysis device includes a memory 910 and a processor 902 “configured to execute one or more stored instructions.” EX1004, 15:7-20; *id.*, 19:50-20:20 (describing “non-transitory computer-readable storage media having stored thereon instructions”). Brezinski FIG. 9 shows details of the analysis device(s), including memory 910 (highlighted) and processor 902:

² Reference numbers in the format of [claim#.limitation#] are added throughout.



Id., FIG. 9 (highlighting added); *see also id.*, 14:23-34, 15:19-21, 23:13-24-50 (CRM claims). It further discloses modules “executing on the analysis device(s) 114, such as the graph generation module 116 and the graph analysis module 120,” among other executable modules. *Id.*, FIG. 9, 15:28-57. Consistently, Brezinski describes performing its disclosed methods using “instructions” stored on “non-transitory

computer-readable storage media.” *Id.*, 19:51-60. The subsequent sections describe functions performed on Brezinski’s analysis device and a POSITA would therefore have understood such functions to be performed by analysis device’s software instructions for performing its functions. EX1003, ¶65.

3. Element [1.3]

[1.3] store in the hardware memory a representation of a first graph, wherein the representation of the first graph comprises representations of a first plurality of nodes corresponding to a first plurality of entities and further comprises representations of a first plurality of edges, wherein the first graph is a directed graph,

Brezinski’s analysis devices store graph(s) 118 in “memory 910,” as shown in Figure 9. EX1004, 15:58-59, 15:65-67, FIG. 9; *supra* Section VII.D.2. EX1003, ¶66. Exemplary FIG. 10 describes “a process for generating a graph 118” performed by “modules executing on the analysis device(s)” (EX1004, 16:15-21) that includes “stor[ing] the graph in memory” (*id.*, 17:15-22; FIG. 10 (Step 1012)). *See generally, id.*, 16:15-17:38 (process for generating, storing, and modifying graph). EX1003, ¶66.

The stored graph includes plural “vertices associated with the entities” (exemplary *representations of a first plurality of nodes corresponding to a first plurality of entities*) and “edges corresponding to events between the entities” (exemplary *representations of a first plurality of edges*). *Id.*, FIG. 10 (Steps 1004, 1012), 16:22-36. *See also id.*, 2:21-25, 2:31-48, 2:60-64, 3:32-33, Abstract.

Brezinski describes a “node” as an exemplary “vertex” (*e.g.*, EX1004, 2:35-36) and the '628 patent recognized that “nodes” were “also called vertices” in the relevant art (*e.g.*, EX1001, 28:17-21; *id.*, 7:47-49 (“‘node’ or ‘vertex’ of the graph”)). *See also id.*, 7:61-65, 28:36-42. Thus, a POSITA would understand any Brezinski disclosure of vertex/vertices as a disclosure of node/nodes. EX1003, ¶67.

Brezinski FIG. 4 shows an exemplary graph with a first plurality of nodes (vertices 402) connected by a first plurality of edges (edges 404):

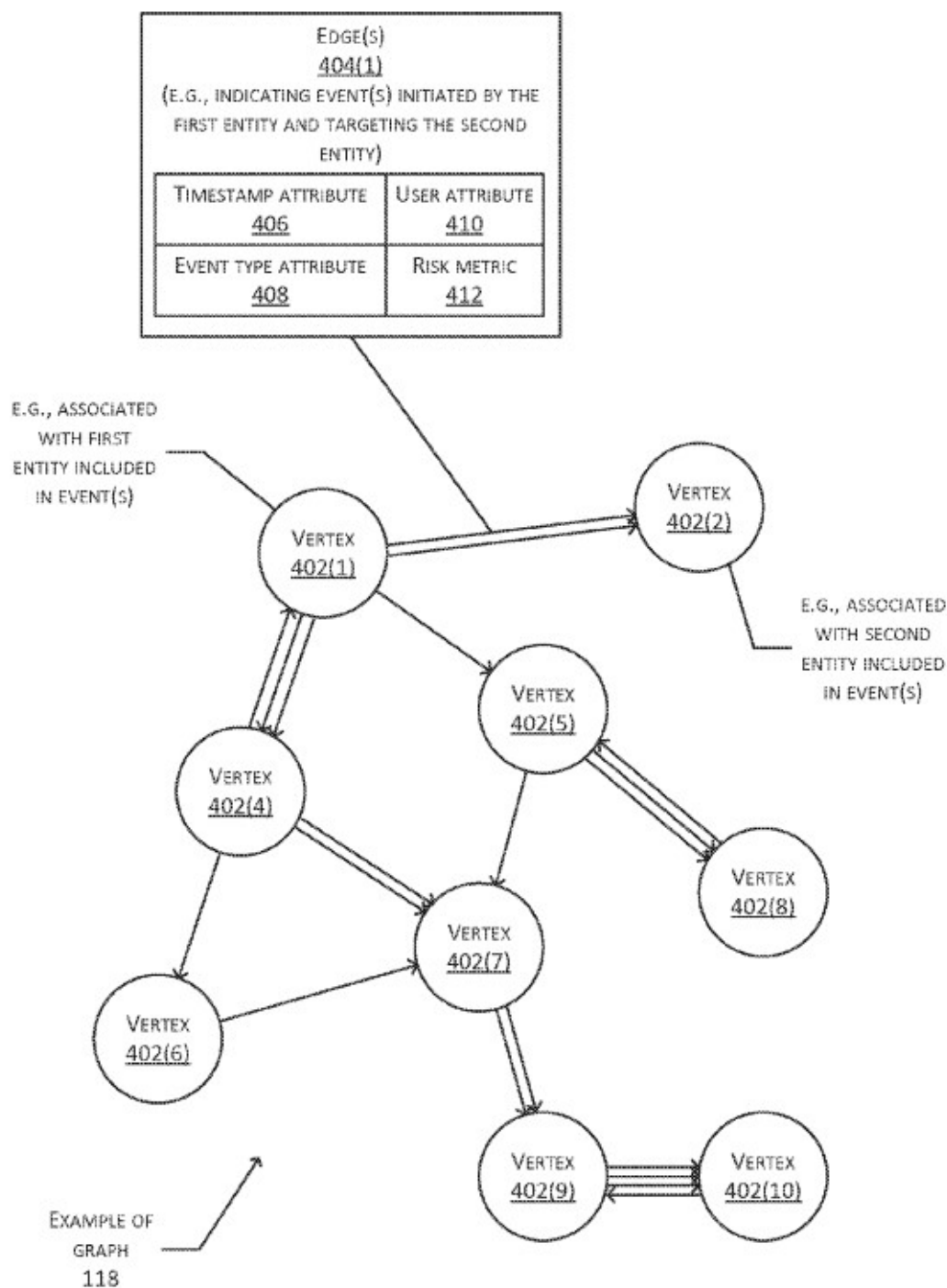


FIG. 4

Id., FIG. 4, 8:29-37; EX1003, ¶68.

Brezinski teaches that “graph 118 may be a directed graph” and Figures 4-7 each depicts a directed graph, with edges including an arrow to indicate direction. *See* EX1004, 8:53-56; *see also id.*, 2:42-44. EX1003, ¶69.

4. Element [1.4]

[1.4] wherein the first plurality of entities comprises a plurality of accounts and a plurality of resources,

Brezinski teaches this element. Brezinski discloses a graph with nodes representing entities such as host devices, processes, services, and files. EX1004, 2:45-51. Each such entity is a resource consistent with the '628 patent, which describes “servers, devices, and other resources” including “a user’s personal computer or a smartphone.” EX1001, 24:9-12, *see also id.*, 21:16-20. Brezinski therefore teaches graphs wherein *the first plurality of entities comprises a plurality of resources*. EX1003, ¶70.

Brezinski also teaches nodes for “other types of entities, such as **users**.” EX1004, 2:49-51 (emphasis added). A POSITA would have understood each Brezinski “user” to be represented by a user account because that is how users were represented in computer networks (like Brezinski’s). EX1003, ¶71; EX1014, 14 (computer dictionary defining “account” as “[t]he record-keeping mechanism used by networks... for keeping track of authorized users” and “to validate users”). This is consistent with Brezinski’s description of a “user ID that identifies one or more

particular users” or “user credentials.” EX1004, 7:41-45, 16:59-60 (“users or user groups identified by the user ID(s)”), 9:39-41, 10:31. At the very least, user accounts would have been the most obvious design choice for tracking user entities in the Brezinski system. EX1003, ¶71. The ’628 patent confirms this by repeatedly describing an “account” as a “user account” (*e.g.*, EX1001, FIG. 9 (steps 901-902), 23:16-21, 24:29-33 , 24:36-38) and describing “users” as entities represented by the graph (*e.g.*, EX1001, 27:27-29 (“entities ... for example, devices, users, ...”), 28:8-13 (same), 19:48-50, 24:8-10). Indeed, the ’628 patent’s sole figure showing a graph depicts nodes corresponding to “users.” *Id.*, FIG. 22, 28:37-40 (“[T]he cyber-physical graph 2200 contains 12 nodes (vertices) comprising: ... two users designated by dashed-line circles.”). EX1003, ¶71.

Crabtree’s Teachings on User Accounts and Resources

Alternatively, if Patent Owner argues that Brezinski’s teachings on users/credentials are insufficient, Crabtree teaches a graph that includes “the internal risk level of network resources, **user accounts**, and devices.” EX1005, [0077] (emphasis added). Crabtree teaches using time series data “**for user accounts**” to produce “a graph visualization of users, servers, devices, and other resources,” thereby including users as graph entities alongside other network resources. EX1005, [0075]-[0076], [0078] (emphasis added). Accordingly, Crabtree teaches a

graph whose nodes represent entities comprising a plurality of accounts and resources. EX1003, ¶72.

A POSITA implementing Brezinski's graph would have been motivated to include nodes representing user accounts along with nodes representing resources because Brezinski already discloses including nodes representing "users" (EX1004, 2:49-50) and Crabtree demonstrates that a "user account[]" is an obvious way of representing a user in a graph-based cybersecurity analysis. EX1005, [0077]. A POSITA would include in Brezinski's graph both account nodes and resource nodes in order to provide a holistic view of the entities acting within a monitored computer network to maximize coverage of the anomaly detection method of Brezinski. EX1003, ¶73. Including user accounts as an entity type would beneficially facilitate identification of anomalous activity associated with specific user accounts, thereby improving Brezinski's anomalous activity identification. EX1005, [0077]. Crabtree emphasizes this benefit by noting the need to improve security for "high risk accounts." EX1005, [0074]. A POSITA would therefore understand the importance of tracking user accounts as entities in the Brezinski graph, especially for identifying and promptly responding to anomalous events involving high risk accounts. EX1003, ¶73.

Moreover, a POSITA was familiar with presenting heterogeneous entity types (e.g., both users and resources) as nodes in a single graph as a POSITA recognized

that this provided a holistic view of all the actors and assets within a network and their interactions. EX1003, ¶74, EX1022, FIG. 3, [0045] [0056] (describing FIG. 3 as a network topology graph including nodes and edges, with nodes corresponding to different entities, including computers, users, files, and keys providing a “holistic view of network and traffic patterns”). EX1003, ¶74.

Further, a POSITA would understand it to be an obvious use for the software instructions of Brezinski-Crabtree to store a graph having nodes corresponding to both a *plurality* of accounts and a *plurality* of resources. Brezinski states that its graph “may include any number of vertices.” EX1004, 2:37-39. And a POSITA recognized that computer networks included many entities accessing and operating on the network. EX1003, ¶75; EX1004, 1:6-8 (“Providers of online services or other computing services may deploy a large number of computing devices to execute processes and services within a computing environment.”); EX1009, 673 (recognizing that a network graph of a large network includes a “large number of compute nodes” and corresponding “large number of edges” between such nodes); EX1018, [0004] (recognizing that a network’s “computer connection history can be large in scale, with hundreds of thousands computers in datasets each generating an average of 200-300 connection events in a typical day”), EX1022, [0056], FIG. 3 (depicting network graph including pluralities of different entity types). Accordingly, a POSITA would recognize that a graph reflecting a network with

many different nodes would therefore commonly contain at least two (a plurality of) nodes representing resources and at least two (a plurality of) nodes representing accounts. EX1003, ¶75.

5. Element [1.5]

[1.5] wherein each edge of the first plurality of edges corresponds to a respective relationship between a respective pair of entities of the first plurality of entities;

Brezinski explains that its graph includes “edge(s) 404(1) that each correspond to a particular event” between the two nodes/vertices connected by the edge. EX1004, 8:37-45. Each edge connects two nodes and reflects a relationship between the pair of entities represented by the nodes, such as “a network connection,” “a communication over a network,” “a login,” or an “access to a resource” or “access . . . to a process.” *Id.*, 6:45-46, 6:48-49, 6:65-66, 7:4-5, 7:21-22; *see also id.*, 32-34 (“A graph is a representation of a plurality of objects and one or more relationships between one or more pairs of the objects.”). As shown in Brezinski FIG. 4 (below), the graph has many pairs of entity-representing nodes (402(1-10)) connected by edges (shown as arrows) with each edge representing a relationship as indicated by an “event type attribute 408.” *Id.*, 8:63-9:3; *see also id.*, 5:20-23, 5:66-6:1, 6:40-44.

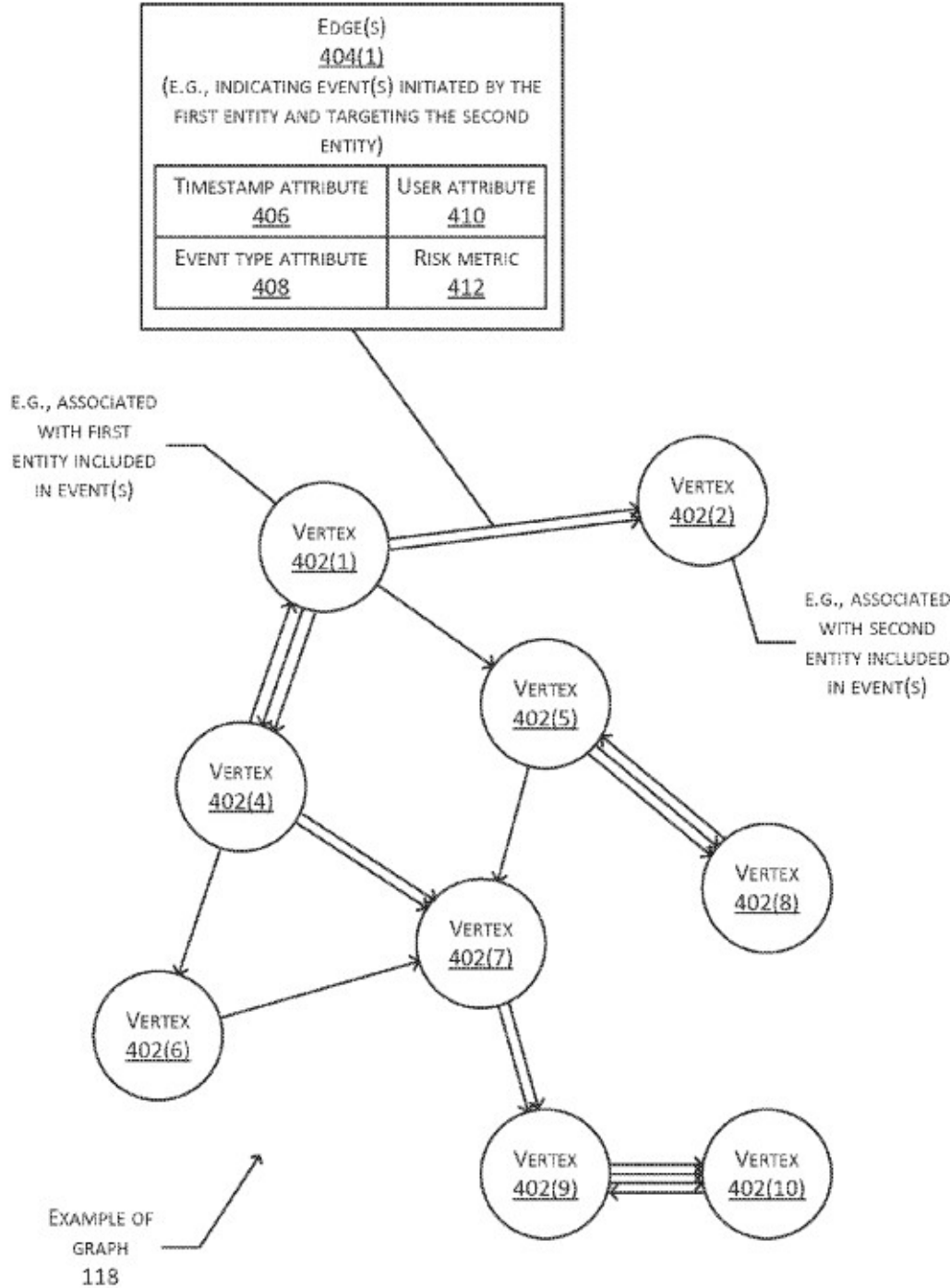


FIG. 4

Id., FIG. 4, 8:43-48; *see also* EX1004, 20:49-52 (claim 1). Thus, Brezinski's graph includes a plurality of edges, and each edge represents a specific event/interaction—i.e., a relationship—between the pair of entities whose nodes are connected by the edge. EX1003, ¶¶76-77.

6. Element [1.6]

[1.6] receive streaming data comprising time-stamped data about events relating to one or more entities of the first plurality of entities;

Brezinski discloses its analysis device(s) 114 receiving a flow of event data from host devices 102, as shown in the following annotated Figure 1:

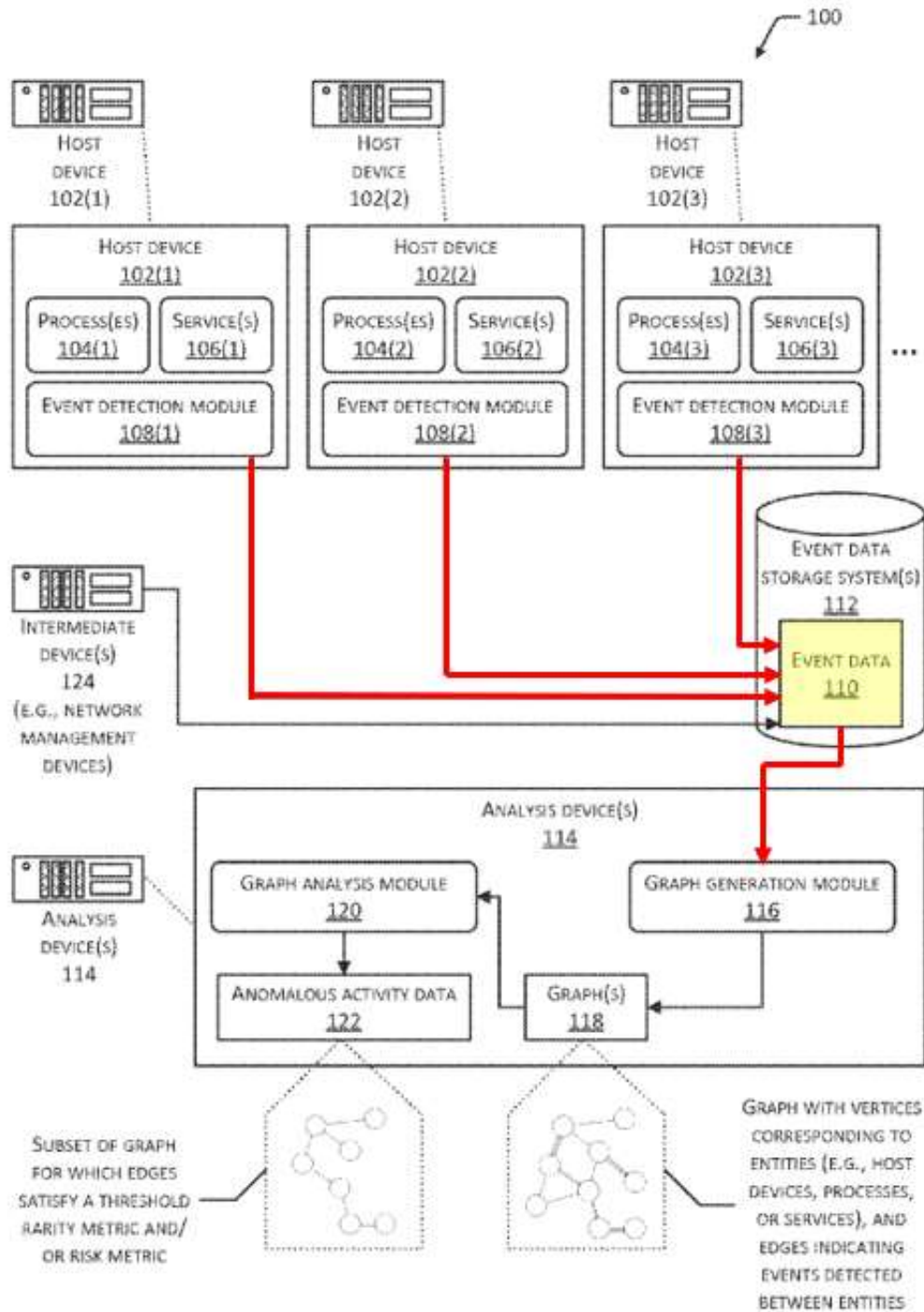
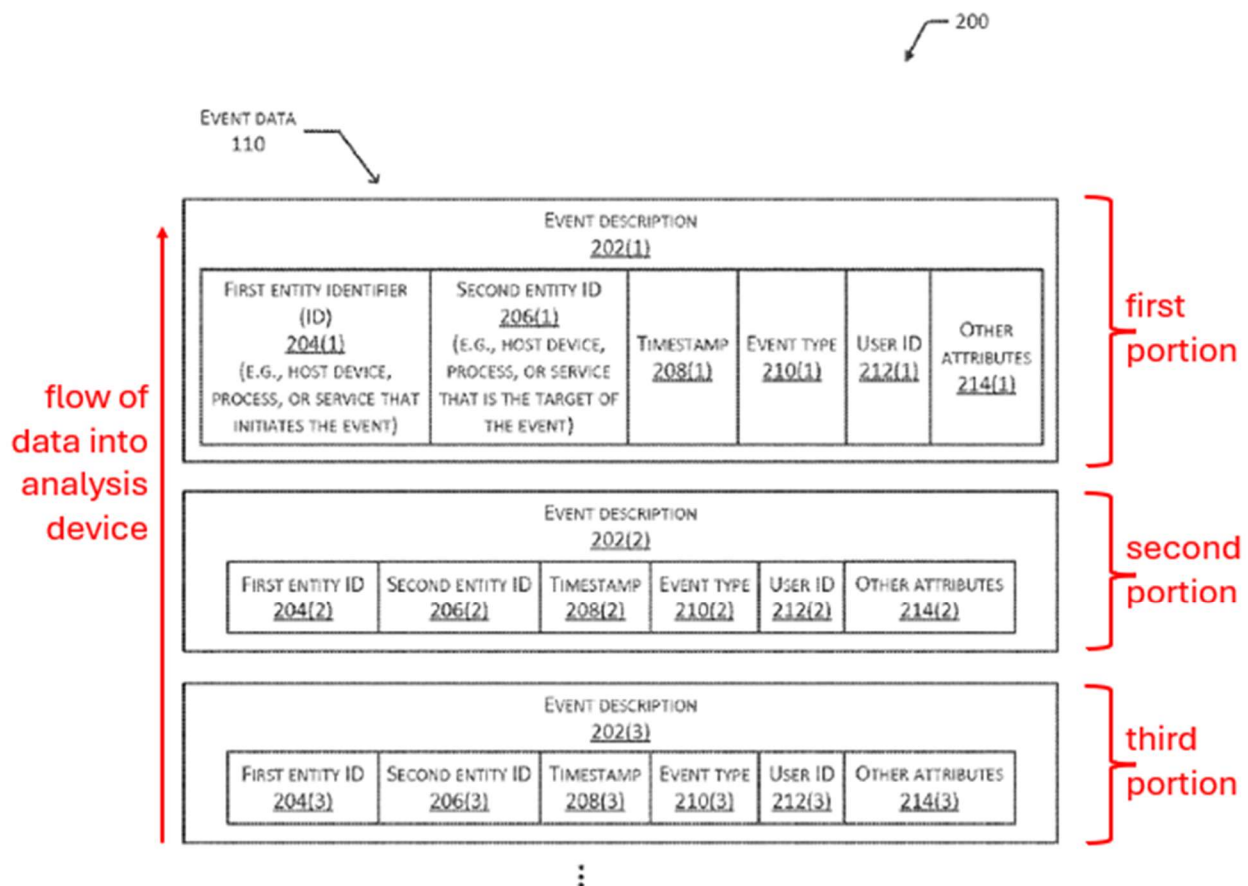


FIG. 1

Id., FIG. 1 (annotated); EX1004, 5:45-48, 8:23-27, 17:23-26. Brezinski also teaches implementing event data storage 112 within analysis device 114. *Id.*, 12:63-66; EX1003, ¶78.

Brezinski discloses exemplary event data as an ongoing series of event descriptions 202 that each includes a “first entity ID 204,” “second entity ID 206,” “timestamp 208, and “event type 210” (*time-stamped data about events relating to one or more entities*). *Id.*, 5:65-6:10, 6:37-44. Brezinski FIG. 2 shows example event descriptions 202(1-3), with an ellipsis at bottom indicating that the flow of event descriptions continues beyond those depicted:



EX1004, FIG. 2 (annotated); *see also id.*, 5:20-23 (describing “event data 110 describing one or more events that occur on the host device(s) 102”); EX1003, ¶79. Brezinski also contemplates the obvious use case of this flow including event descriptions that relate to entities already represented as nodes in the graph (*events relating to one or more entities of the first plurality of entities*). For example, Brezinski describes “additional event data” that leads to incorporating “additional edges” into the graph (EX1004, 17:23-26)—a POSITA would understand that these additional edges are between pre-existing nodes that correspond to known entities, as the description makes no mention of new nodes/entities (EX1003, ¶80). Brezinski reinforces this understanding by going on to describe “cases where the recently detected events involve entities not already described by vertices 402 in the graph.” EX1004, 6:26-29. Since some “cases” of additional events involve new entities, a POSITA would understand that other cases would involve entities already represented in the graph. EX1003, ¶80.

Crabtree’s Teachings on “Receiving Streaming Data”

While Brezinski describes the flow of data to the analysis device, it does not expressly disclose streaming event data. It was obvious, however, to implement a Brezinski-like system with an analysis device that processed streaming data, as taught by Crabtree. *Supra* Section VII.C. Crabtree expressly teaches converting “one or more streams of data from a plurality of sources” “into graph representations of

that data” using standardized protocols. EX1005, [0050]. Crabtree further describes “live monitoring” of a network by “continuously” collecting and receiving time-series data (*id.*, [0075]) and updating graph information “as changes occur and events are logged” (*id.*, [0076]), reflecting an ongoing intake of streaming event data. *E.g.*, EX1014, 499 (defining data “stream” as “data transmission... that occurs in a continuous flow”); *see also* EX1003, ¶81.

In the Brezinski-Crabtree combined system, streaming time-stamped event data is received and processed by the analysis device(s). *Supra* Section VII.C; EX1003, ¶82. With reference to Brezinski’s Figure 2, the combined system would receive streams of event data 110 continuously, with successive event descriptions (e.g., 202(1), 202(2), etc.) received as the corresponding events occurred in the network. EX1004, FIG. 2, 5:64-6:1. This would provide “live monitoring” (EX1005, [0075]) of system events and associated graph modification (EX1004, 17:23-29), thus allowing rapid analysis and reaction. EX1003, ¶82.

The ’628 prosecution history confirms that this satisfies the claim element because the Crabtree disclosures match the ’628 patent disclosures alleged by Applicant to support element 1.6. EX1002, 214/279 (identifying application paragraphs 93-94 as support); *id.*, 91/279 (paragraphs 93-94, referencing FIG. 10 and associated description); *id.*, 89-90/279 (paragraph 90, describing FIG. 10);

EX1005, [0075]; EX1020 (redline comparison of Crabtree and '628 patent specifications). EX1003, ¶83.

7. **Element [1.7]**

[1.7] based on a first portion of the streaming data, identify a first entity that does not correspond to any of the first plurality of nodes, wherein the first entity is not of the first plurality of entities;

The Brezinski-Crabtree analysis device receives streaming event data comprising an ongoing series of event descriptions (each a *portion of the streaming data*). *Supra* Section VII.D.6; EX1003, ¶84.

After storing the initial graph in memory (*supra* Section VII.D.3), the Brezinski analysis device uses “additional event data” to identify “entities not already described by vertices 402 in the graph 118” (*a first entity that does not correspond to any of the first plurality of nodes*). EX1004, 17:23-29; *see also id.*, 6:1-8 (describing ID “identifying the entity”), FIG. 2 (showing entity identifiers/IDs in the event description data) EX1003, ¶¶85-86. As Brezinski teaches event data comprising event descriptions (*supra* Section VII.D.6), a POSITA would understand that this new entity identification is based on an event description (*first portion of the streaming data*). EX1003, ¶¶85-86. And since the new entity was “not already described” in the graph (EX1004, 17:23-29), it is *not of the first plurality of entities*, which comprises, e.g., those entities represented in the graph before arrival of the

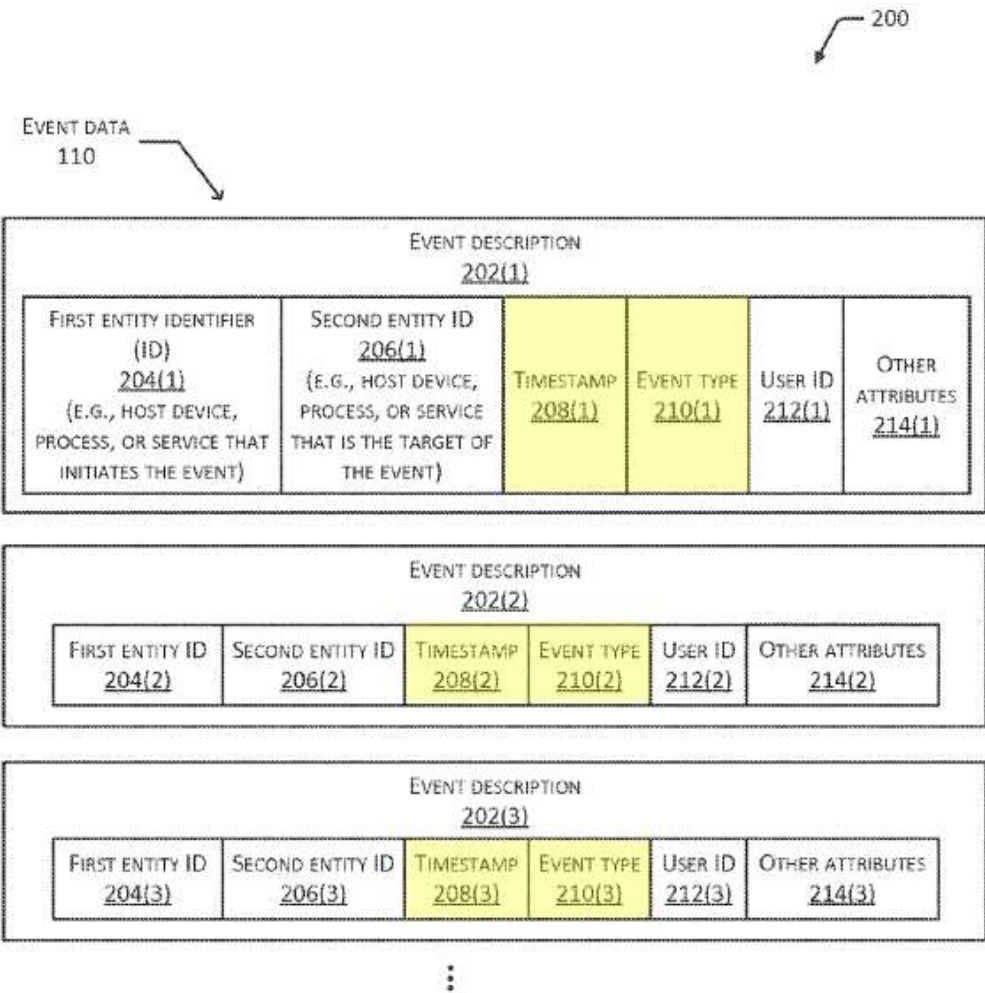
additional event data. EX1003, ¶86; *infra* Section VII.D.8 (explaining first and second portions).

8. Element [1.8]

[1.8] based on a second portion of the streaming data, wherein the second portion is not identical to the first portion, identify a first relationship between a pair of entities of the first plurality of entities that does not correspond to any of the first plurality of edges;

After storing the graph in memory (*supra* Section VII.D.3), the Brezinski analysis device uses “additional event data ... to incorporate one or more additional edges 404 corresponding to recently detected events.” EX1004, 17:23-26. Such “additional edges” are by their name not pre-existing edges and a POSITA would understand that their creation happens when the analysis device identifies a new event in the additional event data (*identify a first relationship that does not correspond to any of the first plurality of edges*). EX1003, ¶88. And because Brezinski describes incorporating additional edges without mention of adding new nodes (EX1004, 17:23-26), a POSITA would understand that these additional edges reflect a relationship between existing nodes-entities (*a first relationship between a pair of entities of the first plurality of entities*). EX1003, ¶88. In any event, such a scenario would be an obvious use case given Brezinski’s teachings on processing event data and additional event data. EX1003, ¶88; EX1004, FIG. 4, 8:43-52 (“multiple edges [] connecting the pair of vertices”).

As explained, the Brezinski-Crabtree analysis device receives streaming event data comprising an ongoing series of event descriptions (each a *portion of the streaming data*). *Supra* Section VII.D.6; EX1003, ¶89. Brezinski’s event descriptions identify the relationships between a pair of entities to be included in the graph because it includes the data used to define the edges of its cybersecurity graph, including an event type and a timestamp. EX1004, 2:53-62, 6:37-44; 10:10-20 (describing Example 1). Brezinski’s Figure 2, reproduced below, shows the event description data entity type and timestamp used for defining an edge.



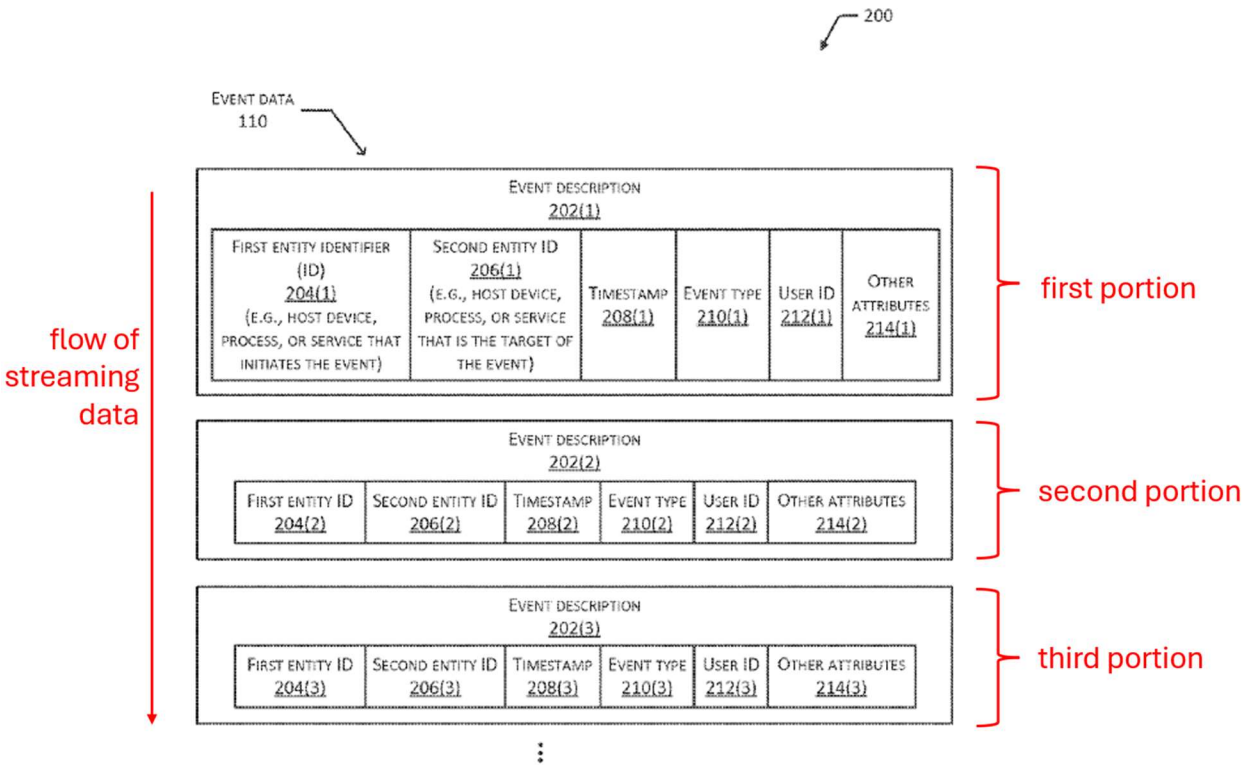
EX1004, FIG. 2. ¶89.

“a second portion of the streaming data ... not identical to the first portion”

The Brezinski-Crabtree analysis device receives streaming event data, one event description after another. *Supra* Section VII.D.6. And as explained in this section and the preceding section, Brezinski gives two exemplary types of “additional event data” that lead to different graph modifications: one type resulting in additional edges (EX1004, 17:23-26) and the other in additional nodes/vertices (*id.*, 17:26-29). Brezinski thus teaches receiving non-identical event descriptions within its flow of data. EX1003, ¶90. As such, in Brezinski-Crabtree the portion of streaming data that leads to new edges is different from the portion that leads to new nodes (*a second portion of the streaming data, wherein the second portion is not identical to the first portion*). *Id.*

And the POSITA would also recognize the obvious use case of a later event description for a new event between original graph node-entities (*second portion of streaming data*) arriving after an earlier event description for a new event with new node(s) (*first portion of streaming data*). *Id.* For example, with reference to FIG. 2, below, in the streaming environment of Brezinski-Crabtree, a POSITA would recognize the case of a new entity/node being identified based on event description 202(1) received at a first time (i.e., a first portion of streaming data), and then a new

relationship/edge being identified based on later received event description 202(2) (i.e., a second, different portion of received streaming data). EX1003, ¶91.



EX1004, FIG. 2 (annotated); EX1003, ¶91. Brezinski’s “additional event data” and “recently detected events” terminology reflects this temporal sequencing: the system may first identify a previously unseen entity from earlier-received event data, and later (based on subsequently received event data) identify a new relationship/event (between existing entities) that is not yet represented by an edge, prompting addition of an “additional edge[.]” EX1004, 17:23-29. EX1003, ¶91.

9. Element [1.9]

[1.9] modify, in the hardware memory, the representation of the first graph to generate a modified representation of the first graph, wherein the modified

representation of the first graph comprises a representation of a first node corresponding to the first entity and a representation of a first edge corresponding to the first relationship, wherein the first node is not of the first plurality of nodes and the first edge is not of the first plurality of edges;

Brezinski discloses incrementally updating its stored graph with newly identified nodes and newly identified edges when new event data is received. *E.g.*, *supra* Sections VII.D.7-8. “[O]n receiving additional event data” the analysis device modifies the graph, e.g., to add new edges for newly detected relationships/events and new vertices for newly detected entities. EX1004, 17:23-29; *see also id.*, FIG. 10 (Step 1014). EX1003, ¶92.

Accordingly, upon identifying a new first entity (*supra* Section VII.D.7) and a new first relationship (*supra* Section VII.D.8), the Brezinski-Crabtree analysis device would modify the stored graph to “incorporate” a first node corresponding to the new first entity and a first edge corresponding to the new first event/relationship, thereby *generating a modified representation of the first graph* with a first node and first edge as recited. EX1004, 17:23-29, EX1003, ¶¶93-94. As the newly identified first entity does not correspond to any of the pre-existing first plurality of nodes (*supra* Section VII.D.7), its corresponding newly added *first node is not of the first plurality of nodes*. Likewise, as the newly identified first event/relationship does not correspond to any of the pre-existing first plurality of edges (*supra* Section VII.D.8),

its corresponding newly added *first edge is not of the first plurality of edges*.
EX1003, ¶¶93-94.

10. Element [1.10]

a) [1.10.a]

[1.10.a] for an anomalous event associated with a node in the modified representation of the first graph, perform a first correlation using the modified representation of the first graph to identify a first plurality of correlated nodes,

As explained more fully below, Brezinski teaches performing a first correlation using a modified representation of the first graph discussed in element 1.9 in two different ways. EX1003, ¶95.

First, Brezinski teaches to “traverse a graph” along edges and identify “the edges 404 and the vertices 402” associated with a rarity and/or risk metric that satisfies a certain threshold (*perform a correlation ... to identify a first plurality of correlated nodes*), for inclusion “in a subset of the graph 118 designated as anomalous activity data 122.” *E.g.*, EX1004, 11:12-43, FIGS. 6-7. “The graph may be traversed according to the risk and rarity metrics, to identify patterns of anomalous activity in the event data.” *Id.* Abstract; *see also id.*, 3:52-53. The traversal may also be “breadth-first.” *Id.*, 3:60-63. This breadth-first graph traversal performs the claimed correlation because it uses the graph to correlate a plurality of nodes with anomalous network activity. *E.g.*, 3:52-67, 11:12-29, 12:12-14, 17:39-49, 18:22-44; 19:4-22; EX1003, ¶96. Brezinski teaches identifying these anomalous

nodes by “analyzing a graph to identify anomalous activity that includes chains of related events” (*e.g.*, 17:39-43) and thus the correlation is *for an anomalous event associated with a node in the graph*, as recited in element 1.10(a). EX1003, ¶96. For example, the traversal process identifies nodes connected to edges that satisfy a “risk threshold” (*e.g.*, *id.*, 18:23-25) and—because edges represent events—these edges indicate an anomalous event associated with a node in the graph. EX1003, ¶96.

Traversal of Modified Graphs

Brezinski further teaches performing this correlation *using the modified representation of the first graph*. Brezinski teaches its analysis device performing graph traversal on “graph(s) 118 ... stored in memory” (EX1004, 11:12-18), which includes modified graphs because “the graph 118 may be modified to incorporate one or more additional edges 404 [or] vertices 402” (*id.*, 17:23-29; FIG. 10 (Step 1014)). *See also id.*, 8:23-27 (“graph generation module”), 16:15-21 (“process for generating a graph 118”). Confirming this, the first step 1102 of Brezinski’s graph traversal process is broadly “a graph 118 may be accessed.” *Id.*, 17:39-49; FIG. 10 (Step 1102 (“access a graph relating events detected on host devices”). Thus, Brezinski teaches identifying anomalous activity by traversing a modified graph such as that described for Element [1.9] (*supra* Section VII.D.9). *Id.*, 17:39-49; EX1003, ¶97. A POSITA would recognize the obvious use case of performing such a traversal on a modified representation of a first graph described, *e.g.*, to identify

anomalous activity caused by newly identified entities and relationships/events.
EX1003, ¶¶97-98.

Figure 7 of Brezinski shows an example of such a traversal procedure. Figure 7 depicts a subset graph resulting from the traversal of example graph of Figure 6 to identify “anomalous activity data 122” by determining edges “for which the rarity metric 502 satisfies and is at least a rarity threshold of 0.8, and for which the risk metric 412 satisfies a risk threshold of ‘high’” and vertexes connected by those edges. EX1004, 11:44-47, 11:65-12:4, FIGS. 6 & 7. Figure 7 of Brezinski therefore constitutes a subset graph of the example graph of Figure 6, including only those edges that were determined by the traverse procedure to satisfy risk and rarity metric thresholds. *Id.* Figure 7 is reproduced below and annotated to show a first plurality of correlated nodes identified by Brezinski’s graph traversal procedure.

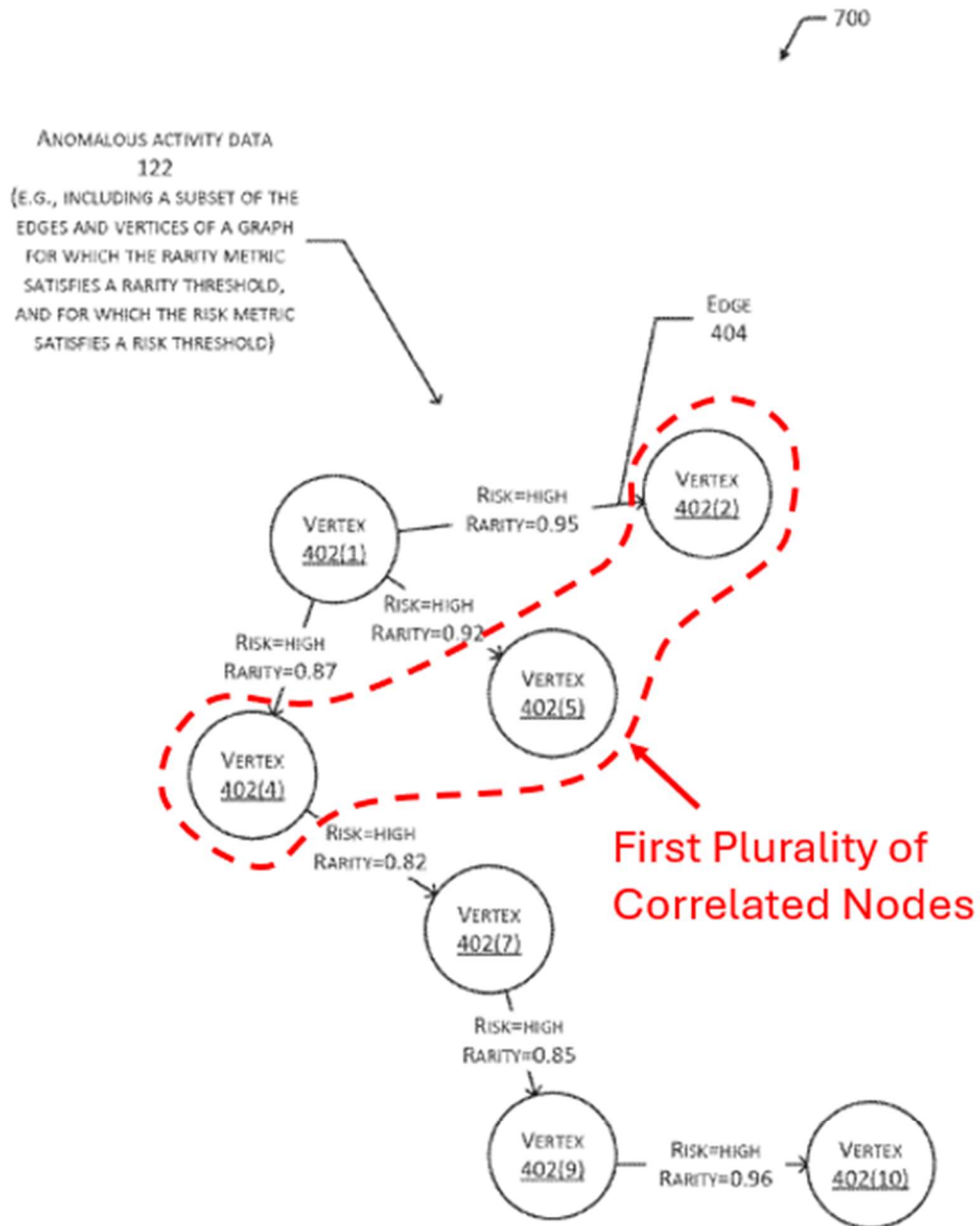


FIG. 7

EX1004, FIG. 7 (annotations in red). EX1003, ¶99. Brezinski specifically discloses that its traversal of the graph (such as that shown in FIG. 6) “may be breadth-first

such that neighboring nodes are explored prior to exploring deeper along a possible branch.” *Id.*, 3:60-63. Such a “breadth-first” search methodology is iterative, identifying neighboring nodes level-by-level from a starting node. In the example of FIGS. 6 & 7, a POSITA would understand that performing a breadth-first traversal of the graph in FIG. 6 from the vertex labeled 402(1) would identify in a first level of a breadth-first search the vertices labeled 402(2), 402(4), and 402(5) (and the edges connecting them to vertex 402(1)) as anomalous activity data in a first level of the breadth first search. EX1003, ¶99. As shown in Figure 7, Brezinski’s breadth-first traversal (EX1004, 3:60-63) identifies each of the vertices labeled as 402(2), 402(4), and 402(5) (and the edges connecting them to vertex 402(1)) as directly connected by an edge to vertex 402(1) (i.e., a node associated with an anomalous event) and identified those nodes as anomalous data for inclusion in the subset graph of Figure 7. EX1004, 11:65-12:4, EX1003, ¶99. Brezinski’s breadth-first graph traversal therefore includes a first “correlation” because it uses the structure of the graph—relationships captured by edges between vertices—to identify other graph elements connected to (i.e. correlated with) an anomalous node. EX1003, ¶99.

Second, Brezinski teaches designating an entity (such as a honey pot) as associated with anomalous behavior and correlating the node corresponding to that entity with any nodes connected to it. *See id.*, 12:35-53. EX1003, ¶100. Anomalous activity is detected and correlated by starting from a node/vertex known to be

compromised or involved in suspicious activity—such as a honeypot or other known entity—and then including connected graph elements as anomalous activity data. EX1004, 12:43-53 (describing “honey pot” as an example of “a particular host device 102 or other entity” associated with “anomalous activity data” and teaching to flag “any interaction with the honey pot entity ... as potentially malicious, and included in the anomalous activity data”). EX1003, ¶100.

Accordingly, Brezinski discloses performing a first correlation by identifying a known compromised entity and identifying as anomalous activity data all nodes (i.e., a first plurality of correlated nodes) connected to the node associated with the known compromised entity. EX1003, ¶101. As an example, referring back to the graph depicted in Figures 6 & 7 of Brezinski, a POSITA would recognize an obvious use case of vertex 402(1) being associated with a known compromised entity and according to Brezinski’s teachings including all edges connecting that node and the nodes connected by those edges as anomalous activity data. In the context of the exemplary graphs of Figures 6 & 7, these would correspond to vertices 402(2), 402(4), and 402(5) being identified as anomalous activity data, resulting in a correlation of those vertices as a “first plurality of correlated nodes.” EX1003, ¶101.

This disclosure therefore teaches that, for anomalous activity reflected by an event/edge associated with a node/vertex, Brezinski performs a graph-based

correlation that identifies multiple connected vertices—i.e., a “first plurality of correlated nodes”—using the graph representation. EX1003, ¶102.

b) **[1.10.b]**

[1.10.b] wherein each of the first plurality of correlated nodes corresponds to a respective event or resource,

Brezinski teaches this element. As explained above for element [1.4], *supra* Section VII.D.4, Brezinski discloses its graph includes vertices/nodes corresponding to a plurality of resources. Accordingly, a POSITA would recognize the obvious use case of Brezinski’s traversal procedure identifying as anomalous a plurality of resource nodes and such identified nodes would constitute a first plurality of correlated nodes corresponding to a respective resource. EX1003, ¶103. In addition, as explained for element [1.5], *supra* Section VII.D.5, Brezinski discloses receiving event data reflecting an event (e.g., a connection) between vertices/nodes and used to define edges corresponding to the event between vertices/nodes. Therefore, each node represented in Brezinski’s graph, including a first plurality of correlated nodes determined by Brezinski’s traversal procedure, also corresponds to a respective event (i.e., an edge connecting each of the first plurality of correlated nodes to the node associated with the anomalous event). EX1003, ¶103.

c) [1.10.c]

[1.10.c] wherein each respective event or resource is associated with the anomalous event, and

Brezinski teaches this element for each alternative way of performing a first correlation described above. *First*, Brezinski teaches this element because its traversal procedure identifies and outputs a subset of the event graph comprising vertices and edges selected because they are associated with the anomalous activity (i.e., the anomalous event) as determined by satisfying risk and rarity metric thresholds. *Supra* Section VII.D.10.a); EX1004, 18:9-31; EX1003 ¶104. Specifically, Brezinski states that “the graph 118 may be traversed along the edges 404 for which the rarity metric 502 satisfies [a criterion] and the risk metric 412 satisfies [a criterion]” and adds edges and vertices identified during such traversal as “anomalous activity data” EX1004, 11:54-57, 12:12-14, 1:41-46 (explaining that in a subset graph generated by a graph traversal, “the subset represents anomalous activity and a potential security risk”). Because the traversal includes only edges meeting the risk/rarity criteria and then includes those edges and their connected vertices/nodes in anomalous activity data, such edges and vertices (i.e., events and resources corresponding to the first plurality of correlated nodes) are associated with the determined anomalous activity (i.e., anomalous event). EX1003, ¶104.

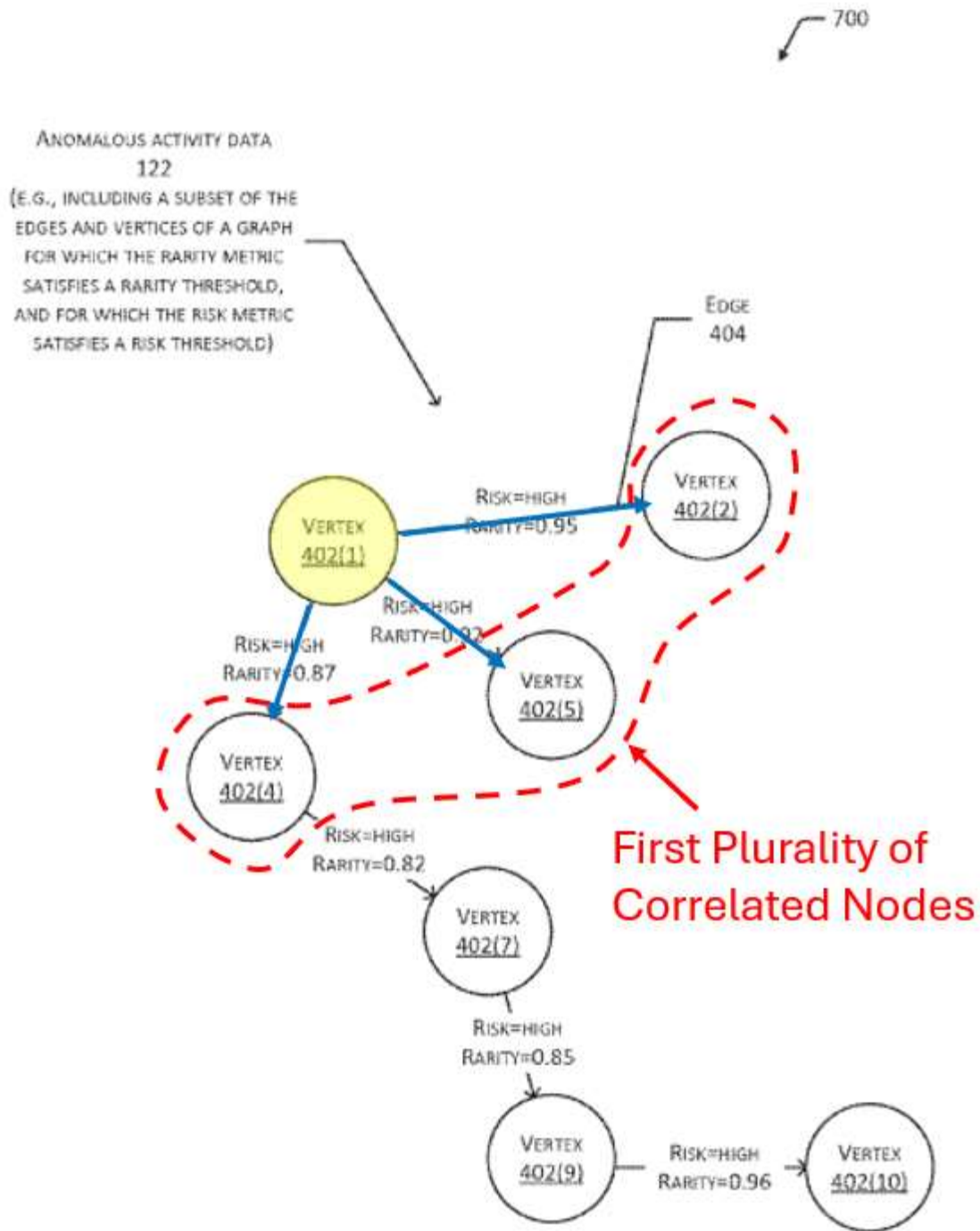
Second, Brezinski’s “honeypot” first correlation identifies all nodes interacting with a known compromised entity as anomalous activity data. *Supra* Section VII.D.10.a). Therefore, such nodes (i.e., respective resources) and their connected edges (i.e., respective events) are identified as being associated with an anomalous event (i.e., interactions by a compromised entity). This disclosure of Brezinski is consistent with how the ’628 patent’s describes an example correlation. *See* EX1001, 21:11-20 (describing correlating anomaly with any “available information that may be contextually linked to the anomaly”). EX1003, ¶105.

d) [1.10.d]

[1.10.d] wherein each of the first plurality of correlated nodes is connected by a respective edge of a second plurality of edges to the node associated with the anomalous event in the modified representation of the first graph;

For each of the two alternative first correlations taught by Brezinski, the first plurality of correlated nodes is connected by a respective edge of a plurality of edges (i.e., a second plurality of edges) to the node associated with the anomalous event. When performing a traversal of the modified graph as taught by Brezinski such a traversal identifies all edges immediately connecting the starting vertex having a risk and rarity rating above a threshold, and including the starting vertex, those identified edges, and vertices connected by those edges to the starting vertex to the anomalous activity data. *See* Section VII.D10.a *supra*; *see also* EX1004, 3:52-58, 18:9-31, 18:58-61. When identifying anomalous activity data based on a known compromised

entity, Brezinski teaches identifying the vertices having directly interacted with the vertex corresponding to the compromised entity as anomalous activity data. *See* Section VII.D10.a *supra*; EX1004, 12:43-53 In either alternative, all of the first plurality of correlated nodes are connected by respective edges directly to the originating node (i.e., the node associated with the anomalous event). For example, in a graph such as depicted in Brezinski's Figure 7, a POSITA would recognize the use case of vertex 402(1) being the originating vertex (either by being designated as the starting vertex for a traversal or as having been designated as a known compromised node). *See* EX1004, 12:42-45, 17:59-61, EX1003, ¶106. Either alternative would result in a first plurality of correlated nodes (in red) connected by a second plurality of edges (in blue) to a node associated with an anomalous event (in yellow) as shown in annotated Figure 7, below. EX1003, ¶106.



EX1004, FIG. 7 (annotated); EX1003, ¶106. As shown, each node of the first plurality of correlated nodes is connected by an edge (a second plurality of edges

annotated in blue) to the node associated with the anomalous event (vertex 402(1) annotated in yellow). EX1003, ¶106.

11. Element [1.11]

[1.11] for one or more of the first plurality of correlated nodes, perform a further correlation using the modified representation of the first graph to identify a second plurality of correlated nodes, wherein each of the second plurality of correlated nodes is connected through a respective edge of a third plurality of edges to the respective node of the first plurality of correlated nodes in the modified representation of the first graph;

As explained above for element [1.10.a], *supra* Section VII.D.10.a, Brezinski teaches performing a traversal of a modified graph to identify a first plurality of correlated nodes. Brezinski expressly discloses that its traversal algorithm “may be breadth-first such that neighboring nodes are explored prior to exploring deeper along a possible branch.” EX1004, 3:60-63. A POSITA would understand that a breadth-first search (BFS) performs correlation iteratively in levels: (i) first identifying a set of nodes directly connected to a starting/anomalous node (a first “plurality of correlated nodes”), and then (ii) continuing the traversal from those first-level nodes to identify additional nodes connected to the first-level nodes by additional edges (a second “plurality of correlated nodes”). *See Id.*, (“[T]he traversal may be breadth-first such that neighboring nodes are explored prior to exploring deeper along a possible branch.”); EX1019, 594 (explaining that in a breadth first search, the “algorithm discovers all vertices at distance k from s before discovering

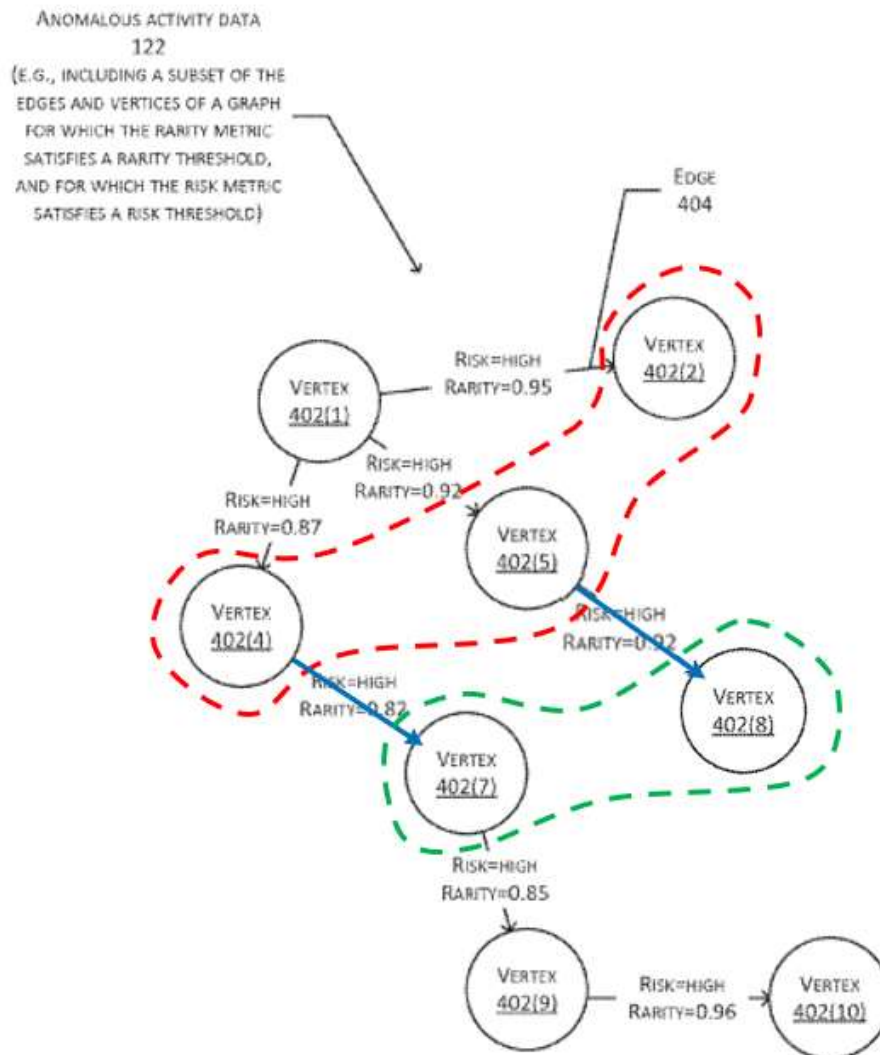
any vertices at distance $k + 1$,” where the distance is the number of edges). EX1003, ¶107. A breadth first traversal procedure such as Brezinski’s is iterative and proceeds by exploring nodes level-by-level. EX1003, ¶107. Therefore, Brezinski’s breadth first search would perform a first level search to identify those nodes directly connected to the node associated with the anomalous event (i.e., a first correlation identifying a first plurality of correlated nodes and) and then perform a further traversal/correlation from each vertex identified in the first plurality of correlated nodes to identify edges (and associated nodes) connected to the first plurality of correlated nodes that satisfy risk and rarity thresholds and add such edges and nodes to as anomalous activity data. EX1003, ¶107.

Alternatively, as explained above for element [1.10.a], Brezinski teaches performing a first correlation to determine a first plurality of correlated nodes based on identifying nodes that interacted with a known-compromised node/honey pot. It would have been obvious to a POSITA to subsequently implement Brezinski’s breadth first traversal procedure to identify additional anomalous activity data beginning with the vertices that are directly connected (via interaction edges) to the honeypot/known-compromised node, because those directly connected vertices represent the immediately implicated accounts/resources in the suspicious interaction. EX1003, ¶108. From there, Brezinski’s breadth-first traversal would proceed outward from those directly connected vertices to identify additional nodes

connected through additional edges to determine the potential spread of anomalous activity through the network from the known-compromised node/honey pot. EX1003, ¶108.

Thus, whether viewed as (i) second-level breadth first traversal following a first-level traversal, or (ii) traversal from a known compromised/honeypot node by first selecting the nodes that interacted with that node and then traversing outward, Brezinski teaches performing a further correlation from one or more first-level correlated nodes to identify second-level correlated nodes connected through respective edges, satisfying element [1.11]. EX1003, ¶109. An example of such a correlation is provided in the below Brezinski Figure 7, modified to illustrate an obvious use case with additional node (Vertex 402(8)) connected by an edge to Vertex 402(5).³

³ While Figure 6 shows Vertex 402(8), it is not included in the anomalous activity data reflected in Brezinski's Figure 7 due to the "low" risk metric of its connecting edge. However, Brezinski discloses that its "risk metric" reflects such attributes as access privileges, sensitive information, password protections, and encryption. EX1004, 2:65-3:16. A POSITA would have found it obvious that in a different network scenario, Vertices 402(5) and 402(8) and their interactions would have



EX1004, FIG. 7 (modified to add Vertex 402(8) and annotated); EX1003, ¶109. As shown in the above figure, such a breadth-first search identifies a second plurality of correlated nodes (in green) each connected by one of a third plurality of edges (in blue) to a respective node of the first plurality of correlated nodes (in red). While the different attributes such that the risk metric for their interaction would be “high,” resulting in Vertex 402(8) being included in the subset graph. EX1003, ¶109.

above figure shows one example of a traverse identifying a plurality of second-level nodes (i.e., a second plurality of correlated nodes) a POSITA would recognize additional obvious use cases. A POSITA understood that networks included many (potentially thousands or more) assets and interactions between them. EX1009, 673; EX1018, [0004]; EX1003, ¶109. The POSITA would therefore have found it obvious for a graph of a network, similar to the one depicted in Figure 6 of Brezinski, to depict additional edges between nodes, including edges between first level Vertices (402(2), 402(4), 402(5)) and other nodes not shown in Figure 6, having associated risk and rarity ratings such that they would be included as second-level nodes in the subset graph. EX1003, ¶109.

12. Element [1.12]

[1.12] generate a representation of a second graph comprising representations of one or more of the first plurality of correlated nodes, representations of one or more of the second plurality of correlated nodes, representations of one or more of the second plurality of edges, and representations of one or more of the third plurality of edges, wherein one or more of the second plurality of edges together with one or more of the third plurality of edges represent one or more event flows that could be involved in a cybersecurity attack; and

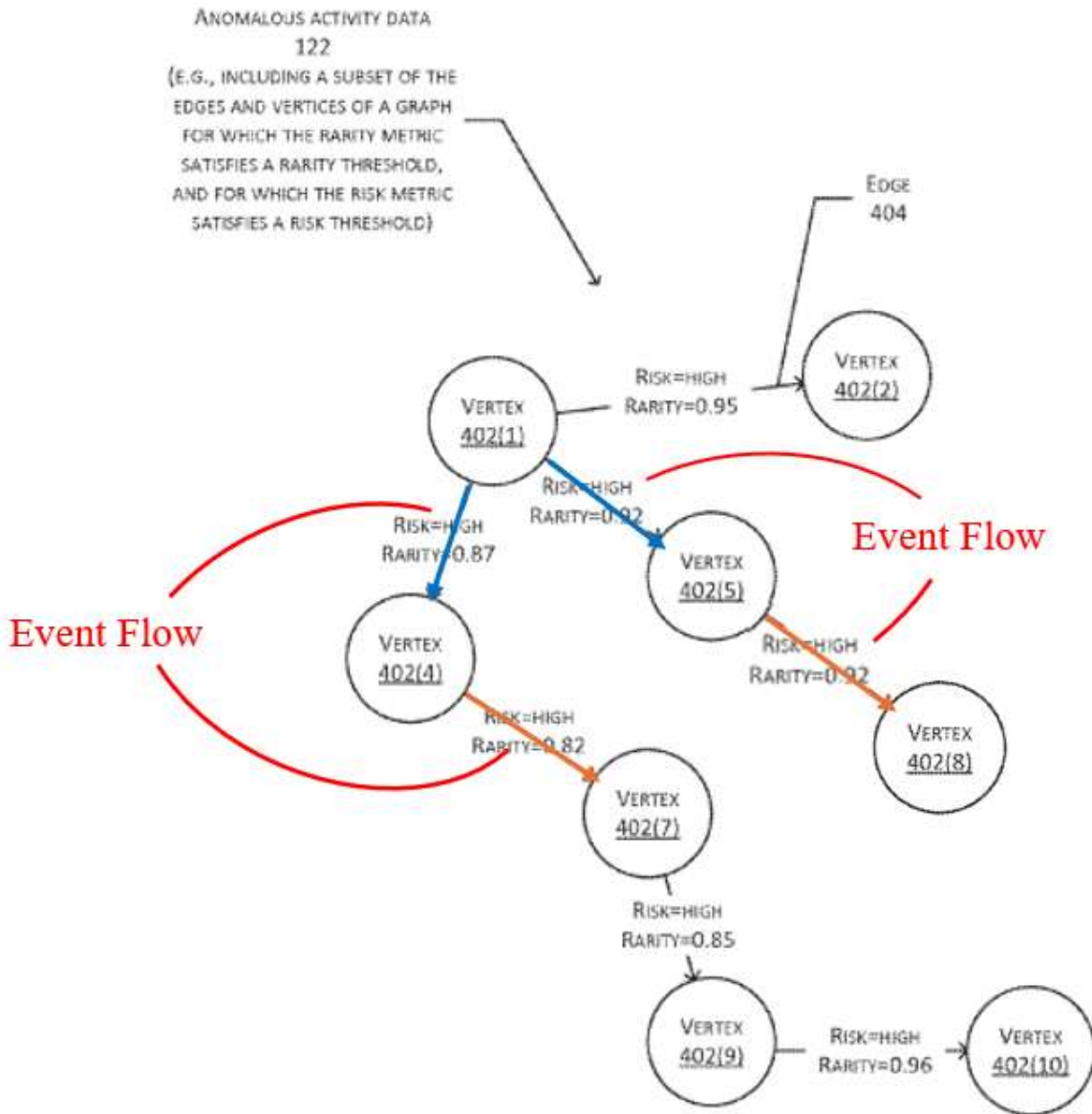
Brezinski teaches generating and outputting a second graph representation in the form of a subset of the event graph describing anomalous activity. Brezinski explains that “[t]he subset of the graph 118 determined by traversing the graph 118 may be described in anomalous activity data 122,” which may be “presented ... as the subset of the graph 118.” EX1004, 12:12-24 (“[T]he subset of the graph 118

determined by traversing the graph 118 may be described in anomalous activity data 122 [and] may be presented in a graphical form.”). Thus, Brezinski discloses generating a graph-based representation (a “subset of the graph”) separate from the full underlying graph—i.e., a “second graph” representation comprising the correlated vertices and edges. As explained above for elements [1.11] and [1.12], *supra* Sections VII.D.11-12, such anomalous activity data and resulting subset graph includes the first and second pluralities of correlated nodes and the second and third pluralities of edges. EX1003, ¶110.

Brezinski teaches that this output subset graph captures event flows because Brezinski states that anomalous activity data “may include a description of one or more events as tuples of vertex-edge-vertex,” including edge attributes such as timestamp and event type. EX1004, 12:27-29. Brezinski further describes the anomalous activity output as an “activity chain” and a “tree of linked [V-e-V] tuples,” reflecting multi-step sequences of events across multiple vertices connected by multiple edges. *Id.*, 19:17-22. A POSITA would understand that such activity chain and linked tuples (and the presented subset graph including such linked tuples) depict event flows through the environment—i.e., sequences of events connecting entities/resources—of the type investigated in cybersecurity attacks (e.g., movement from one compromised entity to another through successive interactions). EX1003, ¶111; *see also* EX1004, 4:5-8, 4:34-38 (“By correlating event data in a graph, and

traversing the graph according to risk and rarity of events, implementations may enable the detection of an attack.”). EX1003, ¶111.

An example of such a second graph including event flows is depicted in the below figure, which modifies Brezinski Figure 7 in the same way as previously described and presented for element [1.11], *supra* Section VII.D.11 *supra*. As with Figure 7 of Brezinski, the modified graph below depicts a subset of a graph following traversal and identification of edges and associated nodes exceeding a risk and rarity metric threshold. EX1004, 1:41-46; EX1003, ¶112.



EX1004, FIG. 7 (modified and annotated to show alternative use cases); EX1003, ¶112. As shown, the figure depicts two separate anomalous event flows (annotated in red), each of which include one of the third plurality of edges (shown in orange) together with one of the second plurality of edges (shown in blue) (i.e., a first event

flow from Vertex 402(1) to 402(4) to 402(7) and a second event flow from 402(1) to 402(5) to 402(8)). EX1003, ¶112.

13. Element [1.13]

[1.13] generate a report comprising information associated with the one or more event flows.

Brezinski teaches this element in that it discloses graphically presenting its anomalous activity data, including the subset graph and the event flows depicted in such graph. EX1004, 12:16-24 (“[T]he anomalous activity data 122 may be accessed and examined by... security analysts... or other personnel to determine whether the events included in the anomalous activity data 122 indicate an attack, exploit, or other form of security breach. **In such cases, the anomalous activity data 122 may be presented in a graphical form (e.g., as the subset of the graph 118) to facilitate analysis.**” (emphasis added)). By providing a graphical presentation of the subset graph including event flows within the graph for user analysis, Brezinski generates a report comprising information associated with the event flows (such information includes edges comprising event flows and nodes involved in event flows). EX1003, ¶113; *see also* EX1014, 450 (computer dictionary defining a “report” as “[t]he presentation of information about a given topic.”). EX1003, ¶113.

Crabtree reinforces this understanding by teaching that cybersecurity analysis results may be presented to users in the form of reports, dashboards, and

visualizations identifying relevant entities and resources. EX1005, [0063]-[0064]. Accordingly, Brezinski teaches graphically presenting (i.e., reporting) its subset graph including anomalous event flows, and Crabtree reinforces that it was well known to present such graph analysis results in a report. EX1003, ¶114.

E. Claim 4

[4] The computer system of claim 1, wherein the computer system is further configured to execute software instructions stored on nontransitory machine-readable storage media comprising software instructions that identify the anomalous event.

Brezinski teaches its analysis device identifying an anomalous event as part of its graph-based anomaly detection as explained in for element [1.10.a]. *See* Section VII.D.2 Brezinski explains that anomalous activity (i.e., an anomalous event) is identified by traversing the event graph and selecting edges/events meeting anomaly criteria, stating that “the graph may be traversed according to the risk and rarity metrics, **to identify patterns of anomalous activity in the event data.**” EX1004, Abstract (emphasis added); *see also id.*, 11:54-57, 12:12-24, 12:35-38. The resulting subset graph is “described in anomalous data 122.” 12:12-24, 12:35-38. Also as explained for element [1.10.a], *supra* Section VII.D.10.a, Brezinski teaches identifying anomalous activity by identifying edges and nodes interacting with a known compromised entity. Brezinski’s anomalous activity identification is

performed by the software instructions of Brezinski's graph analysis module.

Section VII.D.2, *supra*. EX1003, ¶115.

F. Claim 5

[5] The computer system of claim 4, wherein the anomalous event is identified by comparing one or more of the events relating to one or more entities of the first plurality of entities to a pattern of normal event behavior.

As explained for element [1.10] and claim 4, Brezinski teaches that anomalous activity is identified by traversing the event graph along edges (events) selected using a rarity metric. Sections VII.D.10 & E, *supra*. As Brezinski explains, its rarity metric indicates “a rarity or a frequency of events of that type that have occurred between the entities.” EX1004, 3:43-46. A POSITA would have understood that concept of a “rarity” metric requires a baseline of what is normal—i.e., the frequency/typicality of event behavior across the environment—so that an edge/event can be assessed as rare relative to that baseline. EX1003, ¶¶116-117. In other words, a POSITA would understand Brezinski's rarity metric as representing a comparison of a current event against a “pattern of normal event behavior” (events that commonly occur, in typical frequencies and combinations) to determine whether the current event is anomalous/abnormal. EX1003, ¶117. This comparison is expressly reflected in Brezinski's Formula 1, reproduced below, used to calculate a rarity metric.

$$\text{Rarity} = \frac{1}{1 + \log_2 |V1, \{E | \text{exclude timestamp}\}, V2|} \quad \text{Formula 1}$$

EX1004, 10:37-39. EX1003, ¶117.

As Brezinski explains, this formula is based on the count of the number of edges between two vertices, and for a given event type determines how rare or common such event type is within total count of edges (i.e., events) connecting the two vertices. EX1004, 10:41-58. Brezinski thereby determines how rare an event between a pair of nodes is based on the normal pattern of behavior between that node pair, and Brezinski subsequently uses that rarity metric to identify an anomalous event, i.e., by traversing a graph and including edges and connected nodes as anomalous activity data based on their exceeding a rarity threshold. EX1003, ¶118.

Alternatively, this element is taught by Crabtree. Crabtree explicitly discloses “collect[ing] activity information” on a network and processing such information to “analyze behavior patterns.” EX1005, [0072]. Crabtree further explains that such “behavior analytics information... may be received at a graphing service 145 for inclusion in a [cyber-physical graph].” *Id.*, [0076]. Moreover, Crabtree’s determined behavior patterns are used to recognize anomalous behavior. *Id.*, [0072] (explaining that “anomalous behavior may be recognized... based on a threshold of variance from an established pattern or trend”). Crabtree therefore compares events

relating to entities in a network graph against a pattern of normal behavior by comparing observed events to established behavioral patterns to determine whether the observed behavior exceeds a threshold of variance. EX1003, ¶119. The '628 patent describes identifying anomalous behavior based on a pattern of normal behavior in the same manner. *See* EX1001, 20:66-21:5 (“When an event occurs that is recognized to be outside the normal patterns, for example if the event exceeds a configured threshold for aberrance ... it may then be identified as an anomaly.”). EX1003, ¶119.

A POSITA would have been motivated to combine the teachings of Brezinski with Crabtree’s teaching that “anomalous behavior may be recognized ... based on a threshold of variance from an established pattern or trend” (EX1005, [0072]) because that teaching would have predictably improved Brezinski’s graph-based anomaly detection by providing an additional, complementary way to distinguish suspicious activity from ordinary network behavior. Brezinski already seeks to identify anomalous activity by traversing a graph of entities and events using rarity and risk criteria (*see* EX1004, 11:12-29); Crabtree’s variance-from-normal-pattern approach addresses the same objective—detecting cybersecurity-relevant anomalies—but does so by evaluating deviations from established behavioral baselines over time (EX1005, [0072]). A POSITA would have recognized that incorporating Crabtree’s behavior-pattern threshold based detection into Brezinski

would make Brezinski's anomaly determinations more robust, allowing the system not only to identify anomalies based on meeting risk and rarity criteria, but also to recognize when activity departs from normal trends for users, devices, or other entities represented in the graph. EX1003, ¶120. And Crabtree provides an explicit motivation for using its behavior-based anomaly detection, explaining that its method “for behavioral analytics enables proactive and high-speed reactive defense capabilities against a variety of cyberattack threats, including anomalous human behaviors as well as nonhuman ‘bad actors’..., enabling rapid response to threats to mitigate any potential impact” and a POSITA would implement such method into the system of Brezinski for the same reasons. EX1005, [0073], EX1003, ¶120. Because both references operate in the same field, use graph-based security information, and aim to detect anomalous or attack-related behavior, a POSITA would have had a reasonable expectation of success in implementing Crabtree's behavioral analysis into Brezinski's existing anomaly-analysis framework. EX1003, ¶120; *see also supra* Section VII.C (describing additional motivations to combine and reasonable expectation of success).

G. Claim 6

[6] The computer system of claim 1, wherein the computer system is further configured to execute software instructions stored on nontransitory machine-readable storage media comprising software instructions that identify a portion of the modified representation of the first graph that matches a known attack

pattern, wherein the portion of the modified representation of the first graph comprises the node associated with the anomalous event.

Brezinski teaches this element. As explained for element [1.10], Section VII.D.10 *supra*, Brezinski discloses identifying anomalous nodes based on interactions with a known compromised node or honey pot. And Brezinski discloses “identify[ing] a portion of the modified representation of the first graph” matching a known attack pattern by identifying as anomalous activity any edge connecting a node with such a known compromised/honey pot node. EX1004, 12:50-53. (“[A]ny interaction with the honey pot entity may be flagged as potentially malicious, and included in the anomalous activity data 122.”). By automatically classifying any interaction with a known compromised node/honey pot as anomalous activity, Brezinski treats such interaction as matching a known attack pattern because Brezinski recognized that other entities would have no other reason to interact with the known malicious entity/honey pot unless such interaction occurred as part of a potential cybersecurity attack; that is Brezinski recognized that interactions with a known compromised node or honey pot constituted known attack patterns. *See id.*, 12:47-50 (“The honey pot may be accessible to other (e.g., malicious) entities in the environment 100, but authorized entities may have no reason to interact with the honey pot.”); EX1003, ¶121. The identification of nodes and edges constituting anomalous activity data is performed by Brezinski’s graph analysis module of its

analysis device (*see* EX1004, FIG. 1), which, as explained for element [1.2] comprises computer instructions stored in nontransitory memory of Brezinski's analysis device for analyzing Brezinski's graph structure. *See* Section VII.D.2, *supra*. EX1003, ¶121.

The above teaching of Brezinski is consistent with the '628 patent's disclosures for this element because '628 patent's specification describes identifying anomalous events based on a known attack pattern in the very same way. *See* EX1001, 20:61-21:6⁴ (“When an event occurs that is recognized to be outside the normal patterns, for example if the event ... involves a known-compromised node or external address or resource (such as a deliberate ‘honeypot’ node for intrusion detection), it may then be identified as an anomaly.”). EX1003, ¶122.

H. Claim 9

[9] The computer system of claim 1, wherein at least one entity of the first plurality of entities is at least one of a user, a place, a device, a resource, an activity, an event, a group, or a service.

⁴ In prosecution, the Applicant identified this disclosure in the '628 patent's as-filed application as providing support for claim 6. *See* EX1002, 215/279 (identifying ¶ [83] of the as filed specification as providing specification support for claim 5), 86/279. EX1003, ¶122.

Brezinski teaches this element because the claim disjunctively claims (using the term “or”) eight different entity types and Brezinski’s teaches a first plurality of entities containing at least one of those types. As explained for element [1.4], Section VII.D.4 *supra*, the first plurality of entities comprises a plurality of resources. Brezinsky further describes “each vertex corresponding to an entity (e.g., a host **device, process or service**).” EX1004, 2:45-58 (emphasis added). EX1003, ¶123.

I. Claim 10

[10] The computer system of claim 1, wherein at least some of the edges of the second plurality of edges or of the third plurality of edges in the representation of the second graph are assigned a numerical weight that indicates a likelihood of a successful cybersecurity attack gaining access from one node to another.

Brezinski meets this element because its **risk metric–based graph traversal** identifies anomalous activity based on the likelihood of gaining access from one node to another. Brezinski explains that in a “malicious intrusion or attack..., an attacker may seek to leverage the initial point of compromise to gain access to other host devices, processes, services, or stored data that enables further access along a chain of multiple accesses.... Through this exploration, the attacker's access may spread until it is able to gain access to a potentially valuable target.” EX1004, 4:1-22. Accordingly, Brezinski discloses traversing its network graph using risk metric assigned to respective edges, and that such risk metrics “may be based on a potential impact or access capability of the associated event.” *Id.*, 9:8-10. For example,

Brezinski discloses assigning a higher risk score for an edge associated with a user having “wide access privileges” (*id.*, 9:19) or if the associated event for the edge “accesses a host device 102, a process 104, or a service 106 that enables access to sensitive, confidential, or private information, or if the associated event accesses information that enables such access.” *Id.*, 9:26-30. Brezinski therefore teaches assigning a higher risk score to edges associated with greater access to other entities (e.g., resources) in the network, with such greater access indicating a higher likelihood of a successful cybersecurity attack gaining access to such entities (i.e., from one node to another). EX1003, ¶124. Similarly, Brezinski teaches that its risk metric for an edge may be based on whether a connection is secured (e.g., encrypted) or unsecured (e.g., unencrypted). EX1004, 3:5-12. A POSITA would recognize that whether a connection is secured or not would determine the likelihood of success of a cyberattack gaining access to a node through such connection, and that this would be reflected in the risk metric associated with the connection. EX1003, ¶124. Brezinski also discloses that its risk score may be reflected as a numerical weight. EX1004, 9:42-44 (“The risk metric 412 may range from 0 to 10, with a higher number corresponding to a higher security risk for an entity or an attribute of an event.”). Brezinski therefore teaches assigning such numerical risk scores indicating a likelihood of a successful cybersecurity attack gaining access to a node to edges within its graph, including to the respective second and third pluralities of edges

described for elements [1.10] and [1.11] (Sections VII.D.10-11 *supra*). EX1003, ¶124.

J. Claim 11

1. [11.1]-[11.5], [11.8], [11.11] And [11.15]-[11.17]

Elements [11.1]-[11.5], [11.8], [11.11], [11.15]-[11.17], are substantively identical to elements [1.1]-[1.5], [1.6], [1.8] and [1.11]-[1.13], respectively.⁵ See EX1010 (claims listing showing claims 1 and 11 side-by-side). Brezinski-Crabtree satisfies those elements for the same reasons explained above for elements [1.1]-[1.5], [1.6], [1.8] and [1.11]-[1.13]. *Supra* Sections VII.D.1-6, 8, 11-13. EX1003, ¶125.

2. [11.6]

[11.6] wherein a representation of a first node of the first plurality of nodes includes information about a first security vulnerability associated with a first entity, wherein the first node corresponds to the first entity, and

Brezinski discloses that edges in its graphical representation are represented with a “risk metric indicating a level of security risk associated with the event.”

⁵ Elements [1.12] and [11.15] differ only in that the former uses the term “represent” while the latter uses “represents” but this does not create a substantive difference in the elements. The remaining elements are verbatim identical. EX1010.

EX1004, 2:65-67; *see also id.*, FIG. 6 (depicting edges assigned with risk metric ratings). Brezinski also discloses that “the risk metric [] for an edge [] may be determined by accessing security risk data which lists a risk metric [] for one or more particular host devices [], processes [], [or] services” (i.e., entities). *Id.*, 9:37-41. Such a risk metric associated with an entity constitutes “information about a security vulnerability” because Brezinski describes such metric reflecting whether such entity “enables access to sensitive, confidential, or private information;” that is, it describes the extent to which the entity makes the system vulnerable to a cybersecurity attack. *Id.*, 9:26-30; *see also id.*, FIG. 10 (Step 1008: “Access security risk data describing one or more security risk levels associated with the entities”), 16:51-54; EX1003, ¶126. And Brezinski expressly teaches associating risk/security information with the vertex representation of an entity. EX1004, 17:11-14 (disclosing assigning, in a graph “a risk metric . . . to one or both of the vertices [] connected by [an] edge []....”), FIG. 10 (Step 1008). Accordingly, a POSITA implementing Brezinski-Crabtree would have understood it to be desirable to store, as part of the node/vertex representation for graphed entities, entity-specific security-risk information reflecting vulnerability-related “security risk data”) to provide a more complete and accurate representation of the security risks present in the network. *Id.*, 9:37-41; EX1003, ¶126. A POSITA would therefore include such a representation of a security vulnerability (i.e., a first security vulnerability) for

graphed nodes, including for a first node corresponding to a first entity. EX1003, ¶126.

Alternatively, Crabtree teaches including information about security vulnerabilities about entities in a network security graph. Crabtree discloses a “method [] for continuous network resilience scoring” to “measure an overall level of risk for a network infrastructure.” EX1005, [0077]. This method involves “first collecting [] information on publicly-disclosed vulnerabilities, such as (for example) using the Internet or common vulnerabilities and exploits (CVE) process.” *Id.* “This information may then [] be incorporated into a CPG..., and the combined data of the CPG and the known vulnerabilities may then be analyzed [] to identify the relationships between known vulnerabilities and risks exposed by components of the infrastructure.” *Id.* Crabtree therefore teaches including information about known security vulnerabilities for the entities in its network graph. EX1003, ¶127.

A POSITA would have been motivated to combine Brezinski with Crabtree’s teaching of including security-vulnerability information for entities in the graph because that addition would have predictably improved Brezinski’s graph-based anomaly analysis. EX1003, ¶128. Crabtree expressly describes the benefit of its approach, stating that “the combined data of the CPG and the known vulnerabilities may then be analyzed 1203 to identify the relationships between known vulnerabilities and risks exposed by components of the infrastructure.” EX1005,

[0077]. A POSITA would have therefore recognized that incorporating Crabtree's vulnerability information into Brezinski's graph would make Brezinski's graph more useful for analysis purposes, because suspicious events involving vulnerable hosts, services, or other resources could be prioritized and interpreted in light of the entities' known exposure to attack. EX1003, ¶128. A POSITA would have had a reasonable expectation of success in doing so because Brezinski already teaches assigning security-related data (i.e., risk metric data) to nodes and would therefore expect success in also including vulnerability information as taught by Crabtree. EX1004, 2:65-67, 17:11-14; EX1003, ¶128. A POSITA would therefore find it obvious to include such security vulnerability information into Brezinski's graph, including for a first node corresponding to a first entity. EX1003, ¶128; *see also supra* Section VII.C (describing additional motivations to combine and reasonable expectation of success).

3. [11.7]

[11.7] wherein the first security vulnerability is identified based on a database of known security vulnerabilities;

Brezinski teaches identifying an entity's risk metric (i.e., security vulnerability) from a database maintained by the analysis device. Brezinski explains that the analysis device's memory includes "data storage 922" and that "[t]he data storage 922 may comprise a database ... and may be a relational or a non-relational

datastore.” EX1004, 15:58-62. Brezinski further explains that this data storage “may store security risk data 924 that may provide a risk metric [] associated with” entities such as “host devices [], processes [], [and] services,” and that the security risk data is accessed and used by the risk analysis module. *Id.*, 16:1-5 Accordingly, Brezinski teaches a database of known security vulnerabilities for entities within the network used to provide risk metrics for those entities, including for the first security vulnerability described for element [11.6] (Section VII.J.2, *supra*). EX1003, ¶129.

Alternatively, as explained for element [11.6], Section VII.J.2, *supra*, Crabtree teaches collecting known security vulnerabilities, including a first security vulnerability. EX1005, [0077]. A POSITA implementing Crabtree’s teaching of including security-vulnerability information for entities in the graph-based system of Brezinski would have naturally, after collecting such information, stored it in a database, such as Brezinski’s, in order to make such information available to the analysis device in building and analyzing its network graph structure. *See* EX1004, 15:58-60, EX1003, ¶130. Such vulnerability information for an entity would therefore be identified in the database for use in the graph. EX1003, ¶130.

4. [11.9]

Element [11.9] differs from element [1.7] only in that the former recites “identify a second entity that does not correspond to any of the first plurality of nodes” while the latter recites “identify a first entity that does not correspond to any

of the first plurality of nodes.” EX1010. As explained for element [1.7], Section VII.D.7 *supra*, Brezinski discloses identifying a newly detected entity in newly received event data and adding it to a graph as a new node. While such disclosure was mapped to a “first entity” for purposes of element [1.7], such disclosure equally satisfies the “second entity” for purposes of element [11.9]. EX1003, ¶131.

5. [11.10]

[11.10] identify a second security vulnerability associated with the second entity,

As explained for elements [11.6] and [11.7], Section VII.J.2-3 *supra*, Brezinski, and alternatively Crabtree, teach identifying security vulnerabilities and including information related to such security vulnerabilities for graphed nodes corresponding to respective network entities. Accordingly, such security vulnerability would also be identified and included in the graph for a node corresponding to a second entity for the same reasons as explained for the first vulnerability of elements [11.6] and [11.7]. EX1003, ¶132.

6. [11.12]

Element [11.12] differs from element [1.9] only in that the former recites a “second entity” and “second node” where the latter respectively recites a “first entity” and “first node.” EX1010. As explained above (*supra* Section VII.J.4) Brezinski’s disclosures teaching a “first entity” for purposes of claim 1 also equally apply to the “second entity” for purposes of claim 11. And, as explained for element

[1.9], Section VII.D.9 *supra*, Brezinski teaches modifying a graph to add a new node corresponding to a newly identified entity. While such disclosure was mapped to a “first node” for purposes of element [1.9], such disclosure equally satisfies the “second node” for purposes of element [11.12]. EX1003, ¶133.

7. [11.13]

[11.13] and wherein the modified representation of the first graph includes information about the second security vulnerability associated with the second entity,

As explained for element [11.10], Section VII.J.5 *supra*, Brezinski, and alternatively Crabtree, teaches identifying a second security vulnerability associated with a second entity. And as explained for element [11.6], Section VII.J.2 *supra*, Brezinski, and alternatively Crabtree, teaches storing security vulnerability information for graphed nodes corresponding to respective network entities. Accordingly, Brezinski-Crabtree includes in its modified graph information about the second vulnerability for the same reasons as explained for the first vulnerability of element [11.6]. EX1003, ¶134.

8. [11.14]

Element [11.14] differs from element [1.10] only in that the former recites “a first event” while the latter recites “an anomalous event.” EX1010. Brezinski teaches an anomalous event for the reasons explained for element [1.10], Section VII.D.10,

and this same anomalous event is a “first event” for purposes of element [11.14].
EX1003, ¶135.

K. Claim 13

[13] The computer system of claim 11, wherein the further correlation using the modified representation of the first graph further identifies a plurality of respective correlated edges that each involve the respective event or resource.

Brezinski teaches this element. As explained for elements [1.10] and [11.14], Sections VII.D.10 & J.8, Brezinski teaches a first correlation identifying nodes corresponding to respective resources (e.g., a device). And as explained for elements [1.11] and [11.15], Sections VII.D.11 & J.1, Brezinski teaches a further traversal iteration that involves a further correlation identifying edges and nodes connecting the first plurality of correlated nodes. A POSITA would understand the obvious use case of one of the first plurality of correlated nodes having multiple outgoing connections/edges to multiple neighboring nodes. *See e.g.*, EX1004, FIG. 4 (showing, as an example, Vertex 402(4) connected by multiple edges to neighboring Vertices 402(6) and 402(7)); EX1003, ¶136. In the situation where corresponding risk and rarity metrics exceeded thresholds, then the Brezinski analysis device would include the multiple outgoing edges (i.e., a plurality of respective correlated edges that each involve the respective resource) as anomalous data. EX1003, ¶136. Alternatively, a node included in the first plurality of correlated nodes may have multiple connections and corresponding edges to the same neighboring node with

each such edge included as anomalous activity data. *See e.g.*, EX1004, FIG. 4 (showing multiple edges connecting Vertex 402(5) and Vertex 402(8)). Accordingly, those multiple outgoing edges constitute a plurality of respective correlated edges that involve the respective resource because they each connect to the node corresponding to the respective resource. EX1003, ¶136.

L. Claim 14

[14] The computer system of claim 11, wherein the first event is an anomalous event.

As explained for element [11.14], Section VII.J.8 *supra*, that element differs from element [1.10] in that the former recites “a first event” while the latter recites “an anomalous event.” Brezinski teaches an anomalous event for the reasons explained for element [1.10], Section VII.D.10 *supra*, and this same anomalous event is a “first event.” EX1003, ¶137.

M. Claims 15 & 17

Aside from their claim dependencies, claims 15 and 17 are substantively the same as claims 5 and 6, respectively, with the only difference being that claims 15 and 17 recite a “first event” and claims 5 and 6 recite an anomalous event. EX1010. As explained for element 11.14 and claim 14, *supra* Sections VII.J.8 & L, Brezinski teaches a first event that is an anomalous event. This first event meets the elements

of claim 15 and 17 for the same reasons explained for claims 5 and 6, Sections VII.F & G, respectively. EX1003, ¶138.

N. Claim 20

Aside from their claim dependencies, claim 20 is substantively the same as claim 9 (*see* EX1010) and Brezinski teaches claim 20 for the same reasons as explained for claim 9, *supra* Section VII.H. EX1003, ¶139.

O. Claim 22

[22] The computer system of claim 11, wherein the node associated with the first event corresponds to a third security vulnerability.

As explained for element [11.6], *supra* Section VII.J.2, Brezinski, and alternatively Crabtree, teach including security vulnerability information for graphed nodes corresponding to entities, which would also apply to the node corresponding to the first event. Accordingly, Brezinski-Crabtree teaches associating a third security vulnerability to the node associated with the first event for the same reasons as for a first security vulnerability associated with a first entity as explained for element [11.6]. EX1003, ¶140. Alternatively, as explained for element [11.6], Crabtree teaches including in a graph security vulnerability information associated with the graph's entities, and therefore teaches identifying a third security vulnerability corresponding to an entity (and its corresponding node)

associated with the first event for the same reasons as for a first security vulnerability associated with a first entity. EX1003, ¶140.

VIII. NO OBJECTIVE INDICIA OF NON-OBVIOUSNESS

Petitioner is unaware of any evidence of objective indicia of nonobviousness having a nexus to the challenged claims.

IX. CONCLUSION

Petitioner requests institution for claims 1, 4-6, 9-11, 13-15, 17, 20, and 22, on Ground 1 specified in this petition.

Respectfully submitted,

Dated: April 3, 2026

By: /Andrew M. Mason/
Andrew M. Mason, Reg. No. 64,034
andrew.mason@klarquist.com

Counsel for Petitioner

**CERTIFICATE OF COMPLIANCE WITH
TYPE-VOLUME LIMITATION PURSUANT TO 37 C.F.R. § 42.24**

This brief complies with the type-volume limitation of 37 C.F.R. § 42.24(a)(1)(ii).

The brief contains **13,835** words, excluding the parts of the brief exempted by 37 C.F.R. § 42.24(a).

The brief has been prepared in a proportionally spaced typeface using Microsoft Word for O365 in a 14-point Times New Roman font.

Dated: April 3, 2026

By: /Andrew M. Mason/

Andrew M. Mason, Reg. No. 64,034
andrew.mason@klarquist.com
KLARQUIST SPARKMAN, LLP
One World Trade Center, Suite 1600
121 S.W. Salmon Street
Portland, Oregon 97204
Tel: 503-595-5300
Fax: 503-595-5301

Counsel for Petitioner

CERTIFICATE OF SERVICE
IN COMPLIANCE WITH 37 C.F.R. § 42.6(e)(4)

The undersigned certifies that the **PETITION FOR *INTER PARTES* REVIEW OF U.S. PATENT NO. 12,301,628 and Exhibits 1001 – 1023** were served on April 3, 2026, via **Express Mail** on the Patent Owner at the following address of record as listed on the USPTO Patent Center:

NK Patent Law
4101 Lake Boone Trail, Suite 218
Raleigh, NC 27607

By: /Andrew M. Mason/
Andrew M. Mason, Reg. No. 64,034
andrew.mason@klarquist.com
KLARQUIST SPARKMAN, LLP
One World Trade Center, Suite 1600
121 S.W. Salmon Street
Portland, Oregon 97204
Tel: 503-595-5300
Fax: 503-595-5301

Counsel for Petitioner