UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

NETSKOPE, INC.,

Petitioner,

v.

FORTINET, INC.,

Patent Owner.

PTAB Case No. IPR2026-00473

U.S. Patent No. 11,290,527

DECLARATION OF PAUL D. MARTIN, PH.D. IN SUPPORT OF PETITION FOR *INTER PARTES* REVIEW OF U.S. PATENT NO. 11,290,527

TABLE OF CONTENTS

I. INTRODUCTION 1
A. Qualifications 2
1. Professional Background 2
2. Teaching 2
3. Research 3
4. Patents 8
5. Industry 8
6. Curriculum Vitae 13
B. Materials Reviewed 13
C. Level of Ordinary Skill in the Art 15
D. Summary of Opinions 17
II. OVERVIEW OF THE TECHNOLOGY 17
A. Priority Date of the Claims 17
B. Overview of Relevant Technology 18
1. Cloud Resource Tagging 18
2. Tags to Govern and Secure Cloud Resources 22
C. The '527 Patent 23
1. Overview 23
2. The CTOS and Security Manager Architecture 25
D. The Challenged Claims 28
E. Claim Construction 28
III. UNPATENTABILITY OF THE '527 PATENT CLAIMS 29
A. Legal Standards 29
1. Obviousness 29
B. Ground 1: Foskett renders claims 1, 3, 4, 6, 7, 8, 10, 11, 13, 14, 15, 17, and 20 obvious. 30
1. Claim 1: 36
2. Claim 3: 57

"The method of claim 1, wherein the cloud environment comprises a hybrid multi-cloud environment including a private cloud and a plurality of public clouds." 57

3. Claim 4: 60

"The method of claim 1, wherein said assigning further enables visibility and control for the plurality of cloud resources by the security manager." 60

4. Claim 6: 62

"The method of claim 1, wherein the plurality of cloud resources comprise virtual machine instances." 62

5. Claim 7: 63

"The method of claim 1, wherein the security manager comprises a cloud based security management appliance." 63

6. Claim 8: 64

7. Claim 10: 66

"The non-transitory computer-readable storage medium of claim 8, wherein the cloud environment comprises a hybrid multi-cloud environment including a private cloud and a plurality of public clouds." 66

8. Claim 11: 67

"The non-transitory computer-readable storage medium of claim 8, wherein said assigning further enables visibility and control for the plurality of cloud resources by the security manager." 67

9. Claim 13: 67

"The non-transitory computer-readable storage medium of claim 8, wherein the plurality of cloud resources comprise virtual machine instances." 67

10. Claim 14: 68

"The non-transitory computer-readable storage medium of claim 8, wherein the security manager comprises a cloud based security management appliance." 68

11. Claim 15: 68

12. Claim 17: 70

"The system of claim 15, wherein the cloud environment comprises a hybrid multi-cloud environment including a private cloud and a plurality of public clouds." 70

13. Claim 20: 70

"The system of claim 15, wherein the plurality of cloud resources comprise virtual machine instances." 70

C. Ground 2: Combination of Foskett and Whitlock renders claims 2, 5, 9, 12, 16, and 19 obvious. 70

1. Motivations to combine 72

2. Claim 2: 74

"The method of claim 1, further comprising facilitating, by the security manager, application of a security policy expressed with reference to one or more unified tags of the plurality of unified tags by retrieving information associated with a particular cloud resource of the plurality of cloud resources from a storage database." 74

3. Claim 5: 78

"The method of claim 1, wherein the pre-defined global tagging policy identifies the unified tag based on one or more of the information regarding the cloud resource, a location of the resource, and a function of the cloud resource." 78

4. Claim 9: 81

"The non-transitory computer-readable storage medium of claim 8, wherein the method further comprises facilitating application of a security policy expressed with reference to one or more unified tags of the plurality of unified tags by retrieving information associated with a particular cloud resource of the plurality of cloud resources from a storage database." 81

5. Claim 12: 81

"The non-transitory computer-readable storage medium of claim 8, wherein the pre-defined global tagging policy identifies the unified tag based on one or more of the information regarding the cloud resource, a location of the resource, and a function of the cloud resource." 81

6. Claim 16: 82

"The system of claim 15, wherein the method further comprises facilitating application of a security policy expressed with reference to one or more unified tags of the plurality of unified tags by retrieving information associated with a particular cloud resource of the plurality of cloud resources from a storage database." 82

7. Claim 19: 82

"The system of claim 15, wherein the pre-defined global tagging policy identifies the unified tag based on one or more of the information regarding the cloud resource, a location of the resource, and a function of the cloud resource." 82

D. Ground 3: Combination of Foskett and Juniper renders claim 18 obvious. 82

1. Motivations to combine 83

2. Claim 18: 85

"The system of claim 15, wherein said assigning enables visibility and control of the plurality of cloud resources via a single pane of glass user interface." 85

E. Ground 4: Combination of Kung and Serhiienko renders claims 1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 13, 14, 15, 16, 17, and 20 obvious. 86

1. Motivations to combine 91

2. Claim 1: 93

3. Claim 2: 108

"The method of claim 1, further comprising facilitating, by the security manager, application of a security policy expressed with reference to one or more unified tags of the plurality of unified tags by retrieving information associated with a particular cloud resource of the plurality of cloud resources from a storage database." 108

4. Claim 3: 110

"The method of claim 1, wherein the cloud environment comprises a hybrid multi-cloud environment including a private cloud and a plurality of public clouds." 110

5. Claim 4: 111

"The method of claim 1, wherein said assigning further enables visibility and control for the plurality of cloud resources by the security manager." 111

6. Claim 6: 113

"The method of claim 1, wherein the plurality of cloud resources comprise virtual machine instances." 113

7. Claim 7: 114

"The method of claim 1, wherein the security manager comprises a cloud based security management appliance." 114

8. Claim 8: 116

9. Claim 9: 118

"The non-transitory computer-readable storage medium of claim 8, wherein the method further comprises facilitating application of a security policy expressed with reference to one or more unified tags of the plurality of unified tags by retrieving information associated with a particular cloud resource of the plurality of cloud resources from a storage database." 118

10. Claim 10: 119

"The non-transitory computer-readable storage medium of claim 8, wherein the cloud environment comprises a hybrid multi-cloud environment including a private cloud and a plurality of public clouds." 119

11. Claim 11: 119

"The non-transitory computer-readable storage medium of claim 8, wherein said assigning further enables visibility and control for the plurality of cloud resources by the security manager." 119

12. Claim 13: 119

"The non-transitory computer-readable storage medium of claim 8, wherein the plurality of cloud resources comprise virtual machine instances." 119

13. Claim 14: 120

"The non-transitory computer-readable storage medium of claim 8, wherein the security manager comprises a cloud based security management appliance." 120

14. Claim 15: 120

15. Claim 16: 123

"The system of claim 15, wherein the method further comprises facilitating application of a security policy expressed with reference to one or more unified tags of the plurality of unified tags by retrieving information associated with a particular cloud resource of the plurality of cloud resources from a storage database." 123

16. Claim 17: 123

"The system of claim 15, wherein the cloud environment comprises a hybrid multi-cloud environment including a private cloud and a plurality of public clouds." 123

17. Claim 20: 123

"The system of claim 15, wherein the plurality of cloud resources comprise virtual machine instances." 123

F. Ground 5: Combination of Kung, Serhiienko, and Whitlock renders claims 5, 12, and 19 obvious. 124

1. Motivations to combine 124

2. Claim 5:..... 126

"The method of claim 1, wherein the pre-defined global tagging policy identifies the unified tag based on one or more of the information regarding the cloud resource, a location of the resource, and a function of the cloud resource."..... 126

3. Claim 12:..... 129

"The non-transitory computer-readable storage medium of claim 8, wherein the pre-defined global tagging policy identifies the unified tag based on one or more of the information regarding the cloud resource, a location of the resource, and a function of the cloud resource." 129

4. Claim 19:..... 129

"The system of claim 15, wherein the pre-defined global tagging policy identifies the unified tag based on one or more of the information regarding the cloud resource, a location of the resource, and a function of the cloud resource."..... 129

G. Ground 6: Combination of Kung, Serhiienko, and Juniper renders claim 18 obvious. 130

1. Motivations to combine 130

2. Claim 18:..... 131

"The system of claim 15, wherein said assigning enables visibility and control of the plurality of cloud resources via a single pane of glass user interface" 131

LIST OF APPENDICES

Appendix A *Curriculum Vitae* of Paul Martin, Ph.D.

Netskope Ex. 1002
Netskope v. Fortinet IPR2026-00473
Page 00008

I. INTRODUCTION

1. I, Paul D. Martin, Ph.D., have been retained by Petitioner Netskope, Inc. ("Petitioner") to investigate and opine on certain issues relating to United States Patent No. 11,290,527 ("the '527 patent") in its Petition for *Inter Partes* Review of that patent. The Petitions request that the Patent Trial and Appeal Board ("PTAB" or "Board") review and cancel claims 1-20 of the '527 patent. I have analyzed the claims of the '527 patent in numerical order.

2. The opinions set forth in this declaration are based on my personal knowledge, my professional judgment, and my analysis of the materials and information referenced in this declaration and its exhibits.

3. If called upon to testify, I could and would testify to the truth of the following.

4. The company for which I work, Harbor Experts, is being compensated at the rate of $775 per hour for time that I spend on this case. Harbor Experts is also being reimbursed for my reasonable and customary expenses. I receive no other forms of compensation related to this case. My compensation does not depend on the outcome of this *inter partes* review or the co-pending district court litigation, and I have no other financial interest in this post-grant review.

5. I understand that the '527 Patent has been assigned to Fortinet, Inc.

6. This declaration is based on the information currently available to me. To the extent that additional information becomes available, I reserve the right to continue my investigation and study, which may include a review of documents and information that may be produced, as well as testimony from depositions that have not yet been taken.

A. Qualifications

1. Professional Background

7. I am the Chief Scientist at Harbor Experts, Inc. and a part-time Lecturer in the Department of Computer Science at Johns Hopkins University.

8. I completed my B.S., M.S.E., and Ph.D. degrees from Johns Hopkins University, with all degrees, including my doctorate, being in computer science.

2. Teaching

9. I am a part-time lecturer in the Department of Computer Science at Johns Hopkins University. I currently teach classes on computer and network security and applied cryptography, and C and C++ programming, developer tools and Linux environments. In the past I have taught classes on the topic of hardware hacking, including analyzing, modifying, and repairing computer hardware and embedded devices and vulnerability analysis and exploitation. I also occasionally substitute teach classes on network security and cryptography.

10. My classes cover a diverse array of topics including hardware and software security design, analysis and reverse engineering for vulnerability assessment; hardware and software-based attacks on computer components including RAM and CPUs; applied cryptography; computer architecture; computer networking; and component-level repair and modification. My classes also utilize hardware and software emulation and virtualization techniques in order to present students with a variety of real-world environments in which to learn and perform hands-on projects.

3. Research

11. In spring 2011, I completed my bachelor's in computer science at Johns Hopkins University. In fall 2011, I began working towards my Ph.D. in computer science, also at Johns Hopkins University, and I began researching computer security and privacy systems in order to both design novel technologies for securing computing systems and to also bridge the interface gap between users and their technology. As such, my research focused not only on developing technical solutions to novel security and privacy systems but to also presenting these solutions in a way that non-technical users could understand. Part of my research also focused on non-deterministically generating virtualized computer networks in order to provide malware education environments for students. I also developed an emulator for analyzing malware code segments as part of a student

project with the NSA. This emulator was integrated into an IDA Pro plugin and was able to observe how binary instruction snippets taken from malware samples modified their computing environment.

12. During my tenure as a Ph.D. student at Johns Hopkins, I was fully funded as a research assistant and/or teaching assistant throughout all of my semesters. I was a member of the Upsilon Pi Epsilon computer science honor society, as well as its treasurer for the 2014-2015 school year. I was also a member of the Johns Hopkins chapter of the Association for Computing Machinery. I earned an award for outstanding teaching assistant in the Computer Science department for the year 2014-2015. I co-instructed a short course called, "Introduction to Hardware Hacking," with a colleague, Dr. Michael Rushanan, which was the highest-rated course in the computer science department (based on student reviews) during the winter session in which it was offered. In this course, we offered lessons on a variety of topics including modifying game consoles and device firmware; electronics repair; binary analysis and modification, network traffic analysis; and web-based vulnerability assessment and exploitation.

13. During my time as a student at Johns Hopkins, I also participated in numerous kinds of service to the department. I represented the university as a student speaker at numerous open houses and admitted student days to help introduce prospective students to the computer science department. I was selected

to speak to the external advisory board for the computer science department to provide insight into the student experience in the department. I served the department as the faculty liaison czar for several semesters, in which I attended and documented faculty meetings, which I then summarized for the students in the department. I also mentored and supervised a high school student on a year-long student project in software design and I supervised a variety of undergraduate research assistants on some of my research projects.

14. I finished my Ph.D. by successfully defending my dissertation, entitled "Securing Medical Devices and Protecting Patient Privacy in the Technological Age of Healthcare," on February 12, 2016, at the age of 26. My doctoral thesis tells the story of a secure healthcare practice of the future in which technology is seamless to use for healthcare providers while providing a much higher standard of security than is typical today. It focuses on the juxtaposition of usability and security with an emphasis on simplicity, automation, and error-proofing of security controls.

15. During my doctorate and afterward, I have produced, peer-reviewed, and published research in areas such as fingerprinting, anomaly detection, multifactor authentication, and embedded systems security. In many of these cases, I focused not only on designing novel security systems but also on designing usable web-based security systems for controlling them and understanding their

data output. In fact, some of my research has formed the basis for commercial products and services. For example, in my work on integrated audit and access control, which was funded in part by Accenture Labs, I developed a Hadoop-based application to perform large-scale statistical analysis of audit logs from an electronic medical record system. As part of this work, I also designed a web application to automatically produce human-readable reports and graphs for use in consulting. Accenture subsequently patented this technology.

16. Similarly, in my research at Applied Communication Sciences in 2013, I designed and implemented a web-based traffic visualization dashboard and analysis system for field area smart grid networks that could be used to quickly gain an understanding of the current state of a SmartGrid network as well as to detect unexpected anomalies in the network. This dashboard was able to aggregate and visualize multiple data streams in realtime. Applied Communication Sciences subsequently patented this work and continued to build on the project. To my knowledge, they sold and/or still sell this product as part of their SecureSmart Managed Security Service product offering.

17. I have worked on and published two research projects related to improving the quality and ease-of-use of authentication technologies, especially in healthcare settings. In one project, I and my co-authors designed an indoor location tracking system consisting of unspoofable Bluetooth Low Energy beacons. These

beacons were placed around a building and mapped to a backend such that a user could report which beacons he or she was within range of in order to be used as a secondary authentication mechanism for accessing patient medical records. In the way that we designed and envisioned the system, a doctor would log into a mobile device which would connect to a web-based backend. As the doctor walked past patient rooms, he or she would be automatically presented with medical records of nearby patients. In another project, I and my co-authors designed an authentication bracelet that would receive a Kerberos ticket upon login to a modified computer terminal through use of low-energy electrical signals transmitted over the wearer's skin. Upon use of other terminals throughout the hospital, this Kerberos ticket could be sent back over the wearer's skin to the custom contact on the terminal in order to allow the user to login without a password. The bracelet was also designed to immediately lose the cryptographic secret upon being removed from the user.

18. My recent research has focused primarily on embedded systems security with an emphasis on binary analysis, anomaly detection and automated security enforcement. In one case, I designed a security system that can be soldered directly to the CPU of embedded devices in order to perform control-flow integrity for purposes of profile building and enforcement. In another case, I designed a system to automatically discern name and version information from binaries on embedded devices in order to build a software profile of the platform configuration

of the device which can then be cross-referenced with a vulnerability database. I have also written a patent on this specific technique.

19. In 2023 and 2024, I served as a member of the program committee for the IEEE Symposium on Security and Privacy. As a member of this committee, I peer reviewed research that was submitted for publication at the Symposium, including research on cryptographic protocols, memory forensics, cryptographic side channel attacks, cryptographic password vaults, differential privacy, computational theory, memory module design, and hardware security.

4. Patents

20. I am a named inventor on five patents. A full listing of these patents can be found on my CV, attached as Appendix A.

5. Industry

21. I began working in the software industry in 2008, when I obtained my first independent consulting client—Brandeis University. I created a program for their computer repair shop's ticketing system to interface with a label printer. This enabled the computer repair shop to automatically print labels associated with tickets in order to easily identify customers' devices. Shortly after finishing my engagement with Brandeis University, I began working for the Johns Hopkins University Digital Research and Curation Center as a student programmer. There, I created a batch importer for a digital archiving system called DSpace in order to

enable hundreds of digitally archived works per week to automatically be added to the repository. I next ported a web interface for statistics tracking and visualization to the same digital archiving system in order to allow library staff to more easily visualize and understand data about how the digital archiving system was being used through a common web-based interface.

22. In 2009, I began working for a security consulting company called Independent Security Evaluators in Baltimore, Maryland. I worked on a wide variety of security and privacy projects including projects to test and break the DRM scheme of a magazine distribution application for IOS and Android (at the behest of the developer of said application), projects to test implementation code for cryptographic secret splitting, projects to assist with fuzz testing for security vulnerabilities of a variety of software applications, software development projects to automate testing of antivirus and antimalware solutions, and a variety of other security-related projects. Of note, I designed automation systems to create and restore virtualized snapshots of Windows machines on which I tested antivirus software on actual malware samples as part of a project for Consumer Reports. I worked for ISE throughout the semesters and summers until August 2011. In 2010, I also designed a security architecture for a large-scale digital curation system meant to be a major inter-university initiative to create a successor to existing digital curation systems that could be used for decades.

23. In spring 2011, I completed my bachelor's in computer science at Johns Hopkins University, and I also retained an independent consulting client, the University of Michigan, for whom I performed a penetration test and security assessment of a cloud-based research system that they planned to deploy.

24. In February 2016, I joined Harbor Labs full-time as a research scientist. In 2018, I was promoted to Director of Firmware Analysis. In December 2023, I was promoted to Vice President of Applied Research and Technology. In October 2024, the division of Harbor Labs that I managed became Harbor Experts. I became Chief Scientist at Harbor Experts during this time. At Harbor Experts, I manage client engagements related to all of the analysis work that Harbor Experts performs. I also lead Harbor Experts' research projects. A substantial portion of my role at Harbor Experts is in managing source code review engagements. As part of this work, I have reviewed software systems of varying sizes, often totaling in the millions or billions of lines of code. I have reviewed products in the security space, television-based set top boxes, network appliances, numerous web-based enterprise systems, email management systems, telephony products, embedded system bootloaders, social network platforms, virtualization platforms and countless other products. I have also authored whitepapers on source code comparison, source code quality assessment, and the source code review process in litigation.

25. Prior to joining Harbor Experts, I was also the technical and development lead for a firmware security analysis engine for a product developed at Harbor Labs called Firmware IQ. Firmware IQ is designed to analyze the firmware of embedded devices for security vulnerabilities in an automated fashion. In one configuration of the product, a developer uploads a firmware image to a web-based portal which uses a broker to forward the firmware to an engine for analysis. The engine unpacks the firmware, breaks it into its constituent components, and then performs more than a hundred automated security checks in order to find vulnerabilities in the firmware image, including cross-referencing identified software components against NIST's National Vulnerability Database. The results are then reported through JSON to a web-based presentation system which displays them in an interactive fashion.

26. Since Firmware IQ, I have devoted a substantial portion of my time to incorporating lessons learned from analyzing large numbers of firmware images into my own Linux distribution, TechniComp Benchtop Linux. This distribution addresses several gaps in the current market and provides improved stability, reliability, and performance relative to existing Linux distributions.

27. I also have direct experience analyzing cloud infrastructure and cloud-based systems, including systems involving multiple cloud service providers and cross-cloud resource management. My experience includes both the design and

operation of such systems and the analysis of cloud-based systems in security and litigation contexts. For example, in 2011 I performed a security assessment and wide-scale penetration test for the University of Michigan of virtualized cloud-based research systems implemented using Amazon EC2 and VMware vSphere. As part of that work, I developed a formal threat model addressing potential attack vectors and prepared reports documenting the results of my testing.

28. I have also analyzed cloud-based technologies in my work as an expert witness. My engagements have included analysis of cloud-based vehicle-management and security systems and cloud-based content-aggregation and streaming systems. More generally, through the large-scale source code reviews that I have led, I have analyzed numerous web-based enterprise systems and virtualization platforms.

29. My engineering and research experience also encompasses technologies used to implement cloud and distributed systems. I was the technical and development lead for Firmware IQ, a commercial security-analysis platform in which software images submitted through a web-based portal were forwarded through a broker to an analysis engine and the results returned through a web-based presentation system. The platform analyzed, among other things, container images and virtual-machine appliances. I have also developed Hadoop-based distributed data-analysis systems and built distributed computing infrastructure,

and my recent research has included automated vulnerability analysis of containers and virtual appliances. I additionally design and operate virtualization, storage, compute, and network infrastructure used for my security research, artificial-intelligence research, and expert-witness work. This infrastructure includes hypervisor-based virtualized environments, isolated and segmented virtual networks, virtualized storage and network appliances, large-scale network storage, and automated and reproducible test environments. My technical experience with cloud and virtualization technologies includes AWS, Azure, Docker, Hadoop, VMware, KVM, QEMU, LXD, and related systems and tools.

6. Curriculum Vitae

30. A more detailed list of my engagements can be found in my attached curriculum vitae, attached as Appendix A to this declaration.

B. Materials Reviewed

31. My opinions expressed in this declaration are based on documents and materials identified in this declaration, including the ’527 Patent and its prosecution history, the prior art references and background materials discussed in this declaration, and the other references specifically identified in this declaration. I have considered these materials in their entirety, even if only portions are discussed here. The following is an exemplary list of the materials on which I based my opinion.

Exhibit	Description
Ex. 1001	U.S. Patent No. 11,290,527 to Martinez et al. ("'527 Patent")
Ex. 1003	File History of U.S. Patent Application No. 16/916,237 ("'527 Patent File Wrapper")
Ex. 1004	U.S. Patent Application Publication 2017/0063720 A1 ("Foskett")
Ex. 1005	U.S. Patent Application Publication 2018/0276254 A1 ("Whitlock")
Ex. 1006	U.S. Patent Application Publication 2020/0059370 A1 ("Juniper")
Ex. 1007	U.S. Patent Application Publication US 2018/0234459 A1 ("Kung")
Ex. 1008	IEEE Article "Extensible Declarative Management of Cloud Resources across Providers" by O. Serhiienko et al. ("Serhiienko")
Ex. 1009	U.S. Patent 9,444,819 ("Muppidi")
Ex. 1010	U.S. Patent 11,171,994 ("Ramachandran")
Ex. 1011	"New Amazon EC2 Feature: Resource Tagging" (https://aws.amazon.com/blogs/aws/new-amazon-ec2-feature-resource-tagging/)
Ex. 1012	"Tagging Best Practices Implement an Effective AWS Resource Tagging Strategy" (https://d1.awsstatic.com/whitepapers/aws-tagging-best-practices.pdf)
Ex. 1013	"Multi-Cloud Tagging Strategies For the Win" (https://securityboulevard.com/2019/10/multi-cloud-tagging-strategies-for-the-win-2/)
Ex. 1014	"Tag Editor Document History" (https://docs.aws.amazon.com/tag-editor/latest/userguide/doc-history.html)
Ex. 1015	"Microsoft.Resources tags 2019-10-01" (https://learn.microsoft.com/et-ee/azure/templates/microsoft.resources/2019-10-01/tags)
Ex. 1016	"The NIST Definition of Cloud Computing" https://nvlpubs.nist.gov/NISTPUBS/LEGACY/SP/NISTSPECIALPUBLICATION800-145.PDF

Exhibit	Description
Ex. 1017	"AWS vs Azure vs Google: Cloud Wars" (https://www.cbtnuggets.com/blog/certifications/microsoft/aws-vs-azure-vs-google-cloud-wars-2019)
Ex. 1018	"Extensible Declarative Management of Cloud Resources across Providers" (https://ieeexplore.ieee.org/document/8752784)

32. I have also relied on my own experience and expertise in designing and working with web-based platform and distributed server environments, including the interaction between clients, services, and cloud or networked service providers. I have also applied my technical background in system administration and cybersecurity to establish appropriate system controls, security configurations, and access restrictions intended for safeguarding systems and data in organizations.

C. Level of Ordinary Skill in the Art

33. I am not an attorney and offer no legal opinions. I have been informed about certain aspects of the law for purposes of my analyses and opinions.

34. I understand that in analyzing questions of invalidity, the perspective of a person having ordinary skill in the art ("POSITA") is often implicated, and the Court may need assistance in determining that level of skill.

35. I understand that the claims and written description of a patent must be understood from the perspective of a POSITA. I have been informed that the following factors may affect the level of skill of a POSITA: (1) the educational

level of the inventor; (2) the type of problems encountered in the art; (3) the prior-art solutions to those problems; (4) the pace of innovation; (5) the sophistication of the technology; and (6) the educational level of active workers in the field. A person of ordinary skill in the art is also a person of ordinary creativity in the art.

36. Based on my experience, as well as my reading of the ’527 Patent, it is my opinion that a person of ordinary skill with respect to the subject matter of the ’527 Patent at the time of the alleged priority date of the ’527 Patent in June 2020 would have had at least a B.S. degree in computer science, computer engineering, or electrical engineering (or equivalent experience) and would have had at least two years of practical experience with network and data security, including the then-current technologies such as cloud infrastructure management, cloud resource organization, and/or virtual server or resource administration. This definition is flexible, and additional educational experience in computer science could make up for less work experience and vice versa.

37. I am at least a person of ordinary skill in the art and was so on the date to which the ’527 Patent claims priority. As shown by my qualifications and my curriculum vitae attached as Appendix A, I am aware of the knowledge and skill possessed by a person of ordinary skill in the art at the time of the priority date of the ’527 Patent. In performing my analysis, I have applied the standard set forth above.

D. Summary of Opinions

38. I have reviewed and analyzed the '527 Patent.

39. Based on my review and analysis, it is my opinion that claims 1-20 of the '527 Patent are invalid based on the following grounds:

Ground	Basis	Reference(s)	Challenged Claims
1	§ 103	Foskett	1, 3, 4, 6, 7, 8, 10, 11, 13, 14, 15, 17, and 20
2	§ 103	Foskett and Whitlock	2, 5, 9, 12, 16, and 19
3	§ 103	Foskett and Juniper	18
4	§ 103	Kung and Serhiienko	1, 3, 4, 6, 7, 8,10, 11, 13, 14, 15, 17, and 20
5	§ 103	Kung, Serhiienko, and Whitlock	2, 5, 9, 12, 16, and 19
6	§ 103	Kung, Serhiienko, and Juniper	18

II. OVERVIEW OF THE TECHNOLOGY

A. Priority Date of the Claims

40. I have been informed that a U.S. patent application may claim the benefit of the filing date of an earlier patent application if the earlier patent application sufficiently disclosed the invention claimed in the later-filed U.S. patent application. I have also been informed that priority is determined on a claim-by-claim basis so that certain claims of a patent may be entitled to the priority date of an earlier-filed patent application even if other claims of the same patent are not entitled to that priority date.

41. I have also been informed that a patented claim is invalid if the claimed invention was patented, described in a printed publication, or in public use, on sale, or otherwise available to the public before the effective filing date of the claimed invention, or the claimed invention was described in an issued patent or in an application for patent published or deemed published and that was effectively filed before the effective filing date of the claimed invention.

42. I understand that the claims of the ’527 Patent are entitled to a priority date of June 30, 2020, which is the date on which the application for the ’527 Patent was filed.

B. Overview of Relevant Technology

1. Cloud Resource Tagging

43. Cloud computing refers to on-demand access to a shared pool of configurable computing resources. It has been a widely understood delivery model since at least 2009, when National Institute of Standards and Technology (NIST) published its foundational working definition covering its core characteristics, service models (e.g., IaaS, PaaS, SaaS), and deployment models (e.g., public, private, community, and hybrid). (Ex. 1009, 7:13-8:38.)

44. As cloud computing developed, so did the practice of tagging: attaching key-value metadata labels to cloud resources for classification and identification. Before the ’527 Patent, tagging was already a commonly established

approach for governing cloud resources: "known tagging solutions have been used to facilitate asset management," and tags could be applied "manually (through a configuration tool), via an automated set of tag rules (rule-based tagging), statically or dynamically, or programmatically." (*Id.*, 10:66–11:5.)

45. As early as 2010, Amazon Web Services introduced a resource-tagging feature for Amazon EC2 allowing customers to attach up to ten key-value pairs to cloud resources such as instances, machine images, and storage volumes, and to use those tags to filter and organize resources through an API or management console, confirming that tagging cloud resources with key-value metadata was already a well-known technique roughly a decade before the '527 Patent. (https://aws.amazon.com/blogs/aws/new-amazon-ec2-feature-resource-tagging/) (Ex. 1011.)

46. In 2018, Amazon Web Services released a whitepaper to help users develop a tagging strategy to manage AWS resources more effectively. The whitepaper defined tags, disclosed various tagging use cases, and shared best practices for identifying tag requirements, naming tags and resources, cost allocation tags, and tag governance and data management. (https://d1.awsstatic.com/whitepapers/aws-tagging-best-practices.pdf) (Ex. 1012.)

47. By 2019, with cloud resources and services proliferating, organizations began to manage disparate resources and services—"spread across multiple cloud providers and accounts/subscriptions"—to "simplify and automate critical tasks across complex multi-cloud environments with relative ease," with a "global tagging policy" that used "user-defined data" to "identify and describe the resources running across one more cloud environments." (https://securityboulevard.com/2019/10/multi-cloud-tagging-strategies-for-the-win-2/) (Ex. 1013.)

48. Tagging provides a structured way to organize and manage distributed computing resources while supporting administrative tasks such as monitoring, cost allocation, access management, and system configuration, and is therefore widely used to manage cloud resources. As an example, by 2020, resource tagging had become an established practice in cloud infrastructure management, with major platforms such as AWS and Microsoft Azure providing dedicated mechanisms for assigning metadata to cloud resources, organizing infrastructure, and identifying resources according to characteristics such as environment, ownership, purpose, and operational function. (https://docs.aws.amazon.com/tag-editor/latest/userguide/doc-history.html (Ex. 1014.); https://learn.microsoft.com/et-ee/azure/templates/microsoft.resources/2019-10-01/tags) (Ex. 1015.)

49. The development of cloud resource tagging can be understood within the broader transition toward increasingly distributed, service-based computing. As these approaches evolved, cloud services provided an infrastructure model in which servers, storage, networks, applications, and other computing resources could be provisioned and managed dynamically over a network. NIST's 2011 definition of cloud computing specifically describes configurable computing resources that can be rapidly provisioned and released, reflecting the distributed and elastic characteristics of modern service environments. (https://nvlpubs.nist.gov/NISTPUBS/LEGACY/SP/NISTSPECIALPUBLICATION800-145.PDF) (Ex. 1016.)

50. Within these types of cloud-based environments, tagging provided administrators with a method for identifying and organizing collections of cloud resources. Tags could associate resources with applications, owners, projects, environments, or operational functions, making them useful for administration, monitoring, cost allocation, automation, and access-control practices. Microsoft Azure documentation, for example, describes resource tagging as a means of "organi[zing] your Azure resources and management hierarchy." (https://learn.microsoft.com/et-ee/azure/templates/microsoft.resources/2019-10-01/tags) (Ex. 1015.) Consequently, tagging can be viewed as part of a broader approach to infrastructure and configuration management in distributed web

environments, helping administrators maintain organizational structure and apply consistent operational and security controls across dynamically provisioned resources.

2. Tags to Govern and Secure Cloud Resources

51. Using tags as the basis for security policy enforcement was also well-established. Technologies such as "network functions virtualization ('NFV'), software defined networking ('SDN'), and other 'cloud' computing architectures and technologies" enabled operators to deploy services on commercial off-the-shelf hardware instead of custom hardware. While virtualization may provide flexibility to network operators and other entities, "several challenges pose difficulties in migrating services to virtualized networks." (Ex. 1010, 1:37-38.) For example, it became a recognized need to create and apply security policies automatically to cloud resources as they are instantiated, scaled, and updated. To fulfill this need, methods were developed for tag-based security policy creation.

52. Compliance-driven security management using cloud resource metadata was likewise standard practice, with audit logs maintained in relational databases to meet well-known regulatory requirements such as the Sarbanes-Oxley (SOX) Act and the Health Insurance Portability and Accountability Act (HIPAA) long before the '527 Patent. Organizations rely on audit logs to demonstrate compliance with regulations and internal policies, noting that "compliance with

these and other regulations may be shown using information contained in audit logs" and that audit logs are "typically made available in relational databases to allow easy querying of the information by reporting programs or software." (Ex. 1009, 1:13–38.) More specifically, organizations need to "obtain visibility into the cloud resources they are managing." (*Id.*, 2:13-16.) Such visibility "needs to be provided at various contexts, e.g., an identity context (who did what), a resource ownership context (who owns what resource), and an environment/workload context (what applications or infrastructure were used)." (*Id.*, 2:25-29.) Stored compliance information (e.g., logs) retrieved from a database is therefore a well-applied mechanism for implementing such visibility.

C. The '527 Patent

1. Overview

53. The '527 Patent is directed to "[s]ystems and methods for automatically tagging cloud resources that are spread across multiple cloud platforms." (Ex. 1001, Abstract.)

54. Figure 1 in the patent (below) depicts a network architecture. It includes a security manager (102) with an orchestrator (120) that is connected to private cloud (110), public cloud (112), hybrid cloud (114), and private network (114). (*Id.*, 6:1–21) Private network (114) has computing devices (116-1 - 116-N).

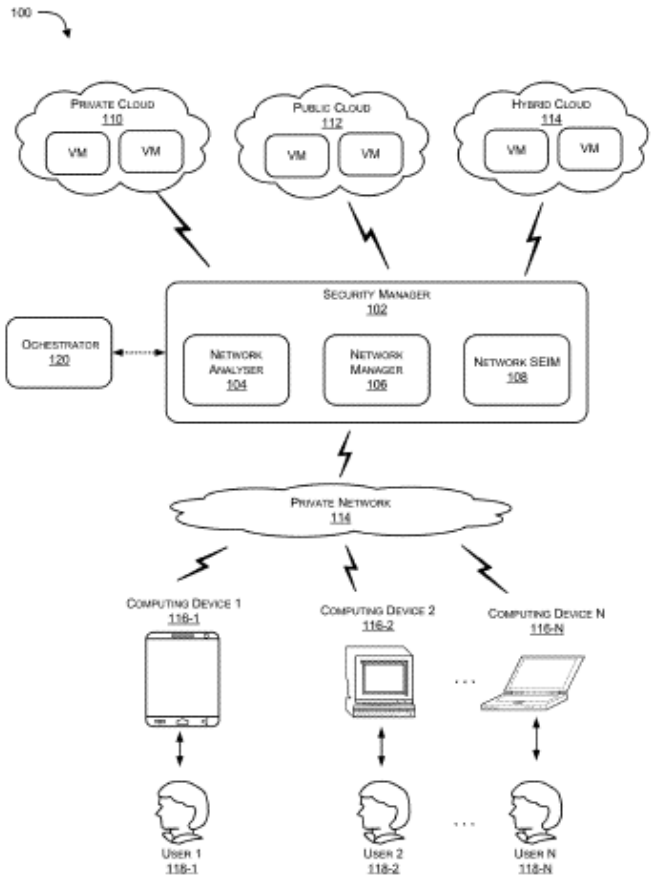


FIG. 1

55. The patent discloses a process flow for automatically tagging cloud resources (e.g., servers or load balancers) from different cloud providers, exemplified in Figure 4 of the patent (below). There, the cloud-tagging orchestrator service ("CTOS") receives information about each cloud provider (block 402), creates a unified tag for each cloud resource based on a pre-defined global tagging policy (block 404), and associates the tag with cloud resource in a storage database, (block 406). The security manager then retrieves tags from the database so the appropriate security policy can be applied (block 408).

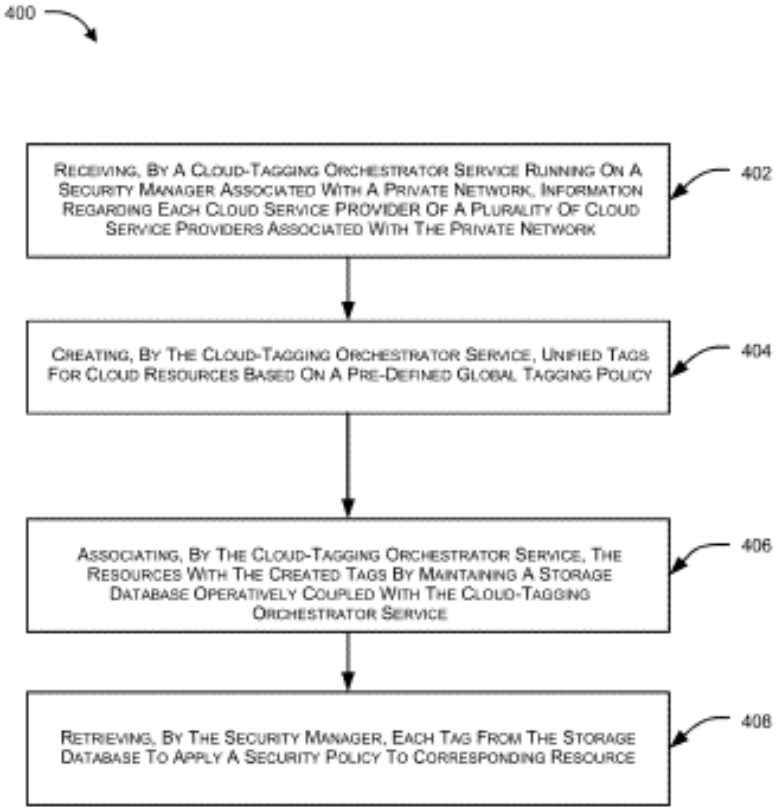


FIG. 4

2. The CTOS and Security Manager Architecture

56. The ’527 Patent describes the security manager (102) to include a range of security-related implementations, encompassing physical, on-premises hardware, software-based systems, and cloud-hosted implementations. In one embodiment, “security manager 102 is an on-premises management appliance associated with network 114 that manages computing devices 116 associated with

network 114." (*Id.*, 6:22–25.) In another embodiment, "security manager 102 is a cloud based security manager that can be implemented in the form of a cloud service (e.g., by private cloud 110, public cloud 112, and/or hybrid cloud 114)." (*Id.*, 6:25–28.) The '527 Patent further describes that in yet another embodiment, "orchestrator 120 acts as a means for communication for security manager 102 such that all cloud environments and computing devices interact with orchestrator 120 in order to interact with security manager 102," and that "[r]egardless of location, security manager 102 or orchestrator 120 can maintain connectivity with each of network appliances of network 114." (*Id.*, 6:28–35.) The '527 Patent also describes that "Depending upon the particular implementation orchestrator 120 may be part of security manager 102 or may be an independent service used by security manager 102." (*Id.*, 7:6–8.)

57. Figure 2 in the patent (below), illustrates the functional components of security manager (102), including processor(s) (202), memory (204), interface(s) (206), a processing engine (208) comprising an information receiving engine (212), a tagging engine (214), a remediation engine (216), other unit(s) (218), and a database (210). (*Id.*, Fig. 2; 8:48–57; 9:15-42.)

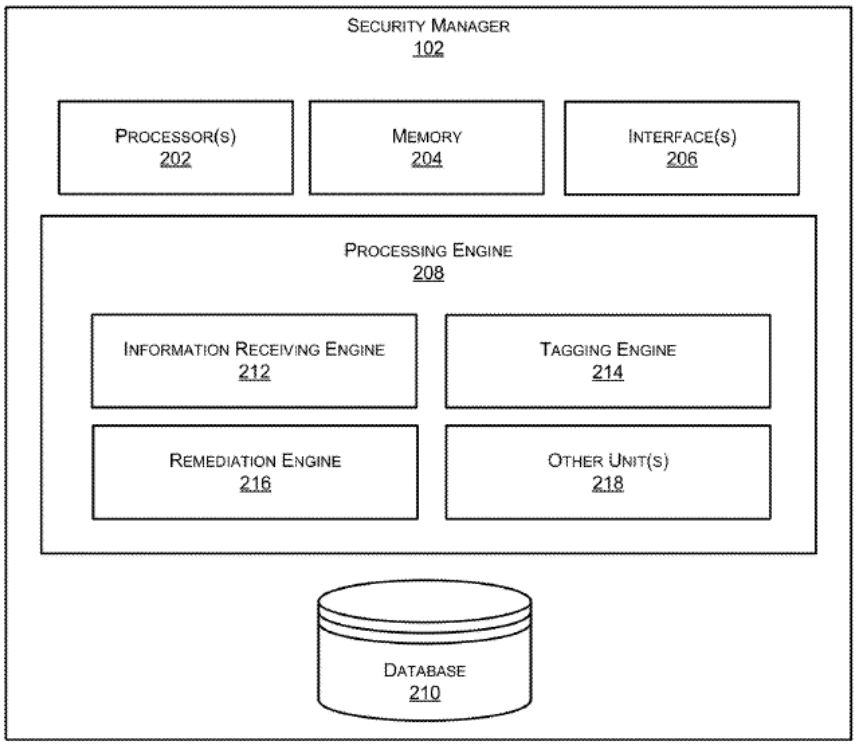


FIG. 2

58. A POSITA would have understood that the information receiving engine (212) and tagging engine (214)—working in conjunction with other components, such as database (210) and other unit(s) (218)—collectively implement the claimed CTOS. In addition, the security manager may embody additional components that fulfill broader security-management functions offered specifically by the security manager. An example is the remediation engine (216) (as seen above in Figure 2 of the patent), which retrieves those tags from database (210) to apply a security policy to the corresponding resource (as depicted at block

408 of Figure 4). Figure 2 in the patent also teaches other units (218) on the security manager. (*Id*., Fig. 2)

D. The Challenged Claims

59. I understand that the Petitioner is challenging claims 1-20 in the ’527 Patent.

E. Claim Construction

60. I understand that claim terms generally are construed in accordance with the ordinary and customary meaning they would have to a POSITA at the time of the invention in light of the claim language, the specification, and the prosecution history. I understand that dictionaries and other extrinsic evidence may be considered as well, though such evidence is typically regarded as less significant than the intrinsic record in determining the meaning of the claim language.

61. For all terms of the challenged claims of the ’527 Patent, I have afforded them their ordinary and customary meaning in light of the intrinsic evidence at the time of the invention on June 30, 2020.

III. UNPATENTABILITY OF THE '527 PATENT CLAIMS

A. Legal Standards

1. Obviousness

62. I am informed and understand that a patent cannot be properly granted for subject matter that would have been obvious to a person of ordinary skill in the art at the time of the alleged invention, and that a patent claim directed to such obvious subject matter is invalid under 35 U.S.C. § 103. It is also my understanding that in assessing the obviousness of claimed subject matter, one should evaluate obviousness in light of the prior art from the perspective of a person having ordinary skill in the art at the time the alleged invention was made (and not from the perspective of either a layman or a genius in that art). It is my further understanding that the question of obviousness is to be determined based on:

- The scope and content of the prior art;
- The difference or differences between the subject matter of the claim and the prior art (whereby in assessing the possibility of obviousness one should consider the manner in which a patentee and/or a Court has construed the scope of a claim);
- The level of ordinary skill in the art at the time of the alleged invention of the subject matter of the claim; and

- Any relevant objective factors (the "secondary indicia") indicating nonobviousness, including evidence of any of the following: commercial success of the products or methods covered by the patent claims; a long-felt need for the alleged invention; failed attempts by others to make the alleged invention; copying of the alleged invention by others in the field; unexpected results achieved by the alleged invention; praise of the alleged invention by the alleged infringer or others in the field; the taking of licenses under the patent by others and the nature of those licenses; expressions of surprise by experts and those skilled in the art at the subject matter of the claim; and whether the patentee proceeded contrary to accepted wisdom of the prior art.

- Any relevant objective factors (the "secondary indicia") indicating obviousness: independent invention of the claimed invention by others before or at about the same time as the named inventor thought of it; and other evidence tending to show obviousness.

B. Ground 1: Foskett renders claims 1, 3, 4, 6, 7, 8, 10, 11, 13, 14, 15, 17, and 20 obvious.

63. ***Foskett***: Five years before the '527 Patent, Foskett disclosed a system to apply uniform tags across multiple cloud service providers. (Ex. 1004, ¶ 0015.) Foskett disclosed that "[p]rovisioning resources into the cloud is a constantly increasing technical challenge as more cloud service providers emerge, each offering disparate computing platforms, services, assets, supported technical

components, and other features. A multi-cloud network proxy controls and normalizes the tagging data applicable across the multiple cloud service providers." (*Id.*, Abstract.) Architecturally, Foskett centered on a Multi-Cloud Network Proxy ("MNP") positioned between cloud service providers and a requester. (*Id.*, ¶¶ 0015, 0031.) The MNP had a metadata tagging repository (114) and proxy circuitry (116) that stored and maintained tag identifiers and values. (*Id.*, ¶¶ 0033–0034, 0040.) The MNP accounted for heterogeneous environments where some cloud providers had native tagging repositories and others did not. (*Id.*)

64. Figure 3A from the '527 Patent (below on left) and Figure 1 from Foskett (below on right) show striking architectural similarities. Specifically, a security manager associated with an orchestrator in the '527 Patent sits between multiple clouds that host virtualization infrastructure. The orchestrator manages tagging of cloud resources. This is analogous the architecture of Foskett in which the MNP sits between multiple networks containing virtualization infrastructure and provides the services that the '527's orchestrator running on the security manager provides.

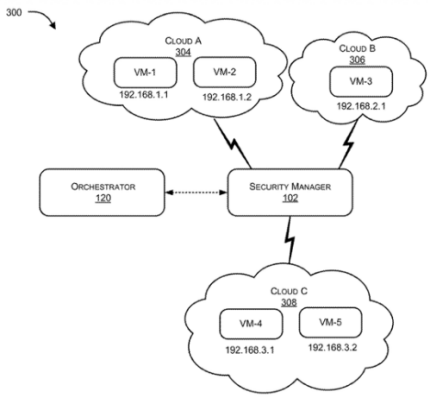


FIG. 3A

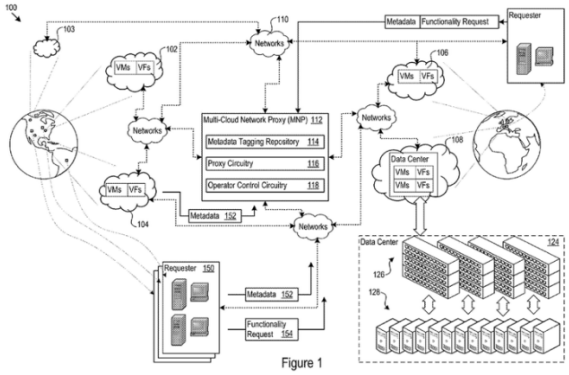


Figure 1

65. The similarities are further borne out in Figure 3B from '527 Patent (below on left) as compared with Figure 6 from Foskett (below on right). "Universal tags" in Figure 3B of the '527 Patent are organized in a table according to fields named "IP Address" and "Cloud Services." Tag "content" values in Figure 6 of Foskett are organized by provider and tag identifier. In Foskett, the provider column is analogous to the "cloud services" column in the '527. The "tag identifier" column in Foskett is a more generalized form of the "IP Address" column in the '527. A POSITA would understand that an IP address is one possible type of tag identifier.

IP Address	Cloud Services	Universal Tag
192.168.1.1	A	CDA11
192.168.1.2	A	CDA12
192.168.2.1	B	CDB21
192.168.3.1	C	CDC31
192.168.3.2	C	CDC32

FIG. 3B

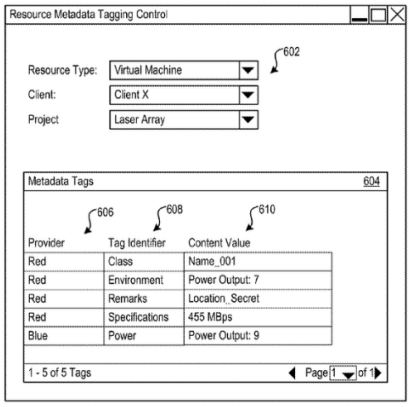


Figure 6

66. Process-wise, the MNP discovered resources and tags by connecting to service provider application programming interfaces (APIs) and sending "discovery requests" to each "target hosting platform" to obtain identifiers of deployed resources. (*Id.*, ¶ 0035.) A "target hosting platform" refers to the specific cloud platform offered by a given service provider (e.g., Amazon Web Services, Google Compute Engine, or Microsoft Azure) to which the MNP connected in order to manage and normalize tags across resources. (*Id.*, ¶ 0018.) Figure 4 in Foskett (below) illustrates the MNP's process.

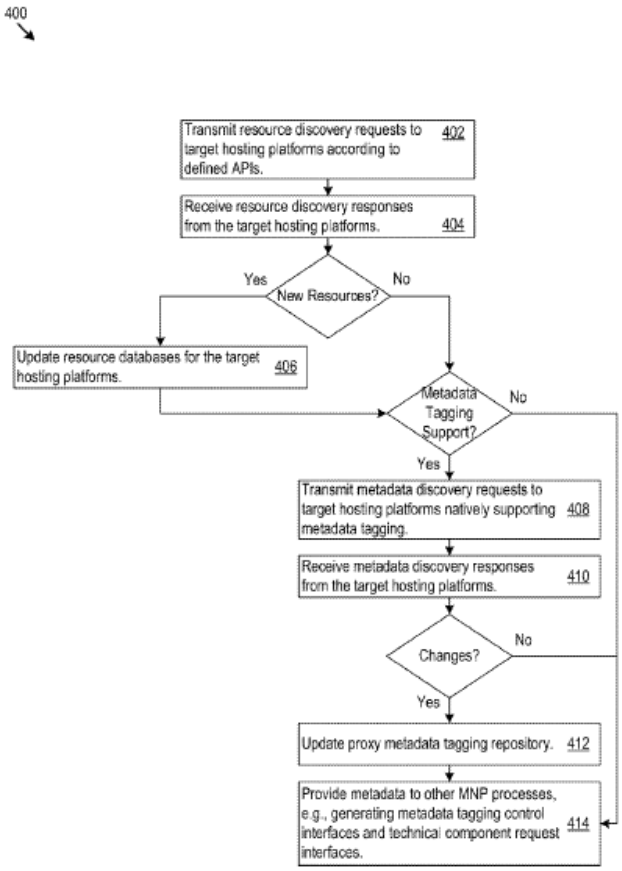


Figure 4

67. Foskett clarified that service providers may also be on-premises. "Service providers may additionally or alternatively provide computing resources in 'on-premises' hosting platforms, which typically refers to a location with increased privacy and security compared to public cloud resources." (*Id.*, ¶ 0018.) An on-premise location may be within a secure facility owned by a resource

requester which has moved computing functionality to a cloud based implementation, for instance. (*Id.*)

68. Additionally, the MNP could transmit a metadata discovery request to the provider's metadata tagging repository to obtain metadata tagging information. The metadata tagging information includes tag identifiers and tag content. (*Id.*, ¶¶ 0032-0035.) The MNP stored the response in its resource database to manage discovered resources and remove deleted resources based on the current state of the resources provided in the resource discovery responses. (*Id.*, ¶ 0034.) When the MNP needed to update tags in a target hosting platform, whether triggered by a manual operator request, a pre-scheduled interval, or a specific event condition, it established an API connection to that platform and transmitted the metadata tags and associated metadata content from its proxy metadata tagging repository to the target hosting platform for synchronization. (*Id.*, ¶¶ 0051, 0055–0063.) In other words, the MNP transmitted the uniform tag from the proxy repository back to each cloud service provider that natively supports tagging. (*Id.*)

69. Because cloud providers used inconsistent tag naming conventions and value formats, Foskett's MNP maintained a proxy metadata tagging repository as the authoritative, provider-neutral baseline for all tags across every connected provider. (*Id.*, ¶ 0015.) A uniform tag was a tag identifier and content value standardized against type definitions and constraint definitions specifying

permissible formats and valid values. (*Id.*, ¶¶ 0047–0048.) The MNP applied the normalized tag by validating and constraining tag content against those definitions before storing it in the proxy repository, then transmitted the standardized identifiers and content back to each target hosting platform via provider API calls to synchronize each provider's repository to that baseline. (*Id.*, ¶¶ 0050–0051, 0054–0055; Fig. 9, steps 916–918.)

70. Foskett disclosed multiple figures and process flows for its MNP that represent variations of the same invention. A POSITA would have been motivated to combine features from Foskett's different figures because Foskett itself explained that Figures 1 and 2 show only "one of many possible different implementation contexts" and that its "technical solutions are not limited in their application to the architectures and systems shown in FIGS. 1 and 2, but are applicable to many other cloud computing implementations, architectures, and connectivity," and further confirmed that "[v]arious implementations have been specifically described" but "many other implementations are also possible." (*Id.*, ¶¶ 0016, 0069.)

1. CLAIM 1:

Element 1[Preamble]: "A method comprising:"

71. Foskett disclosed a method for centrally managing and normalizing metadata tags across multiple cloud service providers. The MNP "provides a

central control mechanism for discovering, monitoring, and updating complex tagging data applied to the technical components provisioned with the cloud service providers, regardless of whether any given service provider natively supports metadata tagging." (*Id.*, ¶ 0015.)

Element 1[a]: "receiving, by a cloud-tagging orchestrator service running on a security manager associated with a private network, information regarding each cloud provider of a plurality of cloud providers associated with a cloud environment used by the private network"

72. Foskett disclosed or rendered obvious this element because the MNP's tagging software (CTOS) runs on the MNP's infrastructure (security manager), which is tied to a resource requester (associated with a private network). That software connects to each of several named cloud providers (plurality of cloud providers) through that provider's own API and receives back (receiving) a discovery response enumerating that provider's resources (information regarding each cloud provider). Those providers, taken together, make up the set of platforms the MNP manages for the requester (a cloud environment used by the private network).

73. Figure 2 in Foskett (below) depicted the internal architecture of the MNP 112, showing communication interfaces 202, system circuitry 204, I/O interfaces 206, and display circuitry 208, with system circuitry 204 further comprising one or more processors 218 and memory 220 storing control

instructions 222 and an operating system 224. (*Id.*, Fig. 2, ¶¶ 0024–0026.) Foskett explained that "[t]he system circuitry 204 may include any combination of hardware, software, firmware, or other logic," and "may be implemented, for example, with one or more systems on a chip (SoC), application specific integrated circuits (ASIC), microprocessors, discrete analog and digital circuits, and other circuitry." (*Id.*, ¶ 0026.) It is this infrastructure of the MNP's system circuitry 204, communication interfaces 202, and I/O interfaces 206, together with the associated supporting software that operates and manages that infrastructure, such as the operating system 224 and any hypervisor, database infrastructure, or other system-level software used to manage and control access to the underlying hardware resources that corresponds to the claimed security manager. In other words, the infrastructure and its supporting system-level software that together house and operate everything the MNP does are just like the '527 Patent's security manager. (Ex. 1001, 6:22–28.)

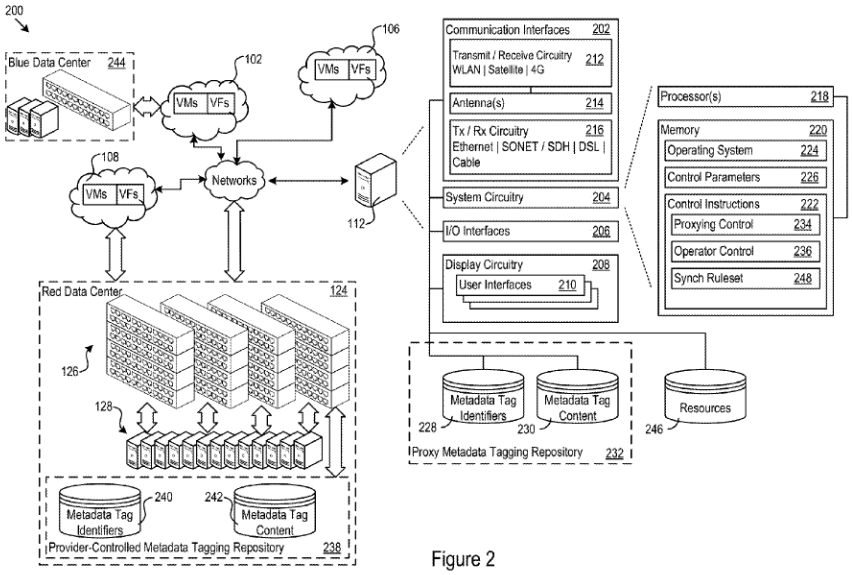


Figure 2

74. The MNP then has software that performs the tagging work, like Foskett's control instructions 222 and central proxying control 234. This tagging-specific software corresponds to the CTOS. Foskett explained that "[t]he system circuitry 204 may implement the proxy circuitry 116 and the operator control circuitry 118, e.g., as control instructions 222 executed by the processor 218," and that "[t]he control instructions 222 implement a central proxying control 234 for discovering, monitoring, updating, and proxying complex tagging data applied to the technical components provisioned with the cloud service providers," as well as "an operator control 236 that dynamically generates technical component request interfaces that include specifically prioritized metadata tags." (Ex. 1004, ¶ 0028.)

75. In other words, the MNP's infrastructure (including its supporting system software such as the operating system 224) is the claimed security manager

while the MNP's tagging specific application such as the control instructions 222 and central proxying control 234 is the claimed CTOS. The MNP's tagging software ***runs*** on the MNP's infrastructure and carries out the discovering, tagging, and proxying functions discussed throughout Foskett's specification.

76. The MNP is also "associated with a private network" because the MNP sits between a resource requester and the various cloud providers that host that requester's resources, acting as the requester's own centralized point of control. (*Id.*, Fig. 1.) Figure 1 in Foskett (below) illustrates how the MNP is associated with various networks.

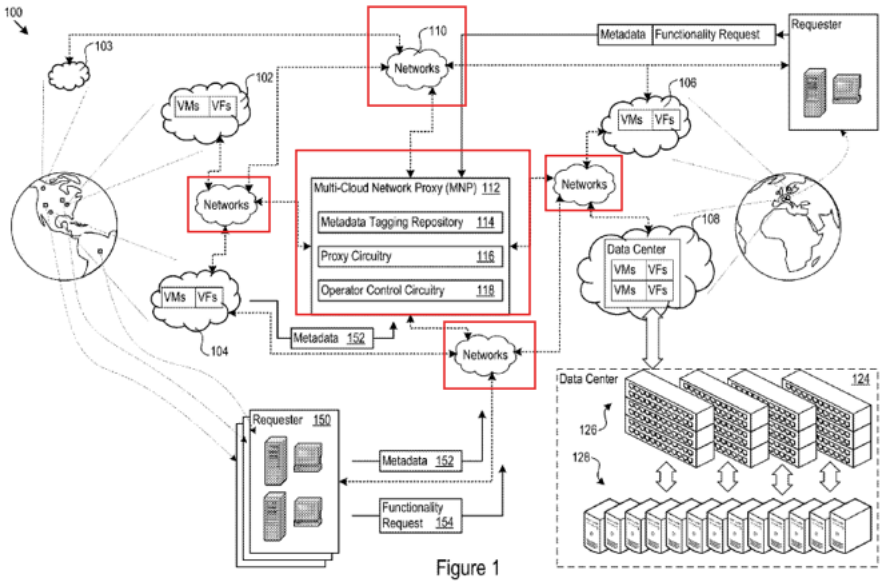


Figure 1

77. Foskett disclosed that discovery by the MNP can include "obtaining access credentials 326, e.g. logon username/password from an account owner at the resource requester 150" and "[t]he MNP 112 submits the access credentials to

the corresponding service provider to access and request the resources and attached metadata (if any) for the resources." (*Id.*, ¶ 0033.) A POSITA reading Foskett would have understood the "requester" referenced throughout Foskett's specification is the client-side entity requesting a resource from a cloud provider, like an enterprise user of the MNP's tagging and provisioning services. The MNP then obtained access credentials from the resource requester to submit to the service provider, effectively allowing the MNP to make provisioning requests on the requester's behalf and for its private network. The MNP is associated with the private network of that enterprise, organization, or other end-user.

78. The MNP receives information about each cloud provider from each one. First, the MNP reaches out to each cloud provider through that provider's own API: "[t]he MNP 112 may, for instance, connect through service provider application programming interfaces (APIs) 302, 304, and 306 to transmit discovery request messages 308, 310, 312 to the target virtual hosting platforms 314, 316, and 318 of the service providers 102, 106, and 108 respectively." (*Id.*, ¶ 0032.) Second, each provider responds back to the MNP: "[t]he MNP 112 receives discovery responses 320, 322, and 324 enumerating the resources as a result of transmitting the discovery request messages." (*Id.*) Figure 3 in Foskett (below) illustrated this exchange, showing the MNP 112 connected to each of the three

target virtual hosting platforms through their respective provider APIs, with discovery requests flowing out and discovery responses flowing back.

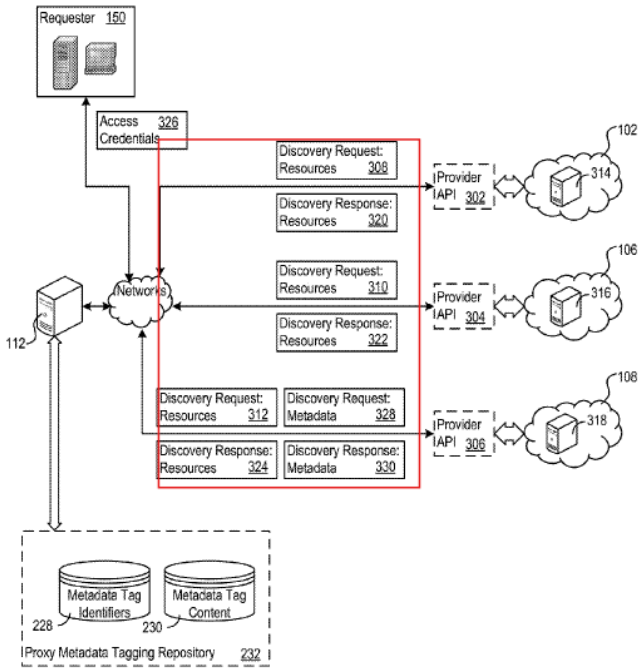


Figure 3

79. Foskett also applied to a "plurality of cloud providers." Foskett disclosed that the MNP is designed from the outset to interact with more than one cloud provider at a time, and names several real-world examples: "[e]xamples of service providers include Amazon, Google, Microsoft, and Accenture, who offer different target hosting platforms, e.g., Amazon Web Services (AWS), Google Compute Engine (GCE), Microsoft Azure (Azure), Accenture Cloud Platform (ACP), and Windows Azure Pack (WAP) for on-premise cloud implementations."

(Ex. 1004, ¶ 0018; 0019 ("A multi-cloud network proxy (MNP) 112 makes complex cloud architectural provisioning and execution decisions across multiple cloud services, taking into account the global network architecture 100, the various service provider locations and capabilities, and other factors.").) Foskett referred to each of these providers' own hosting environments as a "target hosting platform."

80. Additionally, Figure 3 in Foskett (below) disclosed that the MNP contacted several different providers simultaneously, "connect[ing] through service provider application programming interfaces (APIs) 302, 304, and 306." (*Id.*, ¶ 0032.)

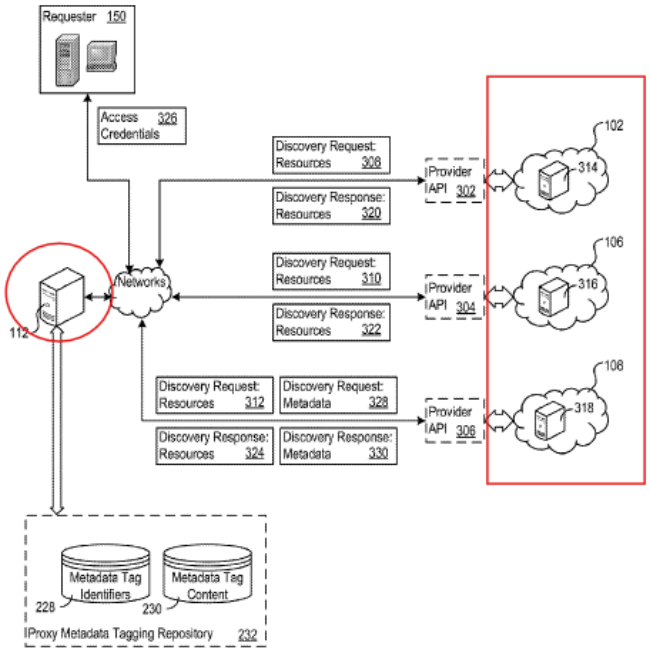


Figure 3

81. These service providers were "associated with a cloud environment used by the private network." Foskett disclosed "cloud environment" as the collection of target hosting platforms, taken together, that the MNP centrally manages on behalf of a given resource requester. Figure 1 in Foskett (below) depicts the MNP 112 positioned between a requester (150) and multiple service providers (102, 104, 106, and 108), with the MNP serving as the requester's unified point of contact across that entire collection of providers. (*Id.*, ¶¶ 0019–0022, Fig. 1.) As noted above, the requester has a username and password for each corresponding service provider that it gives to the MNP so the MNP can interface with the service providers on its behalf. (Fosket, ¶ 0033.) A POSITA would have understood this composite set of target hosting platforms, collectively provisioned, reachable and managed through the MNP on the requester's behalf, are "associated with a cloud environment used by the private network."

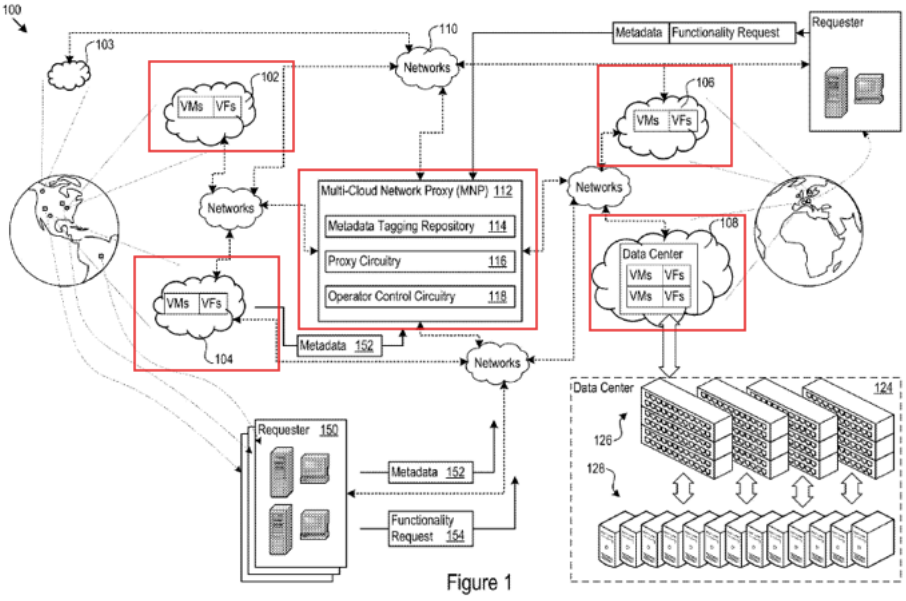


Figure 1

82. Accordingly, Foskett disclosed or rendered obvious all elements of limitation 1[a].

Element 1[b][i]: "based on the received information, for each cloud resource of a plurality of cloud resources hosted by the plurality of cloud providers on behalf of the private network"

83. Foskett disclosed or rendered obvious this element because, using the information it already gathered about each provider (based on the received information), the MNP tracked the virtual machines and virtual functions (cloud resources) provisioned at each provider's platform (hosted by the plurality of cloud providers), all for the benefit of the same requester whose credentials and provisioning requests the MNP handled (on behalf of the private network).

84. Foskett used "data center" to mean the physical infrastructure of a specific cloud provider's target hosting platform (i.e., the actual servers and

equipment where that provider's virtual machines and other resources run), as illustrated by Foskett's own example of "a target hosting platform implemented by a data center 124 for the service provider 108." (*Id*., ¶ 0022.) Foskett explained that a data center "supports running many different virtual machines (VMs), each potentially running many different virtual functions (VFs)." (*Id.*) Foskett further disclosed that "technical components, such as types and assets, provisioned into a region hosted by a resource provider or service provider may be referred to as 'resources.'" (*Id.*, ¶ 0020.) Technical components can be "e.g., VMs, networks, disk images, and a wide range of other types and assets, provisioned in any of the target hosting platforms." (*Id.*) Accordingly, the VMs and VFs discussed above are the technical components that Foskett refers to as resources once provisioned into a target hosting platform.

85. In addition, Foskett specified that such resources could be cloud resources hosted via cloud service providers, such that "[t]he service providers may provide computing resources via hosting platforms that are generally publicly available … Examples of service providers include Amazon, Google, Microsoft, and Accenture, who offer different target hosting platforms, e.g., Amazon Web Services (AWS), Google Compute Engine (GCE), Microsoft Azure (Azure), Accenture Cloud Platform (ACP), and Windows Azure Pack (WAP)." (*Id.*, ¶

0018.) Foskett also specified that such resources may be "on-premise" in addition to cloud resources. (*Id.*)

86. Figure 1 in Foskett (below) depicted VMs and VFs hosted within each of the illustrated service providers (102, 104, 106, 108), confirming that resources of this kind sit inside the target hosting platforms.

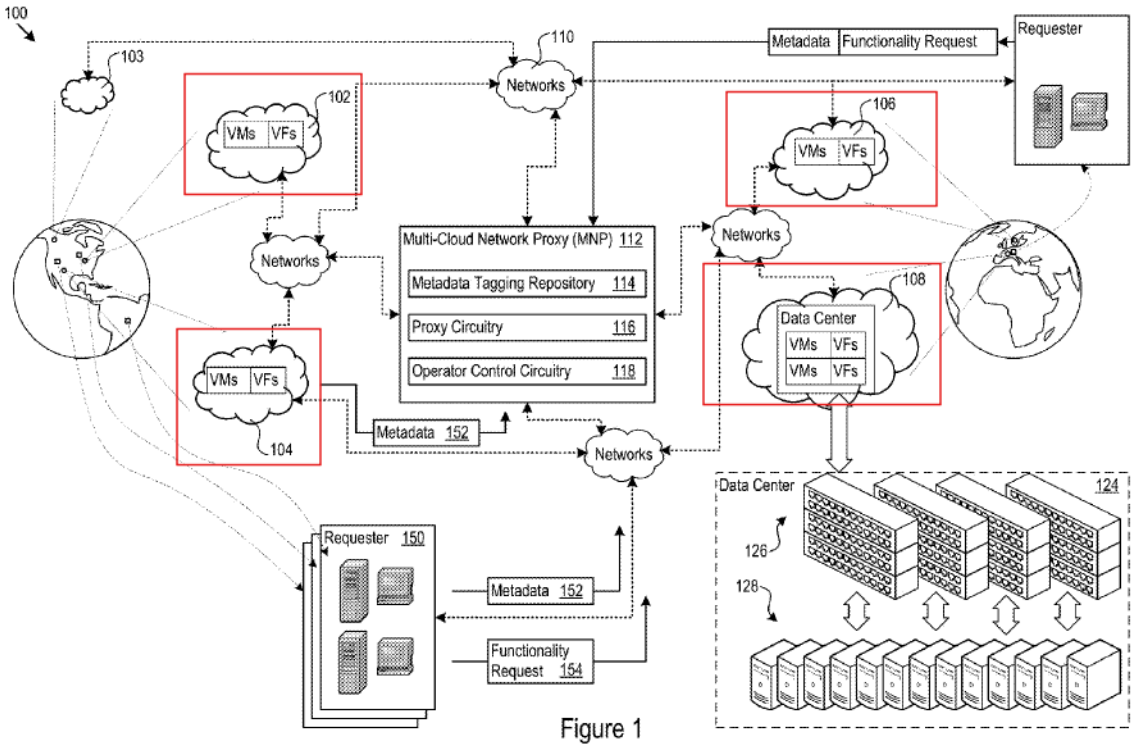


Figure 1

87. As discussed above in connection with Element 1[a], the "requester" referenced throughout Foskett's specification is an end-user, like the enterprise user on whose behalf the MNP operates. Foskett confirmed that resources are provisioned at the requester's own initiative and for the requester's own benefit:

"[t]he resource requester 150 interacts with the request interface 1000 to submit a request for a new resource to provision to a target hosting environment," and the MNP acts "[i]n response to a resource requester 150 requesting provisioning of a new resource." (*Id.*, ¶¶ 0058–0059.) Because cloud-based resources (like these VM and VF resources) are provisioned into a target hosting platform in order to serve that same requester, Foskett disclosed resources "hosted by the plurality of cloud providers on behalf of the private network."

88. Accordingly, Foskett disclosed or rendered obvious all elements of limitation 1[b][i].

Element 1[b][ii]: "retrieving, by the cloud-tagging orchestrator service, information associated with the cloud resource"

89. Foskett disclosed or rendered obvious this element because the MNP's tagging software (CTOS) separately queried the provider's own metadata tagging repository and obtained (retrieving) that resource's existing tag identifiers and content (information associated with the cloud resource).

90. As discussed above in connection with Element 1[a], "[t]he discovery messages specify a request for the target hosting platform to return identifiers of each resource deployed in the target virtual hosting platform," and the "MNP 112 receives discovery responses ... enumerating the resources." (*Id.*, ¶ 0032.) That step tells the MNP which resources exist through their identifiers. The MNP then

"updates its resource database 246 according to the resource discovery responses. For instance, the MNP 112 may add newly discovered resources and remove deleted resources, according to the current state provided in the resource discovery responses." (*Id.*, ¶ 0034.)

91. Next, the MNP retrieves additional information about each resource. As Foskett explained, "the MNP 112 also engages in metadata discovery for the resources provisioned in the target hosting platforms," transmitting "metadata discovery requests 328 to the target hosting platforms" and receiving "metadata discovery responses 330" containing "metadata tag identifiers and metadata tag content stored remotely in the provider-controlled metadata tagging repository." (*Id.*, ¶ 0035.) This metadata discovery request is a distinct, subsequent request from the initial resource discovery request.

92. Figures 3 and 4 in Foskett (below) also disclosed the MNP retrieving information associated with each cloud resource (e.g., identifier or metadata).

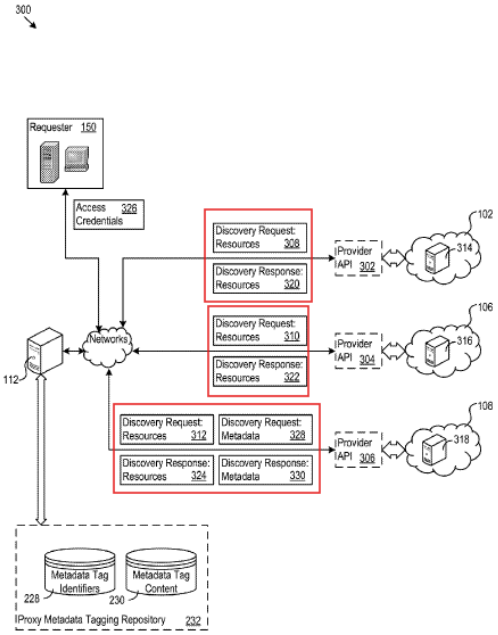


Figure 3

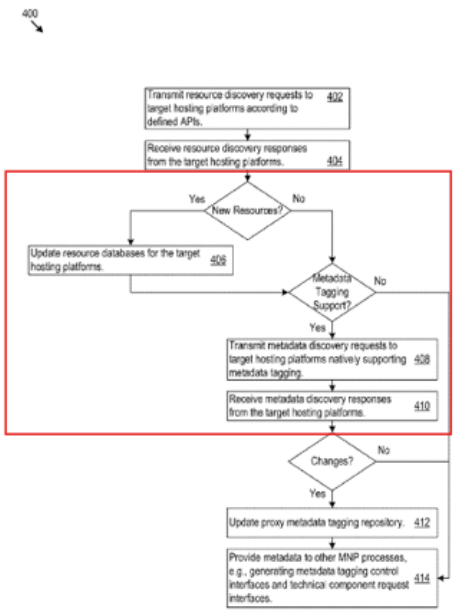


Figure 4

93. Accordingly, Foskett disclosed or rendered obvious all elements of limitation 1[b][ii].

Element 1[c]: "identifying, by the cloud-tagging orchestrator service, a unified tag of a plurality of unified tags for the cloud resource based on a pre-defined global tagging policy"

94. Foskett disclosed or rendered obvious this element because the MNP's tagging software (CTOS) funneled every provider's resources into one shared, provider-neutral repository and identified (identifying) the tag for a given resource (a unified tag of a plurality of unified tags) using definitions and value

constraints an administrator set up in advance (a pre-defined global tagging policy), applied the same way regardless which provider hosted the resource.

95. Foskett's stated purpose was to solve the problem of inconsistent tagging across different cloud providers so that cross-provider policies can be applied, the same problem identified in the '527 Patent. Foskett's Abstract explained that "[a] multi-cloud network proxy controls and normalizes the tagging data applicable across the multiple cloud service providers," and Foskett's Background confirmed that its architecture "provides a technical solution to establishing a normalized data model for cloud resources, given the existing wide disparity and lack of consistency in how metadata tagging has been approached in the past." (*Id.*, Abstract; ¶ 0015.)

96. Foskett's solution was architectural. Instead of letting each provider's own tagging rules control, the MNP funnels every provider's resources through a single, shared repository that the MNP itself defines and maintains. Foskett stated this directly: "[t]he MNP 112 controls and facilitates normalization of the metadata tag identifie[r]s and metadata tag content attached to the technical components hosted in the various target hosting platforms by the service providers," and it does so "regardless of whether any particular target hosting platform natively supports metadata tagging." (*Id.*, ¶ 0031.) That "proxy metadata tagging repository 232" is described as "a centralized source of metadata tagging information." (*Id.*)

97. Figure 2 in Foskett (below) showed this concretely: one provider's data center (i.e., Red Data Center) keeps its own native tag repository (238), another provider's data center (i.e., Blue Data Center) has no native tagging capability at all (244), and both nonetheless connect into the same single proxy repository (232) housed in the MNP. (*Id.*, Fig. 2; ¶¶ 0027, 0029.) Therefore, Foskett teaches that every provider's resources are tagged using tags within that shared repository, regardless of any separate, provider-specific tags. As a result, the same tag has a consistent meaning across cloud service providers. This corresponds to the claimed "unified tag" in the current claim element.

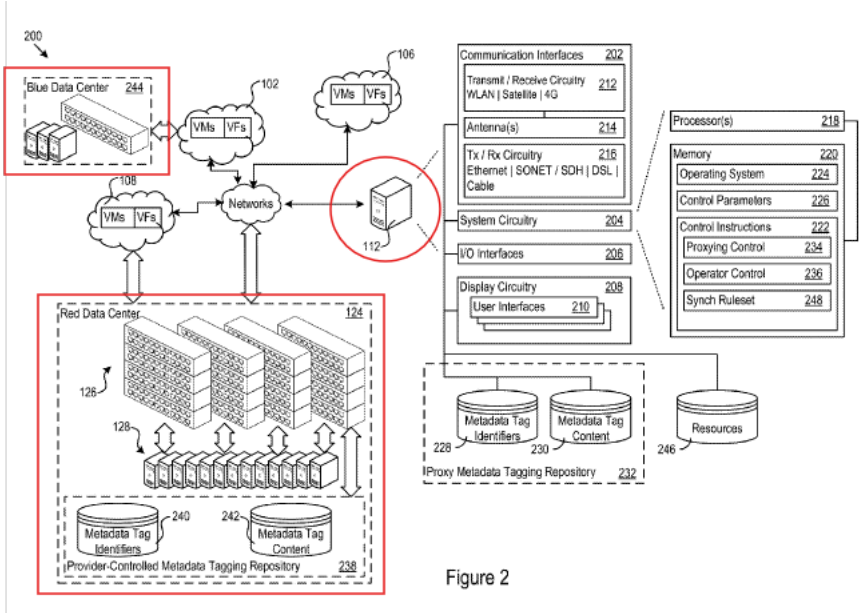


Figure 2

98. These tags are set according to a pre-defined global tagging policy. The specific tag assigned to a resource comes from a policy an administrator configures ahead of time, through control interfaces. (*Id.*, Fig. 5, 6, and 12.) Using

those interfaces, an administrator defines the available metadata tag identifiers, marks certain tags as "prioritized" for particular resource types and providers, and attaches "metadata tag type definitions" that constrain what content each tag can hold, all before any particular resource is ever tagged. (*Id.*, ¶¶ 0042–0043, 0046–0048.) For instance, an "Environment" tag can be limited to a pre-set list such as "Development," "Test," and "Production". (*Id.*)

99. Foskett confirmed that this configuration precedes any specific tagging decision: "the MNP 112 obtains designations of prioritized metadata tags" as a standing configuration, and only afterward, "[i]n response to a resource requester 150 requesting provisioning of a new resource," does the MNP consult those pre-existing designations to generate the request interface. (*Id.*, ¶ 0059.) The administrator's tag definitions and constraints are therefore already in place before any resource-specific request ever arrives. It is this advance configuration, fixed before the MNP acts on any given resource, that makes the resulting policy "pre-defined" within the meaning of the claim.

100. As for "global," the tags in Foskett are applied uniformly across service providers. As I discussed above, policies are configured by an administrator then applied to every applicable resource the MNP manages, irrespective of which provider hosts it. Additionally, because the MNP maintains one proxy metadata tagging repository (232) serving every connected provider, a

tag identifier and its associated content rules obviously carry the same meaning no matter which provider hosts the resource being tagged. (*Id.*, ¶¶ 0027, 0031, 0046-0048.) This is confirmed by Foskett's synchronization ruleset, which reconciles each provider's native tag data against "the metadata currently in the proxy metadata tagging repository 232" which is only possible *if* that repository is the one source of truth every provider is synchronized against. (*Id.,* ¶¶ 0015, 0037, 0040.) At minimum, it would have been an obvious application of Foskett's own disclosed design for a POSITA to configure that shared repository so a given tag carries one consistent meaning across every provider synchronized against it.

101. Foskett also disclosed a plurality of unified tags because the MNP applies its pre-defined tagging policy across the full population of resources it manages for a given requester, not to a single resource in isolation. Foskett's operator control circuitry defines multiple distinct metadata tag identifiers. For example, the "Environment" tag discussed above is one of several other prioritized tags an administrator designates for particular resource types and providers. (*Id.*, ¶¶ 0042-0043, 0046-0048.) Because the MNP's proxy metadata tagging repository stores and normalizes tag identifiers and content across this entire set of resources and providers, and because a POSITA would have understood that resources of different types and functions are assigned different unified tags according to the

same pre-defined policy, Foskett disclosed the claimed "plurality of unified tags." (*Id.*, ¶¶ 0027, 0031, 0040.)

102. Accordingly, Foskett disclosed or rendered obvious all elements of limitation 1[c].

Element 1[d]: "assigning, by the cloud-tagging orchestrator service, the unified tag to the cloud resource via an application programming interface (API) of a cloud provider of the plurality of cloud providers hosting the resource"

103. Foskett disclosed or rendered obvious this element because the MNP's tagging software (CTOS) pushed (assigning) that same tag (the unified tag) out to the resource's hosting provider by calling that provider's API (via an API of a cloud provider hosting the resource).

104. Having identified the unified tag for a resource under a pre-defined global tagging policy, as discussed in Element 1[c] above, the MNP assigned the tag to the appropriate resource via the API of the specific cloud provider that hosts that resource.

105. When an administrator updates an existing resource's tags and chooses to synchronize that change back to the provider. Figure 9 in Foskett (below) depicts this synchronization logic: "[i]n response to the Update option, the MNP 112 establishes a data connection to the target hosting platform (916) and transmits the metadata tags and metadata content currently in the proxy metadata

tagging repository 232 to the specified target hosting platform to perform synchronization (918)," and "[t]o push the data, the MNP 112 may call, e.g., APIs provided by the target hosting platform." (*Id.*, ¶ 0055.)

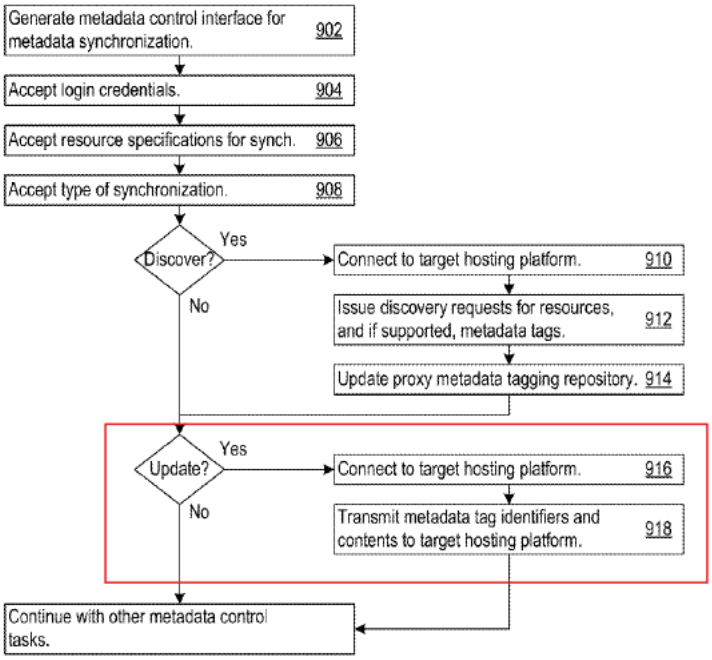


Figure 9

106. Additionally, when a resource requester provisions a brand-new resource. Foskett's Figure 11 process disclosed that: when the requester selects the synchronization option for a new resource, "the MNP 112 initiates the

provisioning, also including the dynamic request content," and "provides the dynamic request content with provisioning by transmitting (e.g., via API calls), to target hosting platform, the metadata tag identifiers and metadata tag contents entered into the request interface." (*Id.*, ¶ 0065.)

107. In both scenarios, the MNP calls the provider's API to push the tag to the resource. And in both scenarios, the tag content the MNP pushes out is the *same* unified tag content that the MNP's proxy metadata tagging repository holds as a result of the pre-defined global tagging policy discussed in Element 1[c].

108. Accordingly, Foskett disclosed or rendered obvious all elements of limitation 1[d].

109. Thus, Foskett disclosed or rendered obvious this claim.

2. CLAIM 3:

"The method of claim 1, wherein the cloud environment comprises a hybrid multi-cloud environment including a private cloud and a plurality of public clouds."

110. Foskett disclosed or rendered obvious this claim because the MNP concurrently managed an on-premise platform (a private cloud) alongside multiple named public cloud platforms (a plurality of public clouds) together, forming the kind of mixed deployment the claim covers (a hybrid multi-cloud environment).

111. Figure 1 (below) depicted various cloud service providers (*Id.*, Fig. 1.)

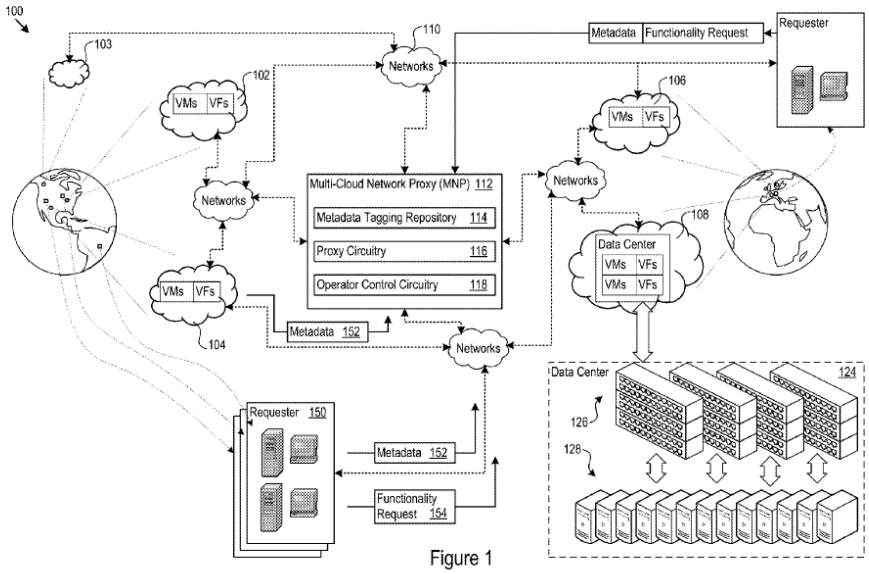


Figure 1

112. Foskett disclosed that these service providers can be public or private.

113. On the public side, Foskett disclosed that service providers can be public. Foskett named multiple public cloud platforms that the MNP manages: "Amazon Web Services (AWS), Google Compute Engine (GCE), Microsoft Azure (Azure)." (*Id.*, ¶ 0018.)

114. On the private side, Foskett explained that "[s]ervice providers may additionally or alternatively provide computing resources in 'on-premises' hosting platforms, which typically refers to a location with increased privacy and security compared to public cloud resources," and that "[a]n on-premise location may be within a secure facility owned by a resource requester which has moved computing functionality to a cloud based implementation." (*Id.*, ¶ 0018.) Foskett gives a

concrete example of such a platform: "Windows Azure Pack (WAP) for on-premise cloud implementations." (*Id.*)

115. In light of these disclosures, it would have been obvious to a POSITA that a cloud environment could be a hybrid multi-cloud environment including a private cloud and a plurality of public clouds. Indeed, before the '527 Patent, enterprises often use various cloud types for a cloud deployment. A private cloud deployment would often house the enterprise's most sensitive assets, over which it needed to exercise the most control and most reliability. But private clouds are often more expensive, less flexible, and more difficult to scale. That is why, before the '527 Patent, enterprises often paired a private cloud with public cloud services. Public cloud services are more scalable, cheaper, and flexible, so enterprises often used them for more routine operations. But to avoid dependence on a single public service provider (e.g., for redundancy) and to take advantage of the best-of-breed services and cost of each service provider, enterprises often used multiple public service providers at a time. E.g., https://www.cbtnuggets.com/blog/certifications/microsoft/aws-vs-azure-vs-google-cloud-wars-2019 (Ex. 1017).

116. Thus, Foskett disclosed or rendered obvious this claim.

3. CLAIM 4:

"The method of claim 1, wherein said assigning further enables visibility and control for the plurality of cloud resources by the security manager."

117. Foskett disclosed or rendered obvious this claim because the MNP's assigning (discussed above in element 1[d]) gives it an observable window into the current tag state of resources at every connected provider (visibility) and the means to determine and carry that tag state into effect at each provider (control) over the plurality of cloud resources hosted across the different connected providers.

118. Foskett disclosed "display circuitry 208 that generates user interfaces 210 locally or for remote display, e.g., in a web browser running at the resource requester 150," through "graphical user interfaces (GUIs)" and other interface elements. (*Id.*, ¶ 0024.) Foskett further disclosed that the metadata tag identifiers and content flowing into the proxy metadata tagging repository as a result of assigning "drive other processes within the MNP," including "the generation of interactive metadata tagging control interfaces." (*Id.*, ¶ 0036.) Because those interfaces are generated from, and reflect, the same tag data the MNP assigned, a POSITA would have understood that an operator using the MNP's interface is observing the actual, current tag state of resources at every connected provider. This observable presentation equates to enabling visibility.

119. Foskett disclosed that the MNP's assigning is also what gives it control over those resources. Foskett explained that the MNP "controls and facilitates normalization of the metadata tag identifie[r]s and metadata tag content attached to the technical components hosted in the various target hosting platforms by the service providers ... regardless of whether any particular target hosting platform natively supports metadata tagging." (*Id.*, ¶ 0031.) A POSITA would have understood that it is the MNP's assignment, which carries its tagging decisions into effect at the resource and constitutes control over the resource's tag state.

120. Without the assigning step, neither capability would exist. If the MNP never assigned tag content to a resource's hosting provider, there would be no current tag state to observe through the MNP's interfaces. This would defeat visibility. Likewise, there would be no mechanism by which the MNP's tagging decisions would be carried into effect at the provider. This would defeat control. It is the MNP's assigning step that gives it both an observable window into, and an operative hand on, the tag state of the plurality of cloud resources across every connected provider.

121. Thus, Foskett disclosed or rendered obvious this claim.

4. CLAIM 6:

"The method of claim 1, wherein the plurality of cloud resources comprise virtual machine instances."

122. Foskett disclosed or rendered obvious this claim because the resources the MNP tagged and managed (the plurality of cloud resources) included virtual machines (virtual machine instances) running at a provider's data center.

123. Foskett disclosed the cloud resources can include virtual machines. Foskett explained that "FIG. 1 shows a target hosting platform implemented by a data center 124 for the service provider 108," and that "[t]he data center 124 supports running many different virtual machines (VMs), each potentially running many different virtual functions (VFs)." (*Id.*, ¶ 0022.) Foskett's Figure 10 process further confirmed that the MNP's resource-request interface is used, for example, "when the resource requester 150 is requesting a new VM hosted by the ... service provider," at which point "the MNP 112 may add to the request interface the metadata tags previously designated as prioritized for VMs for the ... service provider." (*Id.*, ¶ 0061.)

124. Thus, Foskett disclosed or rendered obvious this claim.

5. CLAIM 7:

"The method of claim 1, wherein the security manager comprises a cloud based security management appliance."

125. Foskett disclosed or rendered obvious this claim because the MNP's infrastructure (security manager) was described as capable of distributed implementation across multiple processors, memories, and systems (comprises), communicating with cloud providers over both private and public networks, consistent with a cloud-based security management appliance.

126. Consistent with my mapping in Element 1[a] above, Foskett disclosed that the MNP's infrastructure is not tied to any single physical box: "system circuitry 204 may include any combination of hardware, software, firmware, or other logic," and its implementation "may be distributed," spanning "multiple distinct system components, such as multiple processors and memories," and "multiple distributed processing systems." (*Id.*, ¶¶ 0026, 0068.)

127. Foskett further disclosed that the MNP's infrastructure communicates with the service providers' hosting platforms over networks that "may include private and public networks," consistent with a cloud-based, network-accessible appliance rather than a fixed, locally-tethered box. (*Id.*, ¶¶ 0018–0019.)

128. Thus, Foskett disclosed or rendered obvious this claim.

6. CLAIM 8:

Element 8[Preamble]: "A non-transitory computer-readable storage medium embodying a set of instructions, which when executed by a processing resource of a security manager of a private network, causes the processing resource to perform a method comprising:"

129. Foskett could be implemented on a computer. Specifically, Foskett disclosed that its methods and processing "may store or access instructions for execution" where the "instructions may be stored in a tangible storage medium that is other than a transitory signal, such as a flash memory, a Random Access Memory (RAM), a Read Only Memory (ROM), an Erasable Programmable Read Only Memory (EPROM)," or similar non-transitory media, executed by "an instruction processor." (*Id.*, ¶¶ 0066–0067.) It would have been obvious to a POSITA that the MNP has CRSM, instructions, and other typical computing components because it performs various steps in a computer, as shown in claim 1.

130. Foskett disclosed the "processing resource" for the reasons discussed in Element 1[a] above, because the processing resource, just like the MNP, is what performs the actual method steps discussed below.

131. Foskett disclosed "the security manager of a private network" for the same reasons provided in Element 1[a] above.

132. Foskett disclosed that the MNP's infrastructure "may include any combination of hardware, software, firmware, or other logic," and its

implementation "may be distributed," spanning "multiple distinct system components, such as multiple processors and memories," and "multiple distributed processing systems." (*Id.*, ¶¶ 0026, 0068.) Foskett further disclosed that its "system circuitry 204 may include one or more instruction processors 218 and memories 220" where the "memory 220 stores, for example, control instructions 222 and an operating system." (*Id.*, ¶ 0026; Fig. 2 (below).)

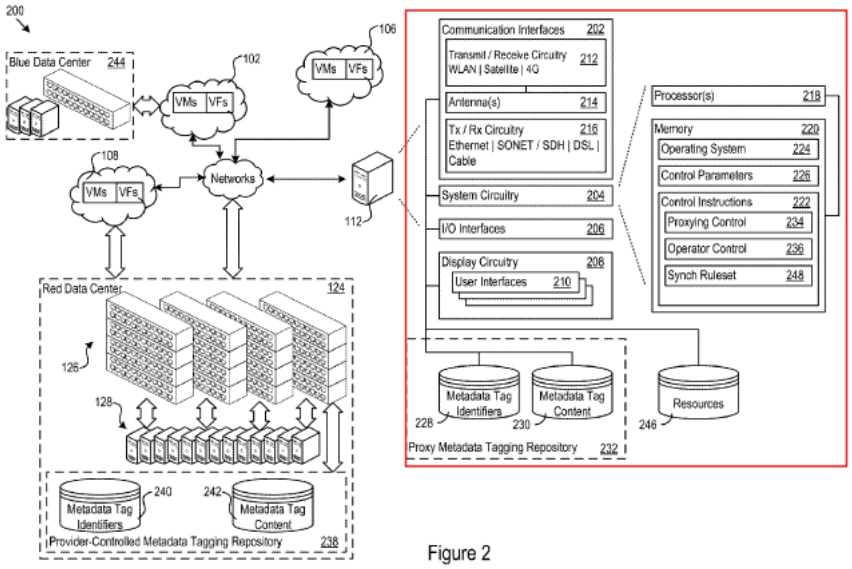


Figure 2

133. Thus, Foskett disclosed or rendered obvious this element.

Element 8[a]: "receiving information regarding each cloud provider of a plurality of cloud providers associated with a cloud environment used by the private network;"

134. Since this element recites the same requirements as element 1[a], Foskett disclosed this element for the same reasons discussed in element 1[a].

Element 8[b][i]: "based on the received information, for each cloud resource of a plurality of cloud resources hosted by the plurality of cloud providers on behalf of the private network:"

135. Since this element recites the same requirements as element 1[b][i],

Foskett disclosed this element for the same reasons discussed in element 1[b][i].

Element 8[b][ii]: "retrieving information associated with the cloud resource"

136. Since this element recites the same requirements as element 1[b][ii],

Foskett disclosed this element for the same reasons discussed in element 1[b][ii].

Element 8[c]: "identifying a unified tag of a plurality of unified tags for the cloud resource based on a pre-defined global tagging policy"

137. Since this element recites the same requirements as element 1[c],

Foskett disclosed this element for the same reasons discussed in element 1[c].

Element 8[d]: "assigning the unified tag to the cloud resource via an application programming interface (API) of a cloud provider of the plurality of cloud providers hosting the resource."

138. Since this element recites the same requirements as element 1[d],

Foskett disclosed this element for the same reasons discussed in element 1[d].

7. CLAIM 10:

"The non-transitory computer-readable storage medium of claim 8, wherein the cloud environment comprises a hybrid

multi-cloud environment including a private cloud and a plurality of public clouds."

139. For the reasons discussed in element 8[Preamble], Foskett disclosed a non-transitory computer-readable storage medium. The remainder of claim 10 recites the same requirements as claim 3, so Foskett disclosed this element for the same reasons discussed in claim 3.

8. CLAIM 11:

"The non-transitory computer-readable storage medium of claim 8, wherein said assigning further enables visibility and control for the plurality of cloud resources by the security manager."

140. For the reasons discussed in element 8[Preamble], Foskett disclosed a non-transitory computer-readable storage medium. The remainder of claim 11 recites the same requirements as claim 4, so Foskett disclosed this element for the same reasons discussed in claim 4.

9. CLAIM 13:

"The non-transitory computer-readable storage medium of claim 8, wherein the plurality of cloud resources comprise virtual machine instances."

141. For the reasons discussed in element 8[Preamble], Foskett disclosed a non-transitory computer-readable storage medium. The remainder of claim 13

recites the same requirements as claim 6, so Foskett disclosed this element for the same reasons discussed in claim 6.

10. CLAIM 14:

"The non-transitory computer-readable storage medium of claim 8, wherein the security manager comprises a cloud based security management appliance."

142. For the reasons discussed in element 8[Preamble], Foskett disclosed a non-transitory computer-readable storage medium. The remainder of claim 14 recites the same requirements as claim 7, so Foskett disclosed this element for the same reasons discussed in claim 7.

11. CLAIM 15:

Element 15[Preamble]: "A multi-tag orchestration system comprising: a processing resource; and a non-transitory computer-readable medium, coupled to the processing resource, having stored therein instructions that when executed by the processing resource cause the processing resource to perform a method comprising:"

143. Foskett disclosed or rendered obvious a multi-tag orchestration system: "[a] multi-cloud network proxy [that] controls and normalizes the tagging data applicable across the multiple cloud service providers," implemented using the MNP's processor(s) 218 (a processing resource) coupled to memory 220 (a non-transitory computer-readable medium) storing control instructions 222 that,

when executed, cause the processing resource to perform the claimed method. (*Id.*, Abstract, ¶ 0026.)

144. For the remaining elements, Foskett disclosed those elements for the same reasons discussed in element 1[Preamble] and element 8[Preamble] above.

Element 15[a]: "receiving information regarding each cloud provider of a plurality of cloud providers associated with a cloud environment used by a private network;"

145. Since this element recites the same requirements as element 1[a], Foskett disclosed this element for the same reasons discussed in element 1[a].

Element 15[b][i]: "based on the received information, for each cloud resource of a plurality of cloud resources hosted by the plurality of cloud providers on behalf of the private network:"

146. Since this element recites the same requirements as element 1[b][i], Foskett disclosed this element for the same reasons discussed in element 1[b][i].

Element 15[b][ii]: "retrieving information associated with the cloud resource"

147. Since this element recites the same requirements as element 1[b][ii], Foskett disclosed this element for the same reasons discussed in element 1[b][ii].

Element 15[c]: "identifying a unified tag of a plurality of unified tags for the cloud resource based on a pre-defined global tagging policy"

148. Since this element recites the same requirements as element 1[c], Foskett disclosed this element for the same reasons discussed in element 1[c].

Element 15[d]: "assigning the unified tag to the cloud resource via an application programming interface (API) of a cloud provider of the plurality of cloud providers hosting the resource."

149. Since this element recites the same requirements as element 1[d], Foskett disclosed this element for the same reasons discussed in element 1[d].

12. CLAIM 17:

"The system of claim 15, wherein the cloud environment comprises a hybrid multi-cloud environment including a private cloud and a plurality of public clouds."

150. For the reasons discussed in element 15[Preamble], Foskett disclosed a system. The remainder of claim 17 recites the same requirements as claim 3, so Foskett disclosed this element for the same reasons discussed in claim 3.

13. CLAIM 20:

"The system of claim 15, wherein the plurality of cloud resources comprise virtual machine instances."

151. For the reasons discussed in element 15[Preamble], Foskett disclosed a system. The remainder of claim 20 recites the same requirements as claim 6, so Foskett disclosed this element for the same reasons discussed in claim 6.

C. Ground 2: Combination of Foskett and Whitlock renders claims 2, 5, 9, 12, 16, and 19 obvious.

152. ***Whitlock***: Whitlock disclosed systems and methods for classifying and organizing IT assets using tags, including automatically generated tags based on asset attributes. (Ex. 1005, Abstract, ¶ 0002.) Whitlock further disclosed that

"certain tag sets are automatically generated" through a "harvesting" process in which "one or more items of data concerning the configuration of the node are automatically stored." (Ex. 1005, ¶ 0059.) This harvested "data can then be used by the tag-based monitoring and management tool ... to automatically assemble system tag sets without user intervention," using "information about the assets that is available from (or harvested from) the assets," including "information about the asset's operating system, database version, IP address, and/or other such[] configuration information." (*Id.*, ¶ 0041.)

153. Whitlock's tagging profile interface further allowed an administrator to define automatically-applied "conditions" such as host name, IP address, and system tag set matching criteria that determine which tags are assigned to a node (i.e., an IT asset that can comprise, for example, "servers, desktops, directory servers, hypervisors, databases, middleware applications, network devices, virtual infrastructures," etc.). (*Id.*, ¶¶ 0037, 0070, 0080.)

154. Whitlock depicted a tag set management interface in which "tag set display 810 is named 'location' and refers to tags that are interrelated to each other because the tags each indicate an asset's location," with example tags for New York, Portland, and Tokyo. (*Id.*, Fig. 8 (below); ¶¶ 0072–0073.) Whitlock also listed "Primary Function" as a tag set category. (*Id.*, Fig. 8 (below).)

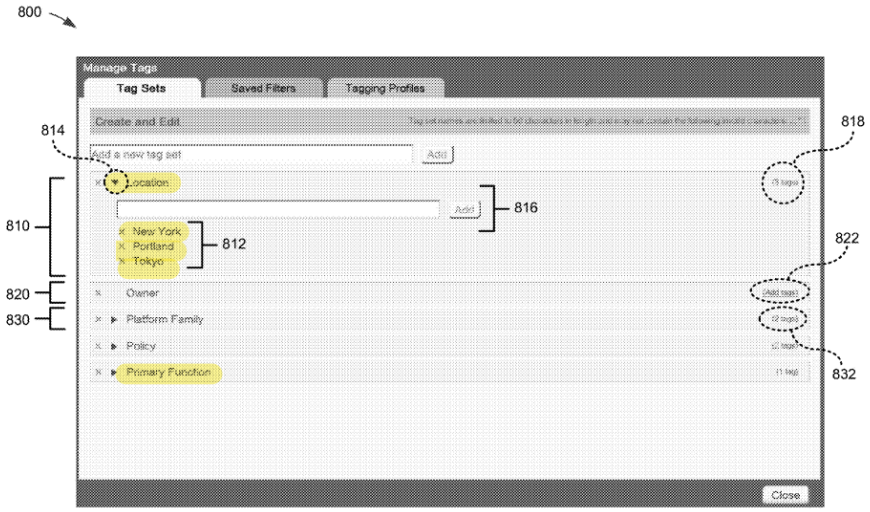


FIG. 8

155. Whitlock additionally disclosed that, once a tagging profile is applied to nodes satisfying its conditions, that step generates "a tagged set of nodes" as a discrete artifact (*id.*, ¶ 0069), and only afterward is that already-tagged set of nodes made available, as a separate downstream step, for use by "a compliance and configuration control tool" to apply a rule or other policy to the resulting tagged set. (*Id.*, ¶ 0070.)

1. Motivations to combine

156. A POSITA would have been motivated to look to Whitlock, and to apply its information tag-set teachings such as location and function within Foskett's MNP for several reasons.

157. First, Foskett and Whitlock addressed the same underlying problem from complementary angles. Foskett's own stated goal was normalizing tag

identifiers and content across disparate cloud providers (Ex. 1004, ¶ 0015), but Foskett's global tagging policy examples focus principally on administrator-defined identifiers, prioritization, and content-format constraints. Foskett itself does not catalog location or function as specific bases for identifying a tag; nor does it expressly separate a resource's tagged state from the subsequent application of policy to that tagged state. Whitlock fills both information gaps by disclosing (1) that IT asset tags are commonly organized around exactly these two bases: a "Location" tag set with example values for specific cities (e.g., New York, Portland, and Tokyo), and a "Primary Function" tag set category (Ex. 1005, Fig. 8; ¶¶ 0072–0073.) and (2) tag assignment and policy application are two distinct, sequential steps: tagging first (*id.*, ¶ 0069), followed by separate use of the resulting tagged set by a compliance and configuration control tool. (*Id*., ¶ 0070) A POSITA implementing Foskett's global tagging policy would naturally have looked to these well-known IT asset tagging categories and conventions, like those Whitlock catalogs, both to decide what the policy's tags should be based on and to understand how tag assignment and downstream policy application relate to one another. This combination would improve and complement the system in Foskett without taking anything away from it.

158. Second, incorporating location and function as tagging bases would have furthered Foskett's own stated objective of avoiding "inconsistent and wildly

variable metadata tagging" (Ex. 1004, ¶ 0063), since location and function are among the most basic, commonly-tracked attributes of any cloud resource, regardless of which provider hosts it. A POSITA would have recognized that a tagging policy omitting these two attributes would be markedly less useful for resource management and security-policy application than one that included them.

159. Third, combining Whitlock's teachings with Foskett would have amounted to nothing more than applying known techniques (organizing tags by location and function and separating tag assignment from policy application in Whitlock) to a known device (the MNP in Foskett) to yield the predictable result of a more complete and useful global tagging policy.

160. At least because Whitlock disclosed location and function tag sets, and the separation between tag assignment and policy application, as well-known, commonly-implemented categories for organizing IT asset tags—the same basic kind of tag identifier and content management that Foskett's MNP already performs—a POSITA would have had a reasonable expectation of success in incorporating location and function as bases for Foskett's pre-defined global tagging policy and security-policy application process.

2. Claim 2:

"The method of claim 1, further comprising facilitating, by the security manager, application of a security policy expressed with reference to one or more unified tags of the

plurality of unified tags by retrieving information associated with a particular cloud resource of the plurality of cloud resources from a storage database."

161. Foskett in view of Whitlock disclosed or rendered obvious this claim because Foskett's proxy metadata tagging repository (storage database) from which the MNP retrieves an already-assigned unified tag, and Whitlock confirmed that a security policy is applied by reference to that tag only after the tag itself has already been assigned.

162. Foskett disclosed that the MNP maintains a storage database in the form of its "proxy metadata tagging repository 232" that stores the unified tag identifiers and tag content applicable to each cloud resource. (*Id.*, ¶¶ 0027, 0031.) Foskett further disclosed that this repository is used to drive further MNP processes, including "the generation of interactive metadata tagging control interfaces and driving dynamic adaptation to technical component request interfaces," under which the MNP "add[s] specifically identified metadata tag identifiers and tag content fields to technical component request interfaces" based on tags retrieved from the repository. (*Id.*, ¶ 0036.) Foskett additionally stated that the MNP "retrieves the matching metadata tags and their metadata tag content, if any is currently set, from the proxy metadata tagging repository 232." (*Id.*, ¶ 0045.)

163. Foskett further disclosed that certain tags carry mandatory-value and content-format requirements that the MNP enforces before accepting tag content: a metadata tag type definition may specify that "a value must be provided for the metadata tag," and the MNP "constrains the input for the metadata tag content according to the type definition," rejecting content that does not conform. (*Id.*, ¶¶ 0049–0051.)

164. A POSITA would have understood that these mandatory-value and content-format requirements are themselves part of what it means to identify and assign a unified tag. They define what a valid tag looks like, not what happens to a resource once it already has been tagged.

165. Whitlock confirmed that a POSITA would have understood tag assignment and security policy application to be two distinct steps, performed in that order. Whitlock described applying system tags to a node, assigning nodes to groups, and then automatically applying tasks, rules and/or policies to these groups (Ex. 1005, ¶ 0060). Whitlock further disclosed generating "a tagged set of nodes" by assigning one or more tags to nodes satisfying a tagging profile's conditions (*Id.*, ¶ 0069), and structuring that tagged set into a data set usable by a downstream tool (*Id.*, ¶ 0070). Only afterward, as a separate step, is a "compliance and configuration control tool ... used to apply a rule (or other policy) to the tagged set of nodes in the data set." (*Id.*, ¶ 0071).

166. Whitlock's compliance and configuration control tool operates by "capturing a baseline of server file systems, desktop file system, directory servers, databases, virtual systems middleware applications and/or network device configurations in a known good state," with "[o]ngoing integrity checks then compar[ing] the current states against these baselines to detect changes." (*Id.*, ¶¶ 0037-0038.) A POSITA would have understood this baseline-and-integrity-check function to be a security policy: detecting unauthorized or unexpected drift from a resource's known-good configuration state is a well-recognized security control, because such drift is a standard indicator of compromise or misconfiguration risk. Whitlock confirmed this rule is applied at the level of a tagged group of resources, not to the tagging step itself: a "rule group" (e.g., "Critical Change Audit Rules") can be selected "for application to a node group" created from tags, so that "tasks and policies can be scoped to the created node groups, thereby ... increasing monitoring and policy compliance capabilities." (*Id.*, ¶ 0094.) This confirms that the rule governs what happens to an already-tagged group of nodes going forward.

167. Consistent with my mapping in Element 1[a] above, Foskett's MNP retrieves the unified tag already assigned to a resource from the proxy metadata tagging repository (storage database) and, consistent with the tag-then-policy sequence Whitlock disclosed, applies a security policy to that already-tagged resource. (Ex. 1004, ¶¶ 0027, 0031; Whitlock, ¶¶ 0037, 0069–0071, 0094.)

168. Accordingly, Foskett in view of Whitlock disclosed or rendered obvious all elements of claim 2.

3. Claim 5:

"The method of claim 1, wherein the pre-defined global tagging policy identifies the unified tag based on one or more of the information regarding the cloud resource, a location of the resource, and a function of the cloud resource."

169. Foskett disclosed or rendered obvious this claim because Foskett's own tag type definitions already key on information about the resource, its location, and its function, and Whitlock independently confirmed location and function as established, named bases for identifying a tag to the extent Foskett's disclosure of those two bases is found insufficient on its own.

170. Foskett alone disclosed that the pre-defined global tagging policy identifies the unified tag based "on one or more of the information regarding the cloud resource." As discussed in Element 1[c] above, Foskett's operator control circuitry defines metadata tag type definitions (e.g., datatype, mandatory-value, and valid-value-list constraints) based on information regarding the resource being tagged, such as its resource type and provider. (Ex. 1004, ¶¶ 0042–0043, 0046–0048.)

171. A POSITA would also have found Foskett alone to disclose that the pre-defined global tagging policy identifies the unified tag based "on the location

of the resource." Foskett disclosed that resources are "provisioned into a region hosted by a resource provider," where a "region" refers to "a unit of hosting capacity in a particular geographic region." (*Id.*, ¶ 0020.) Foskett's synchronization control interface includes "input fields 706 for provider, location, and resource type," which "allow the operator to specify which types of resources to synchronize at which provider and which provider location." (*Id.*, ¶ 0053.) A POSITA reviewing this disclosure would have understood that Foskett's own metadata tag type definitions—which Foskett disclosed can be configured to key on any field an administrator selects, including "resource type" and "provider" (*id.*, ¶¶ 0042–0043, 0045–0048)—would just as readily key on the location field Foskett already collects and displays for each resource, since Foskett provides no basis to treat location differently from the other resource attributes its own control interfaces already expose and constrain.

172. A POSITA would also have found Foskett alone to disclose that the pre-defined global tagging policy identifies the unified tag based on "a function of the cloud resource." Foskett's own example of a prioritized, constrained metadata tag is the "Environment" tag, which the MNP restricts to a pre-defined list of values such as "Development," "Test," and "Production." (*Id.*, ¶¶ 0047–0048.) A POSITA would have understood that these values identify a resource's functional role within the requester's deployment (i.e., whether the resource is functioning as

a development instance, a test instance, or a production instance) and that the same administrator-defined, constraint-based mechanism Foskett already applies to this functional classification would apply equally to other functional roles (e.g., web server, database, application server) an administrator chooses to track.

173. To the extent that Foskett alone does not expressly identify location or function as separate bases, the combination of Foskett and Whitlock disclosed both. Whitlock disclosed a "Location" tag set used to identify assets by physical location, with example tags for New York, Portland, and Tokyo (Ex. 1005, Fig. 8; ¶¶ 0072–0073), and a "Primary Function" tag set used to identify assets by their functional role (*id.*, Fig. 8). A POSITA implementing Foskett's pre-defined global tagging policy, in view of Whitlock's teachings, would have configured that policy to identify the unified tag based on the resource's location or function using the same administrator-defined, constraint-based mechanism Foskett already disclosed for other tag types. (Ex. 1004, ¶¶ 0042–0043, 0046–0048.)

174. Accordingly, Foskett in view of Whitlock disclosed or rendered obvious all elements of claim 5.

4. Claim 9:

"The non-transitory computer-readable storage medium of claim 8, wherein the method further comprises facilitating application of a security policy expressed with reference to one or more unified tags of the plurality of unified tags by retrieving information associated with a particular cloud resource of the plurality of cloud resources from a storage database."

175. For the reasons discussed in element 8[Preamble], Foskett disclosed a non-transitory computer-readable storage medium. The remainder of claim 9 recites the same requirements as claim 2, so Foskett in view of Whitlock disclosed this claim for the same reasons discussed in claim 2.

5. Claim 12:

"The non-transitory computer-readable storage medium of claim 8, wherein the pre-defined global tagging policy identifies the unified tag based on one or more of the information regarding the cloud resource, a location of the resource, and a function of the cloud resource."

176. For the reasons discussed in element 8[Preamble], Foskett disclosed a non-transitory computer-readable storage medium. The remainder of claim 12 recites the same requirements as claim 5, so Foskett in view of Whitlock disclosed this claim for the same reasons discussed in claim 5.

6. Claim 16:

"The system of claim 15, wherein the method further comprises facilitating application of a security policy expressed with reference to one or more unified tags of the plurality of unified tags by retrieving information associated with a particular cloud resource of the plurality of cloud resources from a storage database."

177. For the reasons discussed in element 15[Preamble], Foskett disclosed a system. The remainder of claim 16 recites the same requirements as claim 2, so Foskett in view of Whitlock disclosed this element for the same reasons discussed in claim 2.

7. Claim 19:

"The system of claim 15, wherein the pre-defined global tagging policy identifies the unified tag based on one or more of the information regarding the cloud resource, a location of the resource, and a function of the cloud resource."

178. For the reasons discussed in element 15[Preamble], Foskett disclosed a system. The remainder of claim 19 recites the same requirements as claim 5, so Foskett in view of Whitlock disclosed this claim for the same reasons discussed in claim 5.

D. Ground 3: Combination of Foskett and Juniper renders claim 18 obvious.

179. ***Juniper***: Juniper disclosed a single point of management for a multi-cloud environment, including route propagation, security, and application

deployment. (Ex. 1006, Title.) Juniper explained that "[i]n some existing techniques, each VCE administrator would need to setup policies to establish communication tunnels between the different VCEs," resulting in "delays or errors due to the complications associated with ensuring proper route propagation, security, and application deployment." (*Id.*, ¶ 0010.) Juniper's solution is a "centralized controller... that allows one administrator, with a single pane of glass, to control the interconnections and security within" a multi-cloud network. (*Id.*, ¶ 0032.) Juniper further disclosed that, whereas other techniques would require a network administrator to interface separately with a different controller for each cloud, "[t]he network administrator may interface with only [the] controller... via a single pane of glass to establish the communication links" across every managed cloud. (*Id.*, ¶ 0035.)

1. Motivations to combine

180. A POSITA would have been motivated to look to Juniper, and to present Foskett's centralized visibility and control through a single pane of glass user interface as Juniper disclosed, for several reasons.

181. First, Foskett already disclosed that its MNP has centralized visibility and control over tagged resources across multiple connected providers (as discussed above in claim 4). But Foskett's control interfaces (Ex. 1004, Figs. 5, 6, and 12) are described principally in terms of tag management for individual

resource types and providers, without expressly describing a single consolidated view spanning all providers at once. Juniper fills this gap by disclosing that a multi-provider network management platform can and should present its centralized, cross-environment control through exactly such a consolidated, single pane of glass interface, so that "an administrator may be able to configure communication between different VCEs [virtual computing environments]... rather than establish communication links on a VCE-by-VCE or data center-by-data center basis." (Ex. 1006, ¶ 0074.)

182. Second, both references address the same administrative burden created by managing resources across multiple, heterogeneous cloud environments. Juniper's own stated purpose was to spare an administrator from needing to interface separately with each cloud's own controller across every managed cloud (*id.*, ¶¶ 0010, 0035.), the same problem Foskett's MNP addresses through tag normalization. A POSITA would have recognized that pairing Foskett's centralized tagging repository with Juniper's single pane of glass presentation layer would directly further both references' shared goal of unified, cross-provider administration.

183. Third, combining Juniper's single pane of glass presentation with Foskett's centralized, cross-provider tagging repository would have amounted to applying a known technique (consolidating multi-cloud administration into a single

controller interface in Juniper) to a known device (MNP in Foskett, which already aggregates cross-provider information in its proxy metadata tagging repository) to achieve the predictable result of a unified administrative view.

184. At least because Juniper disclosed a well-known, working means of presenting multi-cloud resource information and control through a single centralized controller interface, a POSITA would have had a reasonable expectation of success in presenting Foskett's centralized, cross-provider tag data through such an interface.

2. Claim 18:

"The system of claim 15, wherein said assigning enables visibility and control of the plurality of cloud resources via a single pane of glass user interface."

185. Foskett and Juniper disclosed or rendered obvious this claim because Foskett's assigning step already gave its MNP centralized visibility and control over resources at every connected provider (as shown in claim 4 above), and Juniper disclosed presenting that same kind of centralized, cross-provider control to an administrator through a single, unified interface (a single pane of glass user interface).

186. Juniper disclosed presenting exactly this kind of centralized, cross-provider resource information through a single pane of glass: "[a] centralized controller (e.g., controller 14) that allows one administrator, with a single pane of

glass, to control the interconnections and security within" a multi-cloud network, such that "the network administrator may interface with only [the] controller... via a single pane of glass to establish the communication links" across every managed cloud. (Ex. 1006, ¶¶ 0032, 0035.)

187. A POSITA combining Foskett with Juniper would have presented Foskett's MNP's centralized, cross-provider visibility and control, through Juniper's single pane of glass interface.

188. Accordingly, Foskett in view of Juniper disclosed or rendered obvious this claim.

E. Ground 4: Combination of Kung and Serhiienko renders claims 1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 13, 14, 15, 16, 17, and 20 obvious.

189. ***Kung***: Kung disclosed a contextual security platform that automatically configures native, network-level security mechanisms across an organization's computer network using security policies specified at an application level. (Ex. 1007, Abstract, ¶ 0009.) Kung's platform runs as software on one or more computers in communication with a hybrid computer network infrastructure "having both traditional hardware resources and virtual resources provided by private and public cloud infrastructure services," including "Amazon AWS, Microsoft Azure and Google GCP" as example public cloud providers and

"OpenStack and VMware vCenter" as example private cloud environments. (*Id.*, ¶¶ 0009, 0046.)

190. Kung's contextual security platform functions as an engine that reads data from, and writes data to, a database storing application-level security policies, a listing of logical groups, a system model comprising a database of infrastructure resource information, and contextual network flow information. (*Id.*, ¶ 0050.) The contextual security platform communicates with the API of each cloud service provider, exchanging messages using the native APIs provided by those services, and can request and receive information on network resources and traffic flows and use the API to configure security mechanisms native to those cloud service providers. (*Id.*, ¶ 0051.)

191. For each infrastructure resource, Kung's platform associates attributes, in the form of key/value pairs, with infrastructure resources using several mechanisms, including reading "infrastructure tags" that cloud infrastructure service providers such as AWS, Azure, GCP, and OpenStack allow customers to associate with their resources, which "are read using the API of the service provider during the process of discovering infrastructure resources and used to automatically map discovered infrastructure resources to logical groups." (*Id.*, ¶ 0141.) Kung further disclosed that logical groups can be created automatically based on various types of properties assigned by cloud service

providers, such as Amazon AWS's own VPC, region, availability zone, account, network, and subnet properties, Microsoft Azure's own Resource Group, region, virtual network, and account properties, and OpenStack's own project, region, network, and subnet properties. (*Id.*, ¶¶ 0145–0148.) Kung's platform selects an infrastructure resource as a member of one or more logical groups using these attributes, and computes a configuration for network security mechanisms using application-level security policies together with the infrastructure resources that are members of the logical groups to which each policy applies. (*Id.*, ¶ 0085.)

192. Kung disclosed that its platform can be used to provide "effective global security" and to enforce policies for an enterprise's full set of resources by combining the different capabilities of native security mechanisms at multiple compute environments in a coordinated way. (*Id.*, ¶¶ 0073, 0077, 0132.)

193. Kung's attributes are populated by reading each provider's own native tags and resource properties such as AWS's, Azure's, and OpenStack's own respective tagging and resource-property conventions. Kung's own disclosure confirmed that its attribute assignment mechanism operates on "infrastructure tags" received from each individual infrastructure service provider "using the API of the service provider" (*id.*, ¶ 0141), reflecting whatever tagging or resource-property convention that particular provider happens to use.

194. ***Serhiienko***: Serhiienko disclosed an extensible tag management middleware based on cloud functions, called the Universal Connector ("UC"), that turns tags into a powerful declarative means of cloud management, demonstrated by a deployment administration scenario involving both AWS and Google Cloud Platform services. (Ex. 1008, Abstract.) Serhiienko explained that "[o]ne of the biggest challenges in multi-cloud tag-based management is that each provider implements tagging in a different way," and that a "universal tagging system can thus aid this process by implementing a ruleset for tags that meets the constraints of all providers simultaneously, and by providing a single interface through which to manage tagging across all providers," so that "an administrator can use the tagging features without studying the internal limitations and constraints of each individual provider." (*Id.*, § III.A.) Serhiienko's implementation achieves this "by providing a common API tagging method that abstracts the provider-specific details and provides a uniform interface, and by providing the ability to apply tags automatically." (*Id.*)

195. The UC's core functions include managing tags "in a secure, audible and cost-effective manner" and performing "best practices cloud management tasks driven by regulated tags," encompassing the management of tags themselves, the tag-driven allocation of resources, and the aggregation of information about the current multi-cloud system state. (*Id.*, § I; § III.) Serhiienko's ruleset defines, for

each resource type and provider filter, a rule composed of conditions plus a set of tag key-value pairs to apply when a resource's metadata satisfies those conditions, and this auto-generation method depends on retrieving the resource's full metadata from its provider so that "rules [can] be created for any field in the metadata." (*Id.*, § III.B.) When a resource satisfies a rule's conditions, "the method then calls the appropriate tagResource function and applies the tag to the resource." (*Id.*, § III.B.)

196. Serhiienko's tagging algorithm iterates over every provider, calling a function to get each resource's metadata and a function to get the list of resources, checking each resource's metadata against the rules, and then calling a function to tag the matching resources. (*Id.*, § III.C, Listing 1.) Serhiienko explained that "[i]n order to satisfy the need for consistent tagging we employ the auto-tagging functionality that allows us to define a set of tagging rules in a YAML file," and that "[t]he rules can apply to any metadata field of each individual resource, across all providers," implemented through an "autoGenerate API method" that "composes several cloud functions from all providers in order to retrieve all resources that satisfy the ruleset and to apply the necessary tags to them," such that "just one invocation of this function can tag every resource across all managed projects with the right tag." (*Id.*, § IV.A.)

197. Serhiienko was presented at an IEEE/ACM conference and published by IEEE in 2019. E.g., https://ieeexplore.ieee.org/document/8752784. (Ex. 1018.)

IEEE is one of the two main professional organizations in computer science—the ACM being the other. I am a member of both. Both the ACM and IEEE publish high-quality peer-reviewed academic research. The IEEE also publishes widely-applied industry standards. I have reviewed such research for IEEE conferences and am presently on an IEEE standards committee., I have consulted IEEE resources for at least twenty years, well before the ’527 Patent was filed, to learn about new technological developments. A POSITA interested in cloud infrastructure and resource management would have known to review IEEE publications.

1. Motivations to combine

198. A POSITA would have been motivated to combine Kung’s contextual security platform with Serhiienko’s UC with a reasonable expectation of success for several reasons.

199. First, the two references address complementary halves of the same problem. Kung disclosed a contextual security platform that computes and applies security policy based on attributes assigned to infrastructure resources, using information read from each provider’s own native tags and resource properties, such as VPCs, region, availability zone, account, network, and subnet properties. (Ex. 1007, ¶¶ 0145–0148.) However, Kung does not itself disclose normalizing those differently formatted, provider-native values into a single tag that carries the

same meaning across providers. (Ex. 1007, ¶¶ 0085, 0141.) Serhiienko disclosed exactly the multi-provider tag normalization that Kung lacks: a universal ruleset and connector that "meets the constraints of all providers simultaneously," applying "a single interface through which to manage tagging across all providers." (Ex. 1008, § III.A.) A POSITA seeking to implement Kung's attribute-based security policy platform across the same multi-cloud deployments Kung already contemplates (e.g., AWS, Azure, GCP, OpenStack, and VMware vCenter) (Ex. 1007, ¶ 0046) would naturally have looked to Serhiienko's universal tagging ruleset to normalize the provider-native tags Kung already reads into a single, provider-independent tag before Kung's platform selects resources into logical groups.

200. Second, both references relied on the same basic mechanism of tags as the means of triggering automated management action, whether that action is Kung's security policy computation or Serhiienko's resource tagging and retrieval. Kung itself disclosed that its platform is designed from the outset to operate across hybrid computer network infrastructures spanning multiple private and public cloud infrastructure services, including AWS, Azure, GCP, OpenStack, and VMware vCenter (Ex. 1007, ¶¶ 0009–0010, 0046), and a POSITA implementing Kung's attribute-based platform across such a genuinely heterogeneous, multi-provider deployment would have recognized the practical need identified by

Serhiienko: that "each provider implements tagging in a different way" (Ex. 1008, § III.A), such that a security platform relying on provider-native tags would benefit from a normalization mechanism to apply consistent, cross-provider logical-group membership and policy.

201. Third, combining Serhiienko's multi-provider tagging ruleset with Kung's attribute-based security policy platform would have amounted to combining known elements according to known methods to yield the predictable result of a security platform capable of assigning consistent, normalized tags across providers before computing security policy from them.

202. At least because Serhiienko disclosed a working implementation of its Universal Connector, evaluated using both AWS and Google Cloud Platform (Ex. 1008, Abstract; § IV.A), the same two providers Kung already disclosed supporting, (Ex. 1007, ¶¶ 0009–0010, 0046.) a POSITA would have had a reasonable expectation of success in incorporating Serhiienko's multi-provider tagging ruleset into Kung's attribute-based security platform.

2. Claim 1:

Element 1[Preamble]: "A method comprising:"

203. Kung in view of Serhiienko disclosed a method for managing cloud resource tags across multiple cloud providers. The UC in Serhiienko "manages

tags in a secure, audible and cost-effective manner" and "performs best practices cloud management tasks driven by regulated tags." (Ex. 1008, § I.)

Element 1[a]: "receiving, by a cloud-tagging orchestrator service running on a security manager associated with a private network, information regarding each cloud provider of a plurality of cloud providers associated with a cloud environment used by the private network"

204. Kung in view of Serhiienko disclosed or rendered obvious this element because Kung's contextual security platform (security manager) runs on an organization's network infrastructure (associated with a private network) and communicates with various cloud APIs (each cloud provider of a plurality of cloud providers), and Serhiienko's UC (CTOS) receives each provider's list of hosted resources and metadata information across that same multi-provider deployment (receiving … information regarding each cloud provider of a plurality of cloud providers associated with a cloud environment used by the private network).

205. Kung's contextual security platform disclosed the claimed security manager. Kung disclosed the contextual security platform is "implemented as software running on a computer server in communication with various security mechanisms native to an organization's or enterprise's data network infrastructure." (Ex. 1007, ¶ 0011.) Kung explained that "[t]o prevent un-authorized accesses to data and resources available in workloads on an organization's or enterprise's computer network," its platform enforces "network

security policies using native network security mechanisms" within "hybrid computer network infrastructures having both traditional hardware resources and virtual resources provided by private and public cloud infrastructure services." (*Id.*, ¶ 0009.) Because of Kung's disclosure regarding its security software running on "an organization's or enterprise's data network infrastructure," a POSITA would have understood that the security manager is "associated with a private network." (*Id.*, ¶ 0011.)

206. Serhiienko's UC disclosed the CTOS. Serhiienko's UC performs the receiving, identifying, and assigning work the claim requires. Serhiienko explained that its middleware "manages tags in a secure, audible and cost-effective manner" and "performs best practices cloud management tasks driven by regulated tags" across multiple cloud providers. (Ex. 1008, § I.) A POSITA implementing Kung's contextual security platform across the multi-cloud deployment Kung already contemplates, such as Amazon AWS, would have ***run*** Serhiienko's UC to orchestrate its tagging functions.

207. Serhiienko's UC receives information regarding each cloud provider by calling a function "to get list of resources" for every provider in turn, which returns the set of resources that provider hosts. (*Id.*, § III.C, Listing 1.) As a part of the combination, Kung's own disclosure confirmed that Serhiienko's UC is enabled to receive such information as described in the current claim element, as

the same basic receiving mechanism is implemented by Kung at the platform level. Kung's contextual security platform "communicates with the application programming interface of one or more cloud service providers ... using the native application programming interfaces provided by the services," and "request[s] and receive[s]" information "from the API of each of the cloud service provider[s]," including "information on network resources provided by the service." (Ex. 1007, ¶ 0051.) A POSITA implementing Kung's contextual security platform could have ***run*** Serhiienko's UC to perform this receiving function.

208. Kung was explicit that its platform is designed to interact with more than one cloud provider simultaneously, not a single provider addressed repeatedly: "[t]he contextual security platform can support any number of cloud service providers, limited only by the scalability of the particular implementation." (Ex. 1007, ¶ 0051.) Figure 2 in Kung (below) depicted the platform communicating with multiple cloud service provider APIs concurrently, with it illustrating "representative examples of which are indicated by blocks 220, 222 and 224." (*Id.*) Kung separately named real-world examples of these providers: "Amazon AWS, Microsoft Azure and Google GCP" as example public cloud providers and "OpenStack and VMware vCenter" as example private cloud environments. (*Id.*, ¶ 0046.)

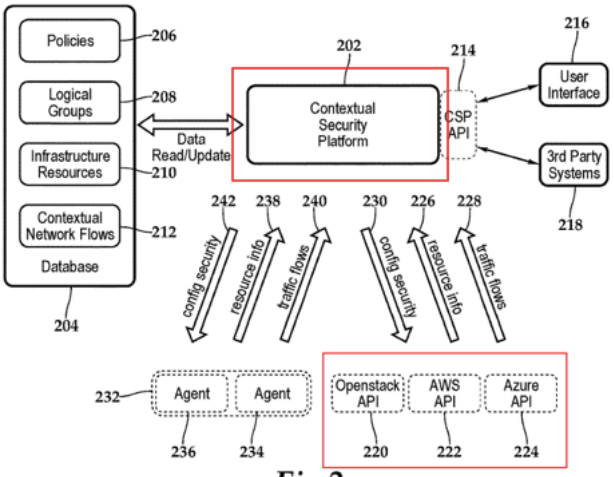


Fig.2

209. Serhiienko confirmed that its UC is likewise capable of operating with a multi-provider system simultaneously, not a single-provider tool applied repeatedly: Serhiienko's tagging algorithm iterates "all resources" within a project or account across multiple cloud service providers, calling functions to retrieve that provider's resource metadata and resource list in turn (Ex. 1008, § III.C, Listing 1), and Serhiienko's own evaluation scenario applied its ruleset across two different providers, AWS and GCP, concurrently. (*Id.*, § IV.A.) A POSITA would have understood that Kung's contextual security platform, running Serhiienko's UC, is genuinely capable of receiving information regarding a plurality of cloud providers at once.

210. Kung and Serhiienko further disclosed that the plurality of cloud providers are "associated with a cloud environment used by the private network." Kung disclosed this aggregate "cloud environment" as the collection of cloud

service providers, taken together, that Kung's contextual security platform centrally manages on behalf of a given organization's or enterprise's network. (Ex. 1007, ¶¶ 0049, 0119.) Kung's platform is "implemented as software running on a computer server in communication with various security mechanisms native to an organization's or enterprise's data network infrastructure" (*id.*, ¶ 0011), and that platform's ability to "support any number of cloud service providers" including public providers and private cloud environments (*id.*, ¶¶ 0046, 0051) means the full set of connected providers, collectively, constitutes the resources available to that organization's network.

211. Serhiienko likewise disclosed that its UC manages resources "across all providers" through a single connector instance for a given organization's deployment. (Ex. 1008, § III.A.) A POSITA would have understood this composite collection of cloud providers to be the claimed "cloud environment used by the private network," as distinct from any single provider's own infrastructure standing alone.

212. Accordingly, Kung in view of Serhiienko disclosed or rendered obvious all elements of limitation 1[a].

Element 1[b][i]: "based on the received information, for each cloud resource of a plurality of cloud resources hosted by the plurality of cloud providers on behalf of the private network"

213. Kung in view of Serhiienko disclosed or rendered obvious this element because Serhiienko's tagging algorithm evaluated, using the metadata each provider returned under Element 1[a] (based on the received information), every resource satisfying its ruleset across every connected provider (for each cloud resource of a plurality of cloud resources), and Kung confirmed those resources are created and hosted on the providers' own infrastructure for the particular owner organization whose network Kung's contextual security platform serves (hosted by the plurality of cloud providers on behalf of the private network).

214. Serhiienko disclosed "based on the received information" by disclosing that, based on the resource and metadata information the UC received from each provider under Element 1[a], its tagging algorithm iterated "∀ metadata ∈ resources_metadata" for every provider evaluating each individual resource in turn against the ruleset's conditions. (Ex. 1008, § III.C, Listing 1).

215. Serhiienko disclosed "a plurality of cloud resources" by disclosing a ruleset defining conditions that determine which resources are tagged and managed, after which the tagging algorithm assembles a "resources_to_tag" set spanning every resource, across every connected provider, to satisfy those conditions. (*Id.*, §§ II.B, III.C, Listing 1.) Serhiienko's evaluation scenario applied

this ruleset to resources hosted across AWS and GCP on behalf of a single organization implementing the connector, retrieving each provider's own resource list. (*Id.*, § IV.A.)

216. Kung confirmed that this kind of resource is created using, and hosted by, the cloud service provider itself. Kung disclosed that infrastructure service providers such as AWS, Azure, GCP, and OpenStack allow tags to be associated with "infrastructure resources created using, and hosted by, the service," and Kung's platform reads those provider-hosted resources' tags "by communicating with the cloud service provider's API." (Ex. 1007, ¶ 0229.) Kung further disclosed that each such resource is associated with an "owner organization," confirming that provider-hosted resources are managed and tagged on behalf of the particular organization that owns them, here the same organization whose network Kung's contextual security platform serves. (*Id.*, Fig. 10, ¶¶ 0011, 0217.)

217. Because Kung's contextual security platform is itself associated with that same organization's computer network, and because Serhiienko's ruleset was applied across a plurality of resources hosted by multiple providers (AWS and GCP) for that same organization, a POSITA would have understood these resources to be "hosted by the plurality of cloud providers on behalf of the private network."

218. Accordingly, Kung in view of Serhiienko disclosed or rendered obvious all elements of limitation 1[b][i].

Element 1[b][ii]: "retrieving, by the cloud-tagging orchestrator service, information associated with the cloud resource"

219. Kung in view of Serhiienko disclosed or rendered obvious this element because Serhiienko's UC (CTOS) called a function to obtain each resource's full metadata from its provider (retrieving information associated with the cloud resource).

220. Serhiienko disclosed that the UC's tagging algorithm iterates for every provider to retrieve that provider's resource metadata and resource list and calls a function to "get resource metadata" for each provider. (*Id.*, § III.C, Listing 1). Serhiienko explained that "by retrieving the entirety of the resource's metadata from its provider, it allows for rules to be created for any field in the metadata." (*Id.*, § III.B.). Because this retrieval obtains substantive information about the resource in the form of the resource's full metadata, Serhiienko disclosed this element.

221. Accordingly, Kung in view of Serhiienko disclosed or rendered obvious all elements of limitation 1[b][ii].

Element 1[c]: "identifying, by the cloud-tagging orchestrator service, a unified tag of a plurality of unified tags for the cloud resource based on a pre-defined global tagging policy"

222. Kung in view of Serhiienko disclosed or rendered obvious this element because Serhiienko's UC (CTOS) matched a resource's metadata against an administrator-authored, cross-provider ruleset (identifying … a unified tag … based on a pre-defined global tagging policy) to determine, from among several rule-specified tags, the one that resource's metadata satisfied (a unified tag of a plurality of unified tags).

223. Serhiienko's UC, functioning as the CTOS, disclosed this "identifying" function directly through its rule-matching mechanism. Serhiienko's tagging algorithm evaluates each resource's metadata against the ruleset and, when a rule's conditions are met, determines the specific tag that rule assigns to that resource before any tag is pushed to the resource: "when rule applied to metadata: resources_to_tag.add the resource from resources list with the tags from rule." (Ex. 1008, § III.C, Listing 1.) This determination is the identifying step where the resource's metadata is matched against the ruleset's conditions to determine which tag key-value pair applies.

224. That rule-satisfaction determination is itself made using a defined logical structure: each rule is composed of conditions related by logical OR, and each condition is composed of statements related by logical AND, such that "the

different conditions have a logical OR relationship, so in order for the rule to be satisfied, at least one condition must be met," while "[t]he statements within a condition have a logical AND relationship, meaning all statements within a condition must be met by a resource simultaneously for it to be compliant with the rule." (Ex. 1008, § III.B.)

225. Serhiienko disclosed this identification step is of a "unified tag." Serhiienko's ruleset mechanism "defines per resource type and provider filter a rule composed of an arbitrary number of conditions plus a set of tag key-value pairs to be applied in case of a match." (*Id.*, § III.B.) Because the same ruleset determines which tag key-value pair applies to a given resource within a project or account, the resulting tag carries a consistent, provider-independent meaning regardless of which provider hosts that resource. Serhiienko does not teach a separate ruleset for each individual cloud service provider. (*Id.*, §§ II.A-B, III.C, IV.B.) Therefore, the resulting tag is a "unified tag," identified by the UC as the outcome of evaluating a resource's metadata against the ruleset's conditions.

226. This element also requires that the tagging policy on which the identification is based be a "pre-defined" tagging policy. Serhiienko disclosed exactly this kind of advance setup: "[i]n order for the auto-generation of tags to be applied, a set of consistency enforcement rules need to be employed by the system's administrator." (*Id.,* § III.B.) Serhiienko confirmed that this ruleset is

authored and loaded before any particular resource is ever evaluated: "we employ the auto-tagging functionality that allows us to define a set of tagging rules in a YAML file," and this pre-authored ruleset is what "the autoGenerate API method" subsequently applies against resources it retrieves at runtime. (*Id.*, § IV.A-B.) This advance configuration, which is fixed in the YAML ruleset before the UC acts on any given resource, makes the resulting policy "pre-defined."

227. Serhiienko disclosed that this same pre-defined policy is a "global" policy because it is applied consistently "across all providers." (*Id.*, § IV.A, Title.) The ruleset "meets the constraints of all providers simultaneously," and the UC provides "a single interface through which to manage tagging across all providers," using "a common API tagging method that abstracts the provider-specific details and provides a uniform interface." (*Id.*, § III.A.) Serhiienko's implementation confirmed this cross-provider scope operationally: "[t]he rules can apply to any metadata field of each individual resource, across all providers," and the autoGenerate method "composes several cloud functions from all providers in order to retrieve all resources that satisfy the ruleset and to apply the necessary tags to them," such that "just one invocation of this function can tag every resource across all managed projects with the right tag." (*Id.*, § IV.A.)

228. This element also requires identifying a unified tag "of a plurality of unified tags." Serhiienko disclosed this plurality because a single ruleset invocation

applies across the UC's entire population of managed resources and rules: "just one invocation of this function can tag every resource across all managed projects with the right tag," where different rules can specify different tag key-value pairs for different resource types, providers, and conditions. (*Id.*, § IV.A-B; § II.B.)

229. Accordingly, Kung in view of Serhiienko disclosed or rendered obvious all elements of limitation 1[c].

Element 1[d]: "assigning, by the cloud-tagging orchestrator service, the unified tag to the cloud resource via an application programming interface (API) of a cloud provider of the plurality of cloud providers hosting the resource"

230. Kung in view of Serhiienko disclosed or rendered obvious this element because Serhiienko's UC (CTOS) called the matched tag's tagResource function (assigning the unified tag to the cloud resource), executed through the specific hosting provider's own native cloud function service (via an API of a cloud provider … hosting the resource).

231. Serhiienko disclosed that once a resource has been matched to a tag under Element 1[c]'s rule-matching determination, "the method then calls the appropriate tagResource function and applies the tag to the resource." (*Id.*, § III.B) Serhiienko's autoGenerate method "composes several cloud functions from all providers in order to retrieve all resources that satisfy the ruleset and to apply the

necessary tags to them." (*Id.*, § IV.A.) A POSITA would have understood this application of the unified tag by Serhiienko's UC to be equivalent to "assigning."

232. Serhiienko disclosed that this tagResource call is carried out through each provider's own API. Serhiienko explained that the UC's management functions, which includes its tagging capabilities, are "implemented for both AWS Lambda and Google Cloud Functions (GCF)," meaning each function is deployed into, and executed using, the specific hosting provider's own native cloud function service. (*Id.*, § IV.A-B.)

233. Figure 1 in Serhiienko (below) depicts this architecture concretely: provider-specific functions running on "AWS Lambda" act on resources ("res") hosted at AWS, while separate provider-specific functions running on "GCP Cloud Functions" act on resources hosted at GCP—with "[t]he functions deployed to the various providers... registered to an event gateway" so that "requests made to the connector reach the functions via the gateway" for the applicable provider. (*Id.*, § IV.A, Fig. 1.)

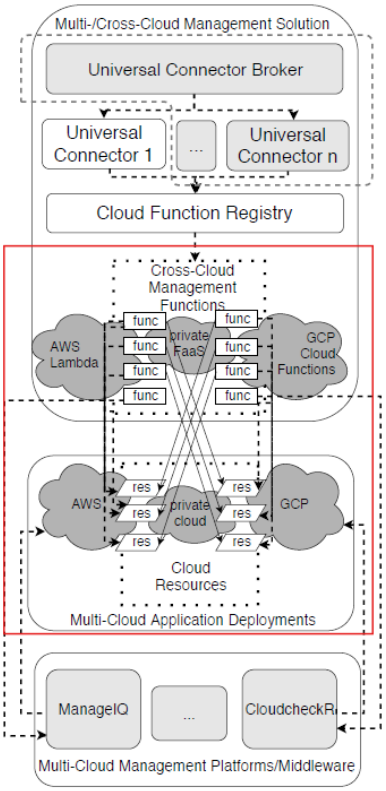


Fig. 1. Universal connector architecture outlining extensibility through FaaS and interoperability with existing management platforms

234. Because each provider's own function acts only on that same provider's own resources using that provider's own native platform, a POSITA would have understood that the tagResource call for a given resource necessarily executes through the API of the specific cloud provider hosting that resource. It is this tagResource call that assigns the unified tag to the resource via the hosting provider's own API.

235. Accordingly, Kung in view of Serhiienko disclosed or rendered obvious all elements of limitation 1[d].

236. Thus, Kung in view of Serhiienko disclosed or rendered obvious this claim.

3. Claim 2:

"The method of claim 1, further comprising facilitating, by the security manager, application of a security policy expressed with reference to one or more unified tags of the plurality of unified tags by retrieving information associated with a particular cloud resource of the plurality of cloud resources from a storage database."

237. Kung in view of Serhiienko disclosed or rendered obvious this claim because Serhiienko's UC maintained a repository of generated tags retrievable on demand (a storage database), and Kung's contextual security platform retrieved tags from that repository to trigger creation of concrete firewall and security-group rules keyed to those tags (facilitating … application of a security policy expressed with reference to one or more unified tags).

238. Serhiienko disclosed the claimed storage database: the UC's ruleset and generated tags are maintained and applied through its own configuration and cloud function deployment, from which resources and their tags can be retrieved on demand. (Ex. 1008, § IV.A-B.)

239. Kung, in turn, disclosed the contextual security platform's role in retrieving tags and applying an actual security policy expressed with reference to them. Kung disclosed that "infrastructure tags" (the same kind of provider-supplied

tag data Serhiienko normalizes into a unified tag) are read "using the API of the service provider during the process of discovering infrastructure resources and used to automatically map discovered infrastructure resources to logical groups." (Ex. 1007, ¶ 0141.) Kung further disclosed that "security policies are represented by logical groups," such that a resource's membership in a logical group (driven by its tags) determines which security policy applies to it (*Id.*, ¶ 0137.) For example, "a logical group PCI defines a selection attribute... This ensures that all resources for application app1 are assigned policies defined for [the] PCI logical group." (*Id.*, ¶ 0144.) Kung's contextual security platform then computes a configuration for network security mechanisms using the security policies and the infrastructure resources that are members of the applicable logical groups. (*Id.*, ¶ 0085.) Kung disclosed that this configuration step is the actual deployment of network security enforcement: "[u]sing infrastructure provider APIs or agents," the "communication between logical groups are use[d] to create security rules in AWS Security groups, OpenStack Security groups, Azure Security Groups, Host Firewalls, etc." (*Id.*, ¶¶ 0154-0155.)

240. A POSITA implementing the combination would have configured the Kung's contextual security platform to retrieve the unified tags generated by the UC in Serhiienko as the attributes used for logical-group membership, in order to facilitate application of these concrete security policies (e.g., the creation and

enforcement of security rules in firewalls and security groups) to the corresponding resource.

241. Thus, Kung in view of Serhiienko disclosed or rendered obvious this claim.

4. Claim 3:

"The method of claim 1, wherein the cloud environment comprises a hybrid multi-cloud environment including a private cloud and a plurality of public clouds."

242. Kung in view of Serhiienko disclosed or rendered obvious this claim because Kung's contextual security platform concurrently managed an on-premise data center (a private cloud) alongside named public cloud service providers (a plurality of public clouds), together forming a hybrid multi-cloud environment.

243. Kung disclosed "a private cloud and a plurality of public clouds." Because a hybrid multi-cloud environment, by definition, includes both private and public clouds, Kung's disclosure of those components necessarily disclosed a "hybrid multi-cloud environment." Kung's contextual security platform operates across environments including "Amazon AWS, Microsoft Azure and Google GCP" as example public cloud providers and "OpenStack and VMware vCenter" as example private cloud environments. (Ex. 1007, ¶ 0046.) Kung's Figure 5 embodiment confirmed that a single contextual security platform concurrently manages both an "on premise data center 502 of an organization or enterprise"

(i.e., example of private clouds) and "a cloud service provider environment 504 that provides virtualized infrastructure resources for use by the organization" (example of public clouds), noting that "any number of environments can be controlled by the contextual security platform." (*Id.*, ¶ 0074.)

244. Serhiienko's UC disclosed managing resources across multiple such providers concurrently through a single ruleset and connector instance. Serhiienko's evaluation scenario applied this single ruleset to resources hosted across AWS and GCP on behalf of a single organization implementing the connector. (Ex. 1008, § IV.A.) A POSITA implementing Serhiienko's UC on Kung's contextual security platform, as discussed above, would have done so across the same hybrid public/private cloud environment Kung already disclosed.

245. Thus, Kung in view of Serhiienko disclosed or rendered obvious this claim.

5. Claim 4:

"The method of claim 1, wherein said assigning further enables visibility and control for the plurality of cloud resources by the security manager."

246. Kung in view of Serhiienko disclosed or rendered obvious this claim because Serhiienko's UC's tag assignment made every connected provider's resources queryable by tag, giving Kung's contextual security platform (security manager) an observable window into resources' current tag state (visibility for the

plurality of cloud resources) and, through logical-group selection and security policy computation from that same tag data, a means to act on those resources at their respective providers (control for the plurality of cloud resources).

247. Consistent with the mapping established above, it is Serhiienko's UC that performs this assigning, and Kung's contextual security platform that must obtain visibility and control as a result of it.

248. Serhiienko's assigning gives Kung's contextual security platform an observable window into the current tag state of resources at every connected provider (visibility): once tags are applied, "[r]etrieval of resources based on their tags … can retrieve information on resources of any type on any provider, based on the tags applied to the resource." (Ex. 1008, § IV.B.) Because that retrieval reflects the same tag data the UC assigned, a POSITA would have understood that querying resources by tag lets an operator observe the actual, current tag state of resources at every connected provider.

249. Serhiienko's assigning likewise gives Kung's contextual security platform the means to determine and carry that tag state into effect at each provider (control): "the capability to manage resources by tag" allows the connector to "invoke a number of resource management cloud functions from both providers" that "act on all resources with the specified tag," so an operator can act on tagged resources at their respective providers based on that same tag data. (*Id.*) Kung's

contextual security platform relies on this same tag data to select resources into logical groups and compute security policy configurations for them, then deploys that configuration back to each resource's provider. (Ex. 1007, ¶ 0085.)

250. Without Serhiienko's UC assigning normalized tags to resources across each provider, there would be no current tag state to observe by tag-based retrieval, defeating visibility, and no consistent, cross-provider tag data on which Kung's contextual security platform could base its logical-group selection and security policy computations, defeating control. It is Serhiienko's UC's assigning step that gives Kung's contextual security platform both an observable window into, and an operative hand on, the tag state of the plurality of cloud resources across every connected provider.

251. Thus, Kung in view of Serhiienko disclosed or rendered obvious this claim.

6. Claim 6:

"The method of claim 1, wherein the plurality of cloud resources comprise virtual machine instances."

252. Kung in view of Serhiienko disclosed or rendered obvious this claim because Serhiienko's ruleset tagged virtual machine resources hosted at AWS and GCP and Kung's contextual security platform likewise managed virtual machine instances (the plurality of cloud resources comprise virtual machine instances).

253. Serhiienko disclosed that its evaluation scenario applied the UC's ruleset to VM resources hosted across AWS and Google Cloud Platform. (Ex. 1008, § IV.A.) Kung likewise disclosed that its contextual security platform manages virtual machine instances as infrastructure resources.

254. Kung confirmed elsewhere that its infrastructure resources include virtual machine instances specifically. Kung disclosed that "[a]n environment managed by the contextual security platform can host nodes with different operating systems that provide different network security mechanisms," and that "[w]hen nodes are virtual machines or containers, their traffic can be enforced by mechanisms running in the hypervisor or operating system hosting the virtual nodes." (Ex. 1007, ¶ 0088.) Kung further disclosed that, in the cloud context specifically, "Amazon AWS offer[s] security groups that can enforce security rules for individual virtual machine instances." (*Id.*, ¶ 0089.)

255. Thus, Kung in view of Serhiienko disclosed or rendered obvious this claim.

7. Claim 7:

"The method of claim 1, wherein the security manager comprises a cloud based security management appliance."

256. Kung in view of Serhiienko disclosed or rendered obvious this claim because Kung's contextual security platform (security manager) runs as server

software that configures native security mechanisms across named public and private cloud infrastructure services (a cloud based security management appliance).

257. Consistent with the mapping established above, Kung disclosed that its contextual security platform is "implemented as software running on a computer server in communication with various security mechanisms native to an organization's or enterprise's data network infrastructure" (Ex. 1007, ¶ 0011), and that this data network infrastructure includes public and private cloud infrastructure services such as Amazon AWS, Microsoft Azure, Google GCP, OpenStack, and VMware vCenter. (*Id.*, ¶ 0046.) Kung's Figure 5 embodiment further confirmed that the contextual security platform concurrently manages resources hosted directly within a cloud service provider environment. (*Id.*, ¶¶ 0074, 0076.) A POSITA would have understood that a security manager implemented as software running on a server in communication with, and configuring security mechanisms native to, such public and private cloud infrastructure services is a "cloud-based security management appliance."

258. Thus, Kung in view of Serhiienko disclosed or rendered obvious this claim.

8. Claim 8:

Element 8[Preamble]: "A non-transitory computer-readable storage medium embodying a set of instructions, which when executed by a processing resource of a security manager of a private network, causes the processing resource to perform a method comprising:"

259. Kung disclosed that its processes may be embodied in machine-executable instructions stored on a "non-transitory machine-readable medium," including semiconductor memory devices such as EPROM and EEPROM, magnetic disks, magneto-optical disks, and CD-ROM/DVD-ROM disks, that are executed by a processor to cause the processor to perform the disclosed processes. (*Id.*, ¶ 0042.) Kung further disclosed "[c]omputer 100 includes a processor 102," which "communicates with a main or working memory 104 and a storage memory 106 over one or more buses." (*Id.*, ¶ 0037.) Consistent with the mapping established in Element 1[a], Kung's processor 102, on which Serhiienko's UC runs to execute the instructions stored on Kung's non-transitory machine-readable medium, forms the processing resource.

260. Kung further disclosed "a security manager of a private network" for the same reasons provided in Element 1[a] above: Kung's contextual security platform is implemented as software running on a computer server in communication with an organization's or enterprise's data network

infrastructure, and it is on that server that the UC runs as the processing resource performing the claimed method. (*Id.*, ¶ 0011.)

Element 8[a]: "receiving information regarding each cloud provider of a plurality of cloud providers associated with a cloud environment used by the private network;"

261. Since this element recites the same requirements as element 1[a], Kung in view of Serhiienko disclosed this element for the same reasons discussed in element 1[a].

Element 8[b][i]: "based on the received information, for each cloud resource of a plurality of cloud resources hosted by the plurality of cloud providers on behalf of the private network:"

262. Since this element recites the same requirements as element 1[b][i], Kung in view of Serhiienko disclosed this element for the same reasons discussed in element 1[b][ii].

Element 8[b][ii]: " retrieving information associated with the cloud resource"

263. Since this element recites the same requirements as element 1[b][ii], Kung in view of Serhiienko disclosed this element for the same reasons discussed in element 1[b][ii].

Element 8[c]: "identifying a unified tag of a plurality of unified tags for the cloud resource based on a pre-defined global tagging policy"

264. Since this element recites the same requirements as element 1[c], Kung in view of Serhiienko disclosed this element for the same reasons discussed in element 1[c].

Element 8[d]: "assigning the unified tag to the cloud resource via an application programming interface (API) of a cloud provider of the plurality of cloud providers hosting the resource."

265. Since this element recites the same requirements as element 1[d], Kung in view of Serhiienko disclosed this element for the same reasons discussed in element 1[d].

9. Claim 9:

"The non-transitory computer-readable storage medium of claim 8, wherein the method further comprises facilitating application of a security policy expressed with reference to one or more unified tags of the plurality of unified tags by retrieving information associated with a particular cloud resource of the plurality of cloud resources from a storage database."

266. For the reasons discussed in element 8[Preamble], Kung in view of Serhiienko disclosed a non-transitory computer-readable storage medium. The remainder of claim 9 recites the same requirements as claim 2, so Kung in view of Serhiienko disclosed this claim for the same reasons discussed in claim 2.

10. Claim 10:

"The non-transitory computer-readable storage medium of claim 8, wherein the cloud environment comprises a hybrid multi-cloud environment including a private cloud and a plurality of public clouds."

267. For the reasons discussed in element 8[Preamble], Kung in view of Serhiienko disclosed a non-transitory computer-readable storage medium. The remainder of claim 10 recites the same requirements as claim 3, so Kung in view of Serhiienko disclosed this element for the same reasons discussed in claim 3.

11. Claim 11:

"The non-transitory computer-readable storage medium of claim 8, wherein said assigning further enables visibility and control for the plurality of cloud resources by the security manager."

268. For the reasons discussed in element 8[Preamble], Kung in view of Serhiienko disclosed a non-transitory computer-readable storage medium. The remainder of claim 11 recites the same requirements as claim 4, so Kung in view of Serhiienko disclosed this element for the same reasons discussed in claim 4.

12. Claim 13:

"The non-transitory computer-readable storage medium of claim 8, wherein the plurality of cloud resources comprise virtual machine instances."

269. For the reasons discussed in element 8[Preamble], Kung in view of Serhiienko disclosed a non-transitory computer-readable storage medium. The

remainder of claim 13 recites the same requirements as claim 6, so Kung in view of Serhiienko disclosed this element for the same reasons discussed in claim 6.

13. Claim 14:

"The non-transitory computer-readable storage medium of claim 8, wherein the security manager comprises a cloud based security management appliance."

270. For the reasons discussed in element 8[Preamble], Kung in view of Serhiienko disclosed a non-transitory computer-readable storage medium. The remainder of claim 14 recites the same requirements as claim 7, so Kung in view of Serhiienko disclosed this element for the same reasons discussed in claim 7.

14. Claim 15:

Element 15[Preamble]: "A multi-tag orchestration system comprising: a processing resource; and a non-transitory computer-readable medium, coupled to the processing resource, having stored therein instructions that when executed by the processing resource cause the processing resource to perform a method comprising:"

271. Kung disclosed or rendered obvious a multi-tag orchestration system. Kung's contextual security platform comprises a computer, its component parts, and the software processes running on it that together discover, tag, and apply security policy to cloud resources across multiple providers. Kung disclosed a computer comprising a processor and memory that communicates with a main or working memory over a bus, where the memory is capable of storing instructions

being executed by the processor, and that such instructions may be stored on a non-transitory machine-readable medium that, when executed, causes the processor to perform the disclosed processes. (Ex. 1007, ¶¶ 0037–0038, 0042.) Kung further disclosed "[c]omputer 100 includes a processor 102," which "communicates with a main or working memory 104 and a storage memory 106 over one or more buses." (*Id.*, ¶ 0037.) Consistent with the mapping established in Element 1[a] and 8[a], Kung's processor 102, on which Serhiienko's UC runs to execute the instructions stored on Kung's non-transitory machine-readable medium, forms the processing resource.

272. For the remaining elements, Kung in view of Serhiienko disclosed those elements for the same reasons discussed in claim 1.

Element 15[a]: "receiving information regarding each cloud provider of a plurality of cloud providers associated with a cloud environment used by a private network;"

273. Since this element recites the same requirements as element 1[a], Kung in view of Serhiienko disclosed this element for the same reasons discussed in element 1[a].

Element 15[b][i]: "based on the received information, for each cloud resource of a plurality of cloud resources hosted by the plurality of cloud providers on behalf of the private network:"

274. Since this element recites the same requirements as element 1[b][i], Kung in view of Serhiienko disclosed this element for the same reasons discussed in element 1[b][ii].

Element 15[b][ii]: "retrieving information associated with the cloud resource"

275. Since this element recites the same requirements as element 1[b][ii], Kung in view of Serhiienko disclosed this element for the same reasons discussed in element 1[b][ii].

Element 15[c]: "identifying a unified tag of a plurality of unified tags for the cloud resource based on a pre-defined global tagging policy"

276. Since this element recites the same requirements as element 1[c], Kung in view of Serhiienko disclosed this element for the same reasons discussed in element 1[c].

Element 15[d]: "assigning the unified tag to the cloud resource via an application programming interface (API) of a cloud provider of the plurality of cloud providers hosting the resource."

277. Since this element recites the same requirements as element 1[d], Kung in view of Serhiienko disclosed this element for the same reasons discussed in element 1[d].

15. Claim 16:

"The system of claim 15, wherein the method further comprises facilitating application of a security policy expressed with reference to one or more unified tags of the plurality of unified tags by retrieving information associated with a particular cloud resource of the plurality of cloud resources from a storage database."

278. For the reasons discussed in element 15[Preamble], Kung in view of Serhiienko disclosed a system. The remainder of claim 16 recites the same requirements as claim 2, so Kung in view of Serhiienko disclosed this element for the same reasons discussed in claim 2.

16. Claim 17:

"The system of claim 15, wherein the cloud environment comprises a hybrid multi-cloud environment including a private cloud and a plurality of public clouds."

279. For the reasons discussed in element 15[Preamble], Kung in view of Serhiienko disclosed a system. The remainder of claim 17 recites the same requirements as claim 3, so Kung in view of Serhiienko disclosed this element for the same reasons discussed in claim 3.

17. Claim 20:

"The system of claim 15, wherein the plurality of cloud resources comprise virtual machine instances."

280. For the reasons discussed in element 15[Preamble] Kung in view of Serhiienko disclosed a system. The remainder of claim 20 recites the same

requirements as claim 6, so Kung in view of Serhiienko disclosed this element for the same reasons discussed in claim 6.

F. Ground 5: Combination of Kung, Serhiienko, and Whitlock renders claims 5, 12, and 19 obvious.

1. Motivations to combine

281. A POSITA would have been motivated to combine Whitlock's information tag-set teachings such as location and function architecture with the combination of Kung and Serhiienko, with a reasonable expectation of success, for several reasons.

282. First, Kung and Serhiienko already identified a unified tag using information, location-related, and function-related resource attributes. Kung disclosed that its platform reads provider-native resource properties such as "region," "availability zone," "account," and "network" to select a resource into a logical group (Ex. 1007, ¶¶ 0145–0148.) while Serhiienko disclosed that its ruleset can key on "any metadata field of each individual resource, across all providers." (Ex. 1008, § IV.A.) Whitlock disclosed these same categories of attributes under their own express, named tag-set headings: a "Location" tag set with example tags for New York, Portland, and Tokyo, and a "Primary Function" tag set identifying an asset's functional role. (Ex. 1005, ¶¶ 0072–0073, Fig. 8.) A POSITA would have looked to Whitlock's named location and function tag-set categories as a

further model for organizing the same kinds of resource attributes that Kung's platform already reads and Serhiienko's ruleset already evaluates.

283. Second, both Whitlock and the combined Kung/Serhiienko platform address the same basic problem of classifying resources for management purposes using attributes beyond a bare resource identifier. Whitlock's tag-based classification techniques allow a user to "easily and quickly select and perform actions on groups of one or more assets" based on "node properties or user-selected criteria or conditions." (Ex. 1005, ¶ 0040.) The Kung and Serhiienko combination likewise selects and manages resources based on provider-reported attributes and ruleset-matched metadata fields. (Ex. 1007, ¶¶ 0145–0148; Ex. 1008, § IV.A.) A POSITA would have recognized that Whitlock's location and function tag-set categories are simply named instances of the same kind of resource-attribute-based classification Kung and Serhiienko already perform.

284. Third, applying Whitlock's named location and function tag-set categories to the Kung and Serhiienko combination would have amounted to applying a known technique (categorizing resources by location and function, as disclosed in Whitlock) to the Kung and Serhiienko combination to yield the predictable result of a platform that identifies a unified tag using those same categories.

285. At least because Whitlock disclosed a fully implemented, working tag-set classification tool that assigns location and function categories to real, monitored IT assets (Ex. 1005, ¶¶ 0072–0073, Fig. 8), a POSITA would have had a reasonable expectation of success in incorporating Whitlock's location and function tag-set categories into the Kung and Serhiienko combination.

2. Claim 5:

"The method of claim 1, wherein the pre-defined global tagging policy identifies the unified tag based on one or more of the information regarding the cloud resource, a location of the resource, and a function of the cloud resource."

286. Kung in view of Serhiienko disclosed or rendered obvious this claim because Serhiienko's ruleset identified the unified tag by keying on metadata fields of the cloud resource, which a POSITA would have understood to include the resource's information, location, and function. To the extent needed, Whitlock independently disclosed or rendered obvious this claim because Whitlock tagged assets using harvested configuration information (the information regarding the cloud resource), a named "Location" tag set (a location of the resource), and a named "Primary Function" tag set (a function of the cloud resource).

287. Serhiienko disclosed that its ruleset conditions can key on "any metadata field of each individual resource, across all providers" (Ex. 1008, § IV.A.) Serhiienko's example ruleset conditions key specifically on resource type

and provider (*id.*, § III.B), while Serhiienko's retrieval function obtains "the entirety of the resource's metadata from its provider" (*id.*, § III.B), such that a POSITA would have understood a cloud resource's information, location or function as metadata fields commonly returned by a cloud provider's resource metadata API to be keyed on using this same mechanism.

288. To the extent that Serhiienko does not itself disclose information, location, or function as specific bases for identifying the unified tag, Whitlock disclosed each of these three bases directly and by name.

289. As to information regarding the cloud resource, Whitlock's tag-based classification tool tags assets using information "about the asset's operating system, database version, IP address, and/or other such[] configuration information," obtained by harvesting data directly from the assets. (Ex. 1005, ¶ 0041.)

290. As to location, Whitlock disclosed a "Location" tag set, illustrating that "tag set display 810 is named 'location' and refers to tags that are interrelated to each other because the tags each indicate an asset's location," with "specific tags for 'New York', 'Portland', and 'Tokyo,'" indicating that a tagging profile can be configured to automatically apply a location tag to a node meeting specified conditions. (Ex. 1005, ¶¶ 0067–0068, 0073, Fig. 13.)

291. As to function, Whitlock disclosed that a tag set may reflect an asset's functional role, listing "Primary Function" as a tag set category alongside "Location," "Owner," and "Platform Family." (*Id.* Fig. 13.) A POSITA would have understood that Whitlock's "Primary Function" tag set identifies a resource's functional role in the same manner as the administrator-defined, condition-based mechanism Whitlock applied to location tagging.

292. A POSITA implementing the UC's ruleset would have configured it to identify the unified tag based on a resource's information, location, or functional role using the same tag-set categories Whitlock disclosed, just as it would key on resource type or provider. Because Whitlock's tag-set categorization scheme and Serhiienko's provider-agnostic, metadata-based ruleset both operate on the same basic principle of classifying a resource using an attribute harvested from (or reported by) the resource itself or its managing platform, a POSITA would have had no difficulty applying Serhiienko's ruleset mechanism to Whitlock's location and function tag-set categories.

293. Thus, Serhiienko disclosed or rendered obvious this claim, and Whitlock provides further, independent support for this claim in the proposed combination of Kung, Serhiienko, and Whitlock.

3. Claim 12:

"The non-transitory computer-readable storage medium of claim 8, wherein the pre-defined global tagging policy identifies the unified tag based on one or more of the information regarding the cloud resource, a location of the resource, and a function of the cloud resource."

294. For the reasons discussed in element 8[Preamble], Kung in view of Serhiienko disclosed a non-transitory computer-readable storage medium. The remainder of claim 12 recites the same requirements as claim 5, so Kung in view of Serhiienko, in further view of Whitlock, disclosed this claim for the same reasons discussed in claim 5.

4. Claim 19:

"The system of claim 15, wherein the pre-defined global tagging policy identifies the unified tag based on one or more of the information regarding the cloud resource, a location of the resource, and a function of the cloud resource."

295. For the reasons discussed in element 15[Preamble], Kung in view of Serhiienko disclosed a system. The remainder of claim 19 recites the same requirements as claim 5, so Kung in view of Serhiienko, in further view of Whitlock, disclosed this claim for the same reasons discussed in claim 5.

G. Ground 6: Combination of Kung, Serhiienko, and Juniper renders claim 18 obvious.

1. Motivations to combine

296. A POSITA would have been motivated to look to Juniper, and to present Kung's security manager's centralized visibility and control through a single pane of glass user interface as Juniper disclosed, for several reasons.

297. First, Kung in view of Serhiienko already disclosed that its combined system provides centralized visibility and control over tagged resources across multiple connected providers (as discussed above in connection with claim 4). But Kung's and Serhiienko's disclosures are described principally in terms of tag creation, retrieval, and management for individual resources and providers, without expressly describing a single consolidated view spanning all providers at once. Juniper fills this gap by disclosing that a multi-cloud management platform can and should present its centralized, cross-provider information through exactly such a consolidated, single pane of glass interface. (Ex. 1006, ¶¶ 0032, 0035.)

298. Second, both references address the same administrative burden created by managing resources across multiple, heterogeneous cloud providers. Juniper's own stated purpose was to spare an administrator from needing to interface separately with each cloud's own controller. (*Id.*, ¶¶ 0010, 0035.) A POSITA would have recognized that pairing the combined Kung/Serhiienko

platform's centralized tag data with Juniper's single pane of glass presentation layer would directly further both references' shared goal of unified, cross-provider administration.

299. Third, combining Juniper's single pane of glass presentation with the combined Kung/Serhiienko platform's centralized, cross-provider tag data would have amounted to applying a known technique (consolidating multi-cloud administration into a single controller interface in Juniper) to a known platform (the combined Kung/Serhiienko platform, which already aggregates cross-provider tag information) to achieve the predictable result of a unified administrative view.

300. At least because Juniper disclosed a well-known, working means of presenting multi-cloud resource information through a single controller interface, a POSITA would have had a reasonable expectation of success in presenting the combined Kung/Serhiienko platform's centralized, cross-provider tag data through such an interface.

2. Claim 18:

"The system of claim 15, wherein said assigning enables visibility and control of the plurality of cloud resources via a single pane of glass user interface"

301. Kung, Serhiienko, and Juniper disclosed or rendered obvious this claim because Serhiienko's assigning step already gave its UC centralized visibility and control over resources at every connected provider (as shown in

claim 4 above), and Juniper disclosed presenting that same kind of centralized, cross-provider control to an administrator through a single, unified interface (a single pane of glass user interface).

302. For the reasons discussed in element 15[Preamble], Kung in view of Serhiienko disclosed the system.

303. Kung in view of Serhiienko already disclosed that the UC's assigning gives the combined system centralized visibility and control over tagged and untagged resources across every connected provider. (Ex. 1007, ¶ 0085; Ex. 1008, § IV.A-B.) Juniper disclosed presenting exactly this kind of centralized, cross-provider resource information through a single pane of glass: "[a] centralized controller ... that allows one administrator, with a single pane of glass, to control the interconnections and security within [a] multi-cloud network," such that "the network administrator may interface with only [the] controller ... via a single pane of glass to establish the communication links" across every managed cloud. (Ex. 1006, ¶¶ 0032, 0035.) A POSITA combining Kung and Serhiienko with Juniper would have presented the contextual security platform's centralized, cross-provider visibility and control through Juniper's single pane of glass interface.

304. Thus, Kung in view of Serhiienko, in further view of Juniper, disclosed or rendered obvious this claim.

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; that these statements were made with knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under 18 U.S.C. § 1001; and further that such willful false statements may jeopardize the validity of the application or any patent issued thereon. I declare under penalty of perjury under the laws of the United States of America that the foregoing is true and correct.

Executed on 2026-09-14 in Potomac, MD.



Paul D. Martin, Ph.D.

APPENDIX A

harborexperts

Paul D. Martin, Ph.D.

Chief Scientist at Harbor Experts
Lecturer at Johns Hopkins University

P: 443.449.9006
E: paul@harborexperts.com

Profile

Dr. Martin has testified more than **19 times** in complex software- and hardware-related litigation, including **13 depositions**, **4 trials**, and **2 evidentiary hearings**. He has led and managed more than 100 large-scale source code reviews involving sophisticated software and hardware systems. Dr. Martin is the Chief Scientist at Harbor Experts and a Lecturer in the Department of Computer Science at Johns Hopkins University, where he teaches courses on software development, computer and network security, and cryptography. His research interests include embedded system security, operating system security, vulnerability analysis, reverse engineering, network protocol analysis, applied cryptography, and privacy-preserving protocols.

Education

2011-2016	*Ph.D. Computer Science, Johns Hopkins University, Baltimore, MD*
2011-2013	*M.S.E. Computer Science, Johns Hopkins University, Baltimore, MD*
2007-2011	*B.S. Computer Science, Johns Hopkins University, Baltimore, MD*

Industry Experience

2024-Present	*Harbor Experts*	**Chief Scientist**
2023-Present	*Johns Hopkins University*	**Lecturer**
2023-2024	*Harbor Labs*	**Vice President of Applied Research and Technology**
2018-2024	*Harbor Labs*	**Director of Firmware Security, Senior Research Scientist**
2013-2018	*Harbor Labs*	**Research Scientist**
2011-2016	*Johns Hopkins University*	**PhD Candidate, Research Assistant**
2013	*Applied Communication Sciences*	**Graduate Intern**
2011	*University of Michigan* *Inter-university Consortium for Political and Social Research* (ICPSR)	**Penetration Tester**
2009-2011	*Independent Security Evaluators*	**Security Intern**
2008-2010	*Johns Hopkins University* *Digital Research and Curation Center* (DRCC)	**Student Software Developer**
2008	*Brandeis University*	**Freelance Software Developer**

Teaching Experience

2025	*Intermediate Programming*	**Lecturer**
2024	*Security and Privacy in Computing*	**Lecturer**
2023	*Security and Privacy in Computing*	**Lecturer**
2015	*Introduction to Hardware Hacking*	**Instructor**
2012-2014	*Security and Privacy*	**Teaching Assistant**
2011	*Practical Cryptographic Systems*	**Course Assistant**

Netskope Ex. 1002
Netskope v. Fortinet IPR2026-00473
Page 00143

harborexperts

Publications

P. Martin, D. Russel, M. Ben Salem, S. Checkoway, A. Rubin, Sentinel: Secure Mode Profiling and Enforcement for Embedded Systems, Proc. ACM/IEEE International Conference on Internet-of-Things Design and Implementation, (IoTDI '18).

P. Martin, M. Rushanan, T. Tantillo, C. Lehmann and A. Rubin, Applications of Secure Location Sensing in Healthcare. In the proceedings of ACM Conference of Bioinformatics, Computational Biology, and Health Informatics (BCB '16).

J. Carrigan, P. Martin, M. Rushanan, KBID: Kerberos Bracelet Identification. In the Proceedings of Financial Cryptography and Data Security (FC '16).

P. Martin, M. Rushanan, S. Checkoway, M. Green, A. Rubin. Classifying Network Protocol Implementation Versions: An OpenSSL Case Study. Technical Report 13-01, Johns Hopkins University (December 2013).

P. Martin, A. Rubin, and R. Bhatti, *Enforcing Minimum Necessary Access in Healthcare Through Integrated Audit and Access Control.* In ACM Conference on Bioinformatics, Computational Biology, and Biomedical Informatics Health Informatics Symposium (BCB-HIS), (September 2013)

Professional Committees & Standards Bodies

IEEE 7024 Working Group — "Standard for the Procurement, Verification and Validation, and Life Cycle Management of Forensic Technologies" (Current Member)
IEEE Symposium on Security and Privacy (Winter 2024)
IEEE Symposium on Security and Privacy (Summer 2024)

Patents

System and Method for automatically extracting information from binary files for use in Database Queries	**US 10,762,214 B1**
System and method for network traffic profiling and visualization	**US 10,230,599 B2**
System and method for network traffic profiling and visualization	**US 9,667,521 B2**
Healthcare privacy breach prevention through integrated audit and access control	**US 8,984,583 B2**
Healthcare privacy breach prevention through integrated audit and access control	**US 9,438,632 B2**

Current Research

Quantifying vulnerability divergence and patch lead time between rolling and enterprise Linux distributions, including whether enterprise distributions should share soft-forked package versions.
Performance analysis and optimization of local large language model inference, including forecasting of architecture and parameter-scaling trends.
Use of multimodal large language models for diagnosis of disease progression.
Use of large language models for certain types of technical research and analysis.

Expert Witness Engagements

Evox Productions, LLC v. Cars.com, LLC

Case:	Case # 5:25-cv-02357-KK-SP
Description:	Analysis related to web architecture.
Services:	Source code review, expert report drafting, comparative analysis.
Expert Testimony at Deposition:	Potomac, MD (August 26, 2026)

Malikie Innovations Ltd, Key Patent Innovations Ltd v. Foundry, Inc.

Case:	Case # 7:25-CV-00567-DC-DTG
Description:	Analysis related to cryptography, elliptic curve cryptography, hardware accelerated cryptography.
Services:	Source code review, claim construction, expert report drafting, validity analysis, infringement analysis.
Expert Testimony at Deposition:	Potomac, MD (July 31, 2026)

The Nielsen Company v. TVision

Case:	Case #. 25-575-CJB
Description:	Analysis related to audience measurement and identification systems and audio fingerprinting systems.
Services:	Source code review, expert report drafting, source code and algorithm analysis and simulations.
Expert Testimony at Trial:	Wilmington, DE (June 8-9, 2026)

CERTAIN FLASH-SPUN NONWOVEN MATERIALS AND PRODUCTS CONTAINING SAME

Case:	ITC Investigation No. 337-TA-1424
Description:	ITC Investigation related to nation-state threat actors, unfair competition and trade secret misappropriation.
Services:	Malware analysis and reverse engineering, cyber-attribution, computer forensics, reasonableness of security measures.
Expert Testimony at Deposition:	Washington, DC (September 6, 2025)
Expert Testimony at Trial:	Washington, DC (January 29, 2026)

Malikie Innovations Ltd, Key Patent Innovations Ltd v. MARA Holdings, Inc.

Case:	Case # 7:25-cv-00222-DC-DTG
Description:	Analysis related to cryptography, elliptic curve cryptography, random number generation and cryptocurrencies.
Services:	Source code review, expert report drafting, validity analysis.
Expert Testimony at Deposition:	Potomac, MD (January 22, 2026)

Bayerische Motoren Werke Aktiengesellschaft v. Onesta IP, LLC

Case:	Case # 6:2025-cv-00581
Description:	Analysis related to power and latency-sensitive process scheduling.
Services:	Expert declaration drafting, patent analysis.

Toyota Motor Corp v. **AutoConnect Holdings LLC**

Case:	PGR2025-00041
Description:	Analysis related to cloud-based vehicle management and security systems.
Services:	Expert report drafting, validity analysis.

Twitch Interactive, Inc. v. Razdog Holdings LLC,

Case:	IPR2025-03017
Description:	Analysis related to cloud-based content aggregation and streaming.
Services:	Invalidity analysis. IPR drafting.
Expert Testimony at Deposition:	Washington, DC (October 10, 2025)

The Nielsen Company v. Hyphametrics, Inc.

Case:	Case # 23-136-GBW-CJB
Description:	Analysis related to audience measurement and identification systems and machine learning systems.
Services:	Source code review, expert report drafting, infringement

analysis.
Expert Testimony at Deposition: Potomac, MD (February 5, 2025)
Expert Testimony at Trial: Wilmington, DE (July 24-25, 2025)

Jonathan Day and Michelle Dobek v. **Advanced Micro Devices, Inc**.
Case: Case # 3:22-cv-04305
Description: Litigation related to computer fTPM and TEE design characteristics.
Services: Technical analysis and expert reports on security and performance characteristics of trusted execution environments and the design of NVRAM interfaces through SPI. Drafting of expert reports.
Expert Testimony at Deposition: Washington, DC (February 7, 2025)

State of New Jersey v. **Paul Caneiro**
Case: Indictment # 19-02-283; PG # 18-4915
Description: Analysis related to reliability of specific types of computerized DNA analysis in criminal proceedings.
Services: Source code review, expert report drafting.
Expert Testimony at Hearing: Freehold, NJ (December 6, 2024)

Amazon.com, Inc. v. B.S.D. Crown, Ltd.,
Case: IPR2025-00057
Description: Analysis related to emulation and virtualization technology.
Services: Invalidity analysis. IPR drafting.

MediaTek Inc. v. Daedalus Prime LLC
Case: IPR2025-00100
Description: Analysis related to security processors embedded in microprocessors.
Services: Invalidity analysis. IPR drafting.

McKinney v. **Corsair Gaming, Inc**.
Case: Case # 4:22-cv-00312-JST
Description: Litigation related to computer memory performance.
Services: Technical analysis and expert reports on characteristics of computer memory and forensic analysis of computer systems.

LoganTree v. Fossil
Case: Case # 21-cv-0385-JDW
Description: Litigation related to fitness tracking wearables.
Services: Technical analysis and expert reports on validity and infringement.
Expert Testimony at Deposition: Monrovia, MD (August 11, 2023)

Nielsen v. TVSquared
Case: Case # 6:22-cv-00244-ADA
Description: Litigation related to network-based audience measurement.
Services: Technical analysis and expert reports on patentable subject matter.

ISI v. **Intuitive Surgical, Inc.**
Case: Case # 3:21-cv-03496-VC
Description: Litigation related to cryptographic security protections.
Services: Technical analysis and expert reports on security and technical aspects of surgical devices.
Expert Testimony at Deposition: Monrovia, MD (March 16, 2023)

Healthcare Advanced Risk Technologies, Inc. and Inspirien Holdings Corp. v. **Terrence Mills, AI.io. Corp., Jane Nemcova and VEUU, Inc.**

Case:	Case # 2:22-cv-04192-JS-AYS
Description:	Litigation related to timeline of software development and misappropriation of trade secrets.
Services:	Source code and documentation review, source code copying analysis, expert report drafting.

Communication Technologies, Inc. v. Samsung Electronics America, Inc., and Samsung Electronics Co. Ltd.

Case:	Case # 2:21-cv-444-JRG
Description:	Litigation related to patents on secure erase processes.
Services:	Source code review of algorithms related to secure erase processes, declaration on aspects of source code review process and representativeness of source code

Planck, LLC D/B/A Patch Media v. **Particle Media, Inc. D/B/A News Break, Et. Al.**

Case:	Case # 20-cv-10959 (LGS)
Description:	Litigation related to copyright.
Services:	Review of web scraping facts, contracts and agreements and syndicated web feed technology, declaration on web scraping and syndicated web feed technology

WSOU Investments, LLC D/B/A Brazos Licensing and Development v. Cisco, Inc.

Case:	Case # 6:21-cv-00128-ADA
Description:	Litigation related to patents to patents on wireless network handoff, network management and authentication technologies.
Services:	Source code review of accused products, declaration on aspects of source code review process.

United States **v. Laffon Ellis**

Case:	Case # 2:19-cr-00369-DWA
Description:	Analysis related to reliability of specific types of computerized DNA analysis in criminal proceedings.
Services:	Source code review, expert report drafting.
Expert Testimony at Hearing:	Monrovia, MD (December 20, 2021)

Sysmex Corporation and Sysmex America, Inc. v. **Beckman Coulter, Inc.**

Case:	CA # 19-1642-RGA-CJB
Description:	Litigation related to hematology analysis machine patents.
Services:	Source code review. Expert report drafting.
Expert Testimony at Deposition:	Monrovia, MD (November 22, 2021)

CERTAIN ROUTERS, ACCESS POINTS, CONTROLLERS, NETWORK MANAGEMENT DEVICES, OTHER NETWORKING PRODUCTS, AND HARDWARE AND SOFTWARE COMPONENTS THEREOF

Case:	ITC Investigation No. 337-TA-1227
Description:	ITC Investigation related to patents on wireless network handoff, network management and QoS technologies.
Services:	Source code review, validity and prior art analysis, expert report drafting.
Expert Testimony at Deposition:	Monrovia, MD (June 9-10, 2021)
Expert Testimony at Trial:	Washington, DC (July 28, 2021)

Micro Focus, Inc. v. Insurance Services Organization

Case:	DE Civil Action # 15-252-RGA
Description:	Litigation related to unlicensed use of runtime environments, libraries and software compilers.

Netskope Ex. 1002
Netskope v. Fortinet IPR2026-00473
Page 00147

Services:	Source code review. Binary reverse engineering and analysis, affidavit drafting, expert report drafting.
Expert Testimony at Deposition:	Wilmington, DE (Feb 2, 2021)

loanDepot.com, LLC v. Sigma Infosolutions, Inc.

Case:	AAA Case # 01-18-0001-5821
Description:	Litigation related to software development practices.
Services:	Source code analysis, experimentation, report drafting.
Expert Testimony at Deposition:	Baltimore, MD (December 17, 2019)

Cypress Lake Software, Inc. v. Samsung Electronics America and Dell, Inc.

Case:	Case # 6:18-cv-00030-RWS
Description:	Litigation related to infringement of UX patents.
Services:	Source code analysis, report drafting.
Expert Testimony at Deposition:	Baltimore, MD (July 9, 2019)

Apple, Inc. Device Performance Litigation

Case:	CA Civil Action # 18-md-02827-EJD
Description:	Litigation related to business practices.
Services:	Technical analysis and expert reports on security and technical aspects of mobile phone forensics.

Italian Antitrust Authority v. **Apple, Inc.**

Case:	PS/11309
Description:	Litigation related to business practices.
Services:	Technical analysis and expert reports on security and technical aspects of software update processes.

Carl Zeiss AG and ASML Netherlands B.V. v. Nikon

Case:	Case # 2:17-cv-07083-RGK (MRWx)
Description:	Litigation related to patents on image detection algorithms.
Services:	Source code review of algorithms related to image processing and detection algorithms, declaration on aspects of source code review process.

Decision Resources, LLC v. **Brigham Hyde, Precision Health Intelligence, LLC and Orr Inbar**

Case:	MA Civil Action # 17-2834J
Description:	Litigation related to timeline of software development and misappropriation of trade secrets.
Services:	Source code and documentation review, development timeline analysis, affidavit drafting.

Litigation Support

Video Solutions v. Cisco, Inc

Case:	Case # 2:23-cv-222-JRG
Description:	Litigation related to videoconferencing and network performance optimization.
Services:	Infringement analysis. Source code review.

OmniTracs, LLC v. Motive Technologies, Inc.

Case:	Case # 3:23-cv-05261-RFL
Description:	Litigation related to driver fleet management.
Services:	infringement analysis. Source code review.

Robocast v. **Netflix, Inc**

Case:	Case # 1:22-cv-00305-JLH-CJB
Description:	Litigation related to hypermedia and multimedia node

structures.

Services: Invalidity and noninfringement analysis and report drafting. Source code review.

Proxense, LLC. v. Google, LLC

Case: Case # 6:23-cv-00320-ADA
Description: Litigation related to patents on biometrics and payment processing.
Services: Claim construction analysis and report drafting.

Epic Games, Inc. & Anor vs. **Apple Inc & Anor**

Case: Case # NSD 1236 of 2020
Description: Litigation related to security and competition.
Services: Document review, interviews, report drafting.

Softex LLC v. Absolute Software Corporation, Dell Technologies Inc, and HP Inc**.**

Case: Case # 1:22-cv-01309-DAE
Description: Litigation related to computer anti-theft technology.
Services: Claim construction analysis, source code review, report review, binary reverse engineering and product testing.

Mediapointe, Inc v. Microsoft Corporation

Case: Case # 2:22-cv-01009-MCS-MRW
Description: Litigation related to content delivery networks.
Services: Claim construction analysis, source code review, validity analysis, infringement analysis, report drafting.

Mediapointe, Inc v. Akamai Corporation

Case: Case # 2:22-cv-06233-MCS-AFM
Description: Litigation related to content delivery networks.
Services: Claim construction analysis, source code review, validity analysis, infringement analysis, report drafting.

Lauri Valjakka v. **Netflix, Inc**

Case: Case # 4:22-cv-01490-JST
Description: Litigation related to content delivery networks.
Services: Claim construction analysis, invalidity analysis, noninfringement analysis, report review.

Fanduel, Inc v. Winview, Inc

Case: IPR 2022-01306, IPR-01307
Description: Litigation related to patents on probability systems.
Services: Document review, invalidity analysis, report review.

Proxense, LLC. v. Samsung Electronics America, Inc., and Samsung Electronics

Case: Case # 6:21-cv-00210-ADA
Description: Litigation related to patents on biometrics and payment processing.
Services: Source code review, documentation review, product testing, validity analysis, infringement analysis, report drafting.

Global Eticket Exchange Ltd. vs. **TicketMaster LLC**

Case: Case # 6:21-cv-00399-ADA
Description: Litigation related to patents on electronic ticketing.
Services: Source code review, documentation review, product testing, invalidity analysis, non-infringement analysis, report drafting.

US Dominion, Inc. vs. Fox News Network

Case:	Case # N21C-03-257-EMD
Description:	Litigation related to defamation and voting machine security.
Services:	Source code review, documentation review, product testing, report drafting.

Centripetal Networks, Inc. vs. **Keysight Technologies, Inc.**

Case:	Case # 2:22-cv-0002-AWA-DEM
Description:	Litigation related to patents on network monitoring devices and security gateways.
Services:	Source code review, documentation review, product testing, noninfringement analysis, infringement analysis.

Milliman, Inc and Vigilytics LLC, vs. Gradient A.I. Corp.

Case:	Case # 1:21-cv-10865-NMG
Description:	Litigation related to breach of contract and source code copying.
Services:	Source code review, documentation review.

WSOU Investments, LLC D/B/A Brazos Licensing and Development v. Microsoft Corporation

Case:	Case # 6:20-cv-00464-ADA, 6:20-cv-00460-ADA, 6:20-cv-00457-ADA,
Description:	Litigation related to patents on telephony management systems and skill-based matchmaking.
Services:	Source code review, documentation review, validity analysis, infringement analysis, report drafting.

10Tales Inc. v. TikTok PTE. Ltd.

Case:	Case # 1:18-cv-826-WCB
Description:	Litigation related to patents on user-adapted video streams.
Services:	Claim construction analysis.

Carriere v. **Symantec Corporation**

Case:	Case # 500-06-000894-176
Description:	Class action litigation related to product security.
Services:	Source code review, documentation review, report drafting.

IOENGINE, LLC v. Ingenico, Inc.

Case:	Case # 1:18-cv-826-WCB
Description:	Litigation related to patents on payment processing systems.
Services:	Source code review, documentation review, validity analysis, infringement analysis, report drafting.

IOENGINE, LLC v. PayPal Holdings, Inc.

Case:	Case # 1:18-cv-452-WCB
Description:	Litigation related to patents on payment processing systems.
Services:	Source code review, documentation review, validity analysis, infringement analysis, report drafting.

AGIS Software Development LLC v. **Uber Technologies**

Case:	Case # 2:21-cv-00026-JRG-RSP
Description:	Litigation related to patents on map overlays and messaging systems.
Services:	Source code review.

Finjan v. **Palo Alto Networks**

Case:	Case # 4:14-CV-04908-PJH
Description:	Litigation related to patents on malware scanning gateways.
Services:	Invalidity analysis, Claim construction analysis, source code review.

Huawei Technologies Co. v. **Verizon Communications Inc.**

Case:	Case # 6:20-CV-00090
Description:	Litigation related to patents on malware scanning gateways with cloud components.
Services:	Source code review, non-infringement analysis.

Epic Games, Inc. vs. **Apple, Inc.**

Case:	Case # 4:20-cv-05640-YGR-TSH
Description:	Litigation related to security and competition.
Services:	Document review, interviews, report drafting.

Philips North America LLC ; Koninklijke Philips N.V. vs. **Summit Imaging Inc.**

Case:	Case # 2:19-cv-01745-JLR
Description:	Litigation related to third-party repair services.
Services:	Source code review, document review, report drafting.

California Physicians Service, Inc D/B/A Blue Shield of California vs. Healthplan Services Inc,

Case:	Case # 3:18-cv-3730
Description:	Litigation related to software development practices and breach of contract.
Services:	Document review, source code review, report drafting.

Finjan v. **Qualys**

Case:	Case # 4:18-cv-07229-YGR
Description:	Litigation related to patents on vulnerability assessment products.
Services:	Invalidity analysis, Non-infringement analysis, source code review, report drafting.

Finjan v. **Sonicwall**

Case:	Case # 5:17-cv-04467-BLF-HRL
Description:	Litigation related to patents on malware scanning gateways.
Services:	Invalidity analysis, Non-infringement analysis, source code review, report drafting.

TecSec Inc. v. Cisco and Oracle

Case:	Case # 1:10-cv-115 LO-TCB
Description:	Litigation related to patents on hardware accelerated cryptographic processors.
Services:	Infringement analysis, validity analysis, source code review, report drafting.

Blackberry Limited v. Facebook, Inc.

Case:	Case # 2:18-cv-01844-KSx
Description:	Litigation related to patents on agent-based network monitoring, configuration and security systems.
Services:	Infringement analysis, validity analysis, source code review, report drafting.

Uniloc, Inc. v. **Big Fish Games, Inc.**

Case:	Case # 2:16-cv-00741-JRG
Description:	Litigation related to patents on hardware cryptographic chips.
Services:	Non-infringement analysis, report drafting, source code review.

SPEX Technologies, Inc. v. **Toshiba America Electronic Components, Inc., et al.**

Case:	Case # 8:16-cv-01800-JVS
Description:	Litigation related to patents on hardware cryptographic chips.
Services:	Non-infringement analysis, report drafting.

Netskope Ex. 1002
Netskope v. Fortinet IPR2026-00473
Page 00151

Symantec Corporation v. Zscaler, Inc.

Case:	Case # 3:17-cv-04414-JST,
Description:	Litigation related to patents on security gateways, URL filtering and categorization
Services:	Infringement analysis, assignor estoppel, document review.

Koninklijke Philips v. **Microsoft Inc.**

Case:	Case # 4:18-cv-01885-HSG,
Description:	Litigation related to patents on secure cryptographic protocols
Services:	Non-infringement analysis, validity analysis, claim construction analysis, document review, source code review.

Netfuel, Inc. v. Cisco Systems, Inc.

Case:	Case # 5:18-cv-2352-EJD
Description:	Litigation related to patents on agent-based network monitoring, configuration and security systems.
Services:	Infringement analysis, claim construction analysis, source code review, report drafting.

Byrd et al. v. Aaron's, Inc., et al.

Case:	PA Civil Action # 1:11-cv-00101-SJM-SPB
Description:	Class action litigation related to privacy.
Services:	Attend depositions, source code review, report drafting.

Finjan v. **Juniper Networks**

Case:	Case # 3:17-cv-05659-WHA
Description:	Litigation related to patents on malware scanning gateways.
Services:	Invalidity analysis, non-infringement analysis, source code review.

Grace et al. v. **Apple Inc.**

Case:	Case # 5:17-cv-00551-LHK (NC)
Description:	Litigation related to device performance and service outages.
Services:	Mobile forensics and device analysis, report drafting, source code review, document review, technical analysis and argument construction.

Rimini Street, Inc. v. Oracle International Corporation, et al.

Case:	Case # 2:14-cv-01699 LRH-CWH
Description:	Litigation related to false claims on security.
Services:	Large-scale testing of IPS techniques for including custom test infrastructure and implementation of techniques to block exploitation of vulnerabilities, technical analysis of vulnerabilities.

Finjan v. **Symantec Corporation**

Case:	Case # 14-cv-02998-HSG
Description:	Litigation related to patents on malware scanning gateways, endpoint protection and firewalls.
Services:	Build and/or test software for Windows, invalidity argument strategy, non-infringement argument strategy, report preparation, source code reviews.

Strikeforce, Inc. v. **Entrust et al.**

Case:	Case # 1:17-cv-00309
Description:	Litigation related to patents on authentication technologies.
Services:	Invalidity argument development, non-infringement argument development, report drafting.

Netskope Ex. 1002
Netskope v. Fortinet IPR2026-00473
Page 00152

Sony Corporation, Inc. v. Arris

Case:	Inv. # 337-TA-1049
Description:	Litigation related to patents on television streaming devices and/or services.
Services:	Validity argument development, source code review of entire platform codebase including numerous embedded platforms, infringement argument development.

Kudelski SA, Nagra USA, Inc., Nagravision SA, and OpenTV, Inc. v. Comcast Corporation

Case:	Case # 2:16-cv-1362-JRG, Inv. # 337-TA-1049
Description:	Litigation related to patents on television streaming devices and/or services.
Services:	Validity argument development, source code review of entire platform codebase including numerous embedded platforms, infringement argument development.

Amazon.com Inc., Hulu, LLC, and Netflix, Inc. v. Uniloc Luxembourg S.A.

Case:	IPR 2017-00948
Description:	Litigation related to patents on DRM protection for content distribution
Services:	Prior art search, PGR Preparation, IPR preparation.

PhishMe v. **Wombat Technologies, Inc.**

Case:	Case # 16-403-LPS-CJB
Description:	Litigation related to patents on anti-phishing training technologies.
Services:	Prior art search, PGR Preparation, IPR preparation.

Nader Asghari-Kamrani and Kamran Asghari-Kamrani v. **United States Automobile Association**

Case:	Case # 2:15-cv-478
Description:	Litigation related to patents on authentication technologies.
Services:	Prior art search, invalidity argument strategy, non-infringement argument strategy, source code reviews

Vir2us v. **Invincea Inc. and Invincea Labs, LLC**

Case:	Case # Case 2:15-cv-00162-HCM-LRL
Description:	Litigation related to patents on virtualization and automated corruption repair.
Services:	Prior art search, invalidity argument strategy, non-infringement argument strategy, source code reviews

Palo Alto Networks v. Finjan

Case:	IPR 2016-00159, IPR 2016-00151, IPR 2015-01974, IPR 2015-02001, IPR 2015-01979
Description:	Litigation related to patents on malware scanning gateways and firewalls,
Services:	Prior art search, patent interpretation, IPR preparation support, claim chart review

TVIIM v. **McAfee**

Case:	Case # 3:13-cv-04545-VC
Description:	Litigation related to patents on vulnerability scanning
Services:	Build and test software for SPARC/Linux/Windows, patch out license checks/crack software (with permission), obtain hard-to-find legacy software, prior art and non-infringement argument strategy support, source code reviews, prior art search

Netskope Ex. 1002
Netskope v. Fortinet IPR2026-00473
Page 00153

Al Cioffi et al. v. Google

Case:	Case # 2:13-cv-103-JRG-RSP
Description:	Litigation related to patents on browser sandboxing and process isolation.
Services:	Code review/software testing to collect evidence of infringement, Infringement argument preparation support, claim chart review

Rovi Solutions & Veracode v. Appthority

Case:	Case # 12-10487-DPW
Description:	Litigation related to patents on static debugging tools
Services:	Source code review refuting opposing expert testimony

Analysis, Design and Development Clients

Arai	ICU Medical
Baxter	Inexto
Bigfoot Medical	Intuitive Surgical
BT Group	Merlin
Cardiac Sciences	Orpheus
Dyadic	Security First Corporation
Fresenius	Texas Instruments
HLFIP Holdings	Thesys
Hospira	Vaxxin

Pre-Harbor Labs Security Design and Software Development Experience

2013 At *Applied Communication Sciences*

Role:	Graduate Intern
Technologies:	JavaScript, Python, Tcpdump, Wireshark

- Developed extensible real-time traffic visualization tool to chart and analyze high-volume tcpdump streams of lossy metropolitan-area mesh network traffic.

2011 At *University of Michigan ICPSR*

Role:	Penetration Tester
Technologies:	Numerous Security Tools, Amazon EC2, VMware VSphere

- Conducted wide-scale penetration testing on virtualized cloud-based systems meant to be secure environments for researchers to store confidential results
- Created formal threat model document detailing potential security vulnerabilities from all possible attack vectors
- Wrote two reports detailing results from penetration test

2009-2011 At *Independent Security Evaluators*

Role:	Security Intern
Technologies:	C++, C#, Dalvik Bytecode, Gcov, GDB, Javascript, Peach Fuzzer, Python, RegEx, XML, Wireshark

- Created log parsing framework to analyze 20+ log file formats
- Assisted with malware testing, research and analysis
- Reverse engineered DRM schemes in Android and IOS applications
- Researched and prototyped secure cryptographic mail delivery system
- Developed web crawler to collect file sets for use in fuzzing
- Wrote code-coverage analysis tool for constructing minimum file set for fuzz testing
- Wrote fuzzing plugins using Peach Fuzzer framework and reverse engineered binary file

Netskope Ex. 1002
Netskope v. Fortinet IPR2026-00473
Page 00154

specifications
- Wrote Internet Explorer and Chrome extensions for cryptographic proxy system
- Created and debugged network protocols for use in network protocol testing
- Wrote and debugged unit tests in C++ and Python for proprietary disk-encryption system

2008-2010 At *Johns Hopkins University DRCC*
Role: Student Software Developer
Technologies: DOM, Java EE, JSP Perl, SAX, XSLT

- Drafted a report detailing security recommendations for an NSF funded data conservancy project
- Set up and deployed a Fedora digital repository with the Islandora frontend
- Ported IRStats statistics package to the DSpace information repository XMLUI
- Wrote batch importer that is now used to import more than 20 digitized books a week into DSPace repository

2008 At *Brandeis University Information Technology Services Hardware Repair Shop*
Role: Freelance Software Developer
Technologies: Java, Visual C++, VBScript

- Sole programmer on project to interface Request Tracker ticketing system with Brother PT- 9500PC Label Printer

Technical Skills

Languages	BASH, C, C++, C#, HTML, Java, JavaScript, Objective-C, Python, Perl, PHP, Regular Expressions, SQL, XML
Architectures	6502, 8051, 8080, ARM Cortex-M, ARMv7, ARMv8, AVR, m68k, MIPS, MSP430, PIC, SPARC, PowerPC, x86, x86-64, Z80
Operating Systems	Android, ChromeOS, FreeBSD, iOS, OpenBSD, Linux, macOS, Windows
DevOps and Development Tools	Ansible, Ant, BitBucket, Confluence, Docker, gdb, git, GitHub, GitLab, Gradle, Hadoop, jad, jd-gui, Jira, Maven, make, MySQL, PostgreSQL, subversion, Trello, Vagrant, valgrind
Security Tools	Aircrack-ng, apktool, binwalk, bulk-extractor, Burp suite, Charles Proxy, curl, dex2jar, ftk, hashcat, IDA Pro, Metasploit, mitmproxy, Nessus, nmap, OpenSSL, ophcrack, p0f, Scalpel, skipfish, snort, sslstrip, sslyze, Volatility, Web Scarab, wget, Wireshark
Cloud and Virtualization	AWS, Azure, Bhyve, KVM, LXD, QEMU, virt-manager, VMware, Xhyve

Honors, Societies and Awards

- Member Upsilon Pi Epsilon International Computer Science Honor Society
- Member Institute for Electrical and Electronics Engineers (#97507890)
- Member Association for Computing Machinery (#9700346)

At Johns Hopkins University

- Computer Science Department Outstanding Teaching Assistant Award (2014)
- Treasurer, Upsilon Pi Epsilon International Computer Science Honor Society (JHU Chapter)
- Computer Science Department Faculty Liaison Czar
- Student Representative to the Computer Science Undergraduate Curriculum Planning Curriculum

Committee

Certifications

- Six Sigma Black Belt
- Six Sigma Green Belt

Netskope Ex. 1002
Netskope v. Fortinet IPR2026-00473
Page 00156