

UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE PATENT TRIAL AND APPEAL BOARD

ZIMPERIUM, INC.,
Petitioner,
v.
TAASERA LICENSING LLC,
Patent Owner.

Case No. IPR2024-00007

U.S. Patent No. 9,071,518 B2

PETITION FOR *INTER PARTES* REVIEW

TABLE OF CONTENTS

I.	Introduction.....	1
II.	Mandatory Notices.....	3
	A. Real Party-in-Interest	3
	B. Related Matters.....	3
	C. Counsel, Service, and Fee Information	3
III.	Requirements for IPR	4
	A. Grounds for Standing	4
	B. Identification of Challenge and Statement of Precise Relief Requested	5
IV.	Institution Should Be Granted	5
	A. There Are No Grounds for a § 314 Discretionary Denial in this Case ..	5
	B. Denial Under § 325(d) Would Be Inappropriate	7
V.	The '518 Patent.....	7
	A. Overview	7
	B. Prosecution History	9
VI.	Claim Construction.....	10
VII.	Level of Ordinary Skill in the Art	11
VIII.	Prior Art Overview	12
	A. Broch	12
	B. Mahaffey.....	14
	C. Stricklen.....	15
IX.	Ground 1: Claims 1-10, 12, 14-17, and 20-23 Are Obvious Over Broch and Stricklen.....	16

A.	Independent Claims	16
1.	Claim 1	16
2.	Claim 10	28
3.	Claim 17	32
B.	Dependent Claims	36
1.	Claim 2	36
2.	Claim 3	36
3.	Claim 4	37
4.	Claim 5	37
5.	Claims 6, 14, and 21	37
6.	Claims 7, 12, and 20	38
7.	Claims 8, 15, and 22	39
8.	Claims 9, 16, and 23	39
C.	Motivation to Combine	40
X.	Ground 2: Claims 1-10, 12, 14-17, and 20-23 Are Obvious Over Mahaffey	45
A.	Independent Claims	45
1.	Claim 1	45
2.	Claim 10	59
3.	Claim 17	63
B.	Dependent Claims	66
1.	Claim 2	66
2.	Claim 3	66

3.	Claim 4.....	67
4.	Claim 5.....	67
5.	Claims 6, 14, and 21	68
6.	Claims 7, 12, and 20	68
7.	Claims 8, 15, and 22	69
8.	Claims 9, 16, and 23	69
C.	Motivation to Combine	69
XI.	Conclusion	72

LISTING OF EXHIBITS

Exhibit	Description
1001	U.S. Patent No. 9,071,518 B2 to Malik et al.
1002	File History of U.S. Patent No. 9,071,518 B2 to Malik et al.
1003	Declaration of Markus Jakobsson, Ph.D.
1004	U.S. Pub 2012/0131685 A1 to Broch et al.
1005	U.S. Pub. 2011/0047597 A1 to Mahaffey et al.
1006	U.S. Pub. 2008/0070495 A1 to Stricklen et al.
1007	U.S. Pub. 2010/0100591 A1 to Mahaffey et al.
1008	U.S. Patent No. 7,519,609 B2 to McConnell
1009	U.S. Patent No. 6,785,673 B1 to Fernandez et al.
1010	U.S. Pub. 2010/0217774 A1 to Marshall
1011	U.S. Patent No. 6,757,706 B1 to Dong et al.
1012	Joint Claim Construction Chart from <i>In re: Taasera Licensing LLC Patent Litigation</i> , Case No. 2:22-MD-03042-JRG (E.D. Tex.)
1013	Docket from <i>Taasera Licensing LLC v. Check Point Software Tech. Ltd.</i> , Case No. 2:22-CV-00063 (E.D. Tex.)
1014	Docket from <i>Taasera Licensing, LLC v. Musarubra US, LLC</i> , Case No. 2:22-CV-00427 (E.D. Tex.)

I. Introduction

Zimperium, Inc. (“Petitioner” or “Zimperium”) respectfully requests *inter partes* review (IPR) of claims 1-10, 12, 14-17, and 20-23 (the “Challenged Claims”) of U.S. Patent No. 9,071,518 (“the ’518 patent”), assigned to Taasera Licensing LLC (“Patent Owner”).

The ’518 patent generally relates to server-based techniques for managing mobile devices to ensure device security or compliance with an organization’s policies. According to the patent, in prior art systems management was performed using software running on the mobile device itself. The patent attempts to distinguish itself from this by using a server for management tasks. The server gathers data from devices, evaluates that data using administrator-defined rules, and then initiates responsive action (like notification of a rule violation or device wiping) based on the evaluation. The system is also designed to manage devices running different operating systems. Thus, data from the devices is formatted to allow the server to apply a consistent set of rules to the devices regardless of operating system.

There was nothing new about this. This Petition focuses on three exemplary references: (1) U.S. 2012/0131685 (Ex. 1004, “Broch”); (2) U.S. 2011/0047597 (Ex. 1005, “Mahaffey”); and (3) U.S. 2008/0070495 (Ex. 1006, “Stricklen”).

Broch (Ex. 1004)—like the ’518 patent—teaches a method and system for managing mobile devices. Broch’s system is intended to improve device security

and policy compliance by detecting undesirable applications or device changes—like SIM card changes or jailbreaking—raising security concerns. Also like the '518 patent, Broch employs a server—“device management system 102”—to gather data from mobile devices, evaluate that data using administrator-defined policies, and then trigger responsive actions like device notification, blacklisting, or wiping. The mobile devices managed by Broch’s system can run a variety of operating systems. Thus, data from these devices is synchronized with a standard data object stored on the server before analysis. While Broch does not detail exactly how rules are applied uniformly regardless of operating system, Stricklen teaches a similar device management system and does provide that detail, including by using an “abstraction layer” to apply such rules. As discussed in Ground 1, this renders the '518 patent’s claims obvious.

Mahaffey also teaches a system for managing mobile devices. Mahaffey’s system improves device security by detecting applications that are malicious, violate policies (by, for instance, allowing unwanted functionality), or adversely impact device performance. Mahaffey provides yet another example of a server-focused system. Mahaffey’s server—“server 151”—gathers device data, analyzes that data using administrator-defined policies, and triggers actions like notification or isolating a device from other network resources. Mahaffey’s system is meant to manage and apply a consistent set of rules to devices running different operating

systems. A “cross-platform system” is used to ensure that the server is provided with a consistent form of data. Mahaffey incorporates another publication naming the same inventor—U.S. 2010/0100591 (Ex. 1007, “the ’591 application”)—detailing how such a “cross-platform system” operates, including by using an abstraction layer. As explained in Ground 2, this also renders the ’518 patent’s claims obvious.

II. Mandatory Notices

A. Real Party-in-Interest

Petitioner Zimperium, Inc. and Musarubra US, LLC d/b/a Trellox are the real party-in-interest.

B. Related Matters

The ’518 patent is at issue in *Taasera Licensing, LLC v. Musaruba US, LLC*, Case No. 2:22-cv-00427 (E.D. Tex.), *In re: Tassera Licensing LLC, Patent Litigation*, Case No. 2:22-md-03042 (E.D. Tex.), and *Taasera Licensing LLC v. Check Point Software Technologies Ltd.*, Case No. 2:22-cv-00073 (E.D. Tex.).

C. Counsel, Service, and Fee Information

Petitioner designates the following counsel:

Lead Counsel	Back-Up Counsel
K. Patrick Herman Registration No. 75,018 (P52PTABDocket@orrick.com)	Jason K. Yu (<i>Pro hac</i> motion to be filed) (J2YPTABDocket@orrick.com)
Postal & Hand-Delivery Address: Orrick, Herrington & Sutcliffe LLP	Postal & Hand-Delivery Address: Orrick, Herrington & Sutcliffe LLP

51 West 52nd Street New York, NY 10019 T: 212-506-3596	1000 Marsh Road Menlo Park, CA 94025 T: 650-614-7363 Claudia Frost (<i>Pro hac</i> motion to be filed) (PTABDocket6CF@orrick.com) Postal & Hand-Delivery Address: Orrick, Herrington & Sutcliffe LLP 609 Main Street, 40th Floor Houston, TX 77002 T: 713-658-6460
--	---

Petitioner consents to service by electronic mail at the addresses listed in the table above. Petitioner's Power of Attorney is attached.

The USPTO is authorized to charge the filing fee and any other fees incurred by Petitioner to the deposit account of Orrick, Herrington, & Sutcliffe LLP: 15-0665.

III. Requirements for IPR

A. Grounds for Standing

Petitioner certifies that the '518 patent is available for IPR, and that Petitioner is not barred or estopped from requesting this IPR. Real party-in-interest Trellix was served with a complaint alleging infringement of the '518 patent on November 2, 2022. As required by 35 U.S.C. § 315(b), the present petition was filed within 1 year of this date of service.

B. Identification of Challenge and Statement of Precise Relief Requested

Petitioner requests IPR of the Challenged Claims on the Grounds below. This petition is supported by the accompanying Declaration of Markus Jakobsson, Ph.D. (Ex. 1003).

Petitioner relies on the following references: (1) Broch (Ex. 1004), (2) Mahaffey (Ex. 1005), and (3) Stricklen (Ex. 1006). These references are all prior art. (*See* Section VIII.)

Petitioner challenges the claims on the following grounds:

Ground 1: Claims 1-10, 12, 14-17, and 20-23 are obvious over Broch and Stricklen.

Ground 2: Claims 1-10, 12, 14-17, and 20-23 are obvious over Mahaffey.

IV. Institution Should Be Granted

Petitioner has established a reasonable likelihood of success on the merits. And, all other requirements for IPR have been met. The Board should institute IPR.

A. There Are No Grounds for a § 314 Discretionary Denial in this Case

There are two pending district court actions involving the '518 patent. The PTAB has explained that it “will not ... discretionarily deny institution in view of a parallel district court litigation where a petition presents compelling evidence of unpatentability.” 6/21/22 Interim Procedure, pp. 2, 4-5. Petitioner submits that this is the case here: the claims of the '518 patent are unambiguously unpatentable in view of the prior art.

Separate from the strength of the petition, Petitioner is not a party to either of the co-pending district court actions and has its own independent reasons for seeking *inter partes* review. Moreover, Petitioner was not made aware of the litigations or the '518 Patent until May 2023 and Petitioner worked diligently to prepare this petition and file within six months of being so-informed. Petitioner's diligence and the fact that it is not a party to any co-pending litigation favors institution.

The petition's grounds are also materially different from those likely to be raised in the co-pending litigation. The invalidity contentions served in those actions do not include the grounds relied on herein. Nor do those contentions identify or rely on the Stricklen or Mahaffey references cited herein. Moreover, this petition challenges many claims that are not asserted in the co-pending litigation (including 2-5, 7, 9, 12, 16, and 20-21). This minimizes the potential of overlap and also strongly favors institution.

Further, the co-pending district court actions remain in relatively early stages. While jury selection has been tentatively set for April 2024 (*see generally* Exs. 1013, 1014), none of the prerequisites for trial have been completed. No summary judgement motions appear on the docket. (*See generally id.*) Moreover, the district court has not yet issued a claim construction order. (*See generally id.*) Finally, the co-pending litigation is part of a set of six related cases and only two of them involve the '518 Patent. This relative lack of substantive progress makes an April 2024 trial

unlikely and also weighs in favor of institution. *See Sand Revolution II v. Continental*, IPR2019-01393, Paper 24 at 8-9, 11 (PTAB June 16, 2020).

B. Denial Under § 325(d) Would Be Inappropriate

The arguments set forth in this petition are different from those addressed during prosecution. Broch was not cited by Patent Owner or considered by the Examiner. (*See generally* Ex. 1002.) While Mahaffey was also not identified, a related publication is listed by the '518 patent. This related publication, however, was not specifically addressed or discussed by the Examiner. (*See generally id.*)

Further, as explained in Section V.B, the Examiner focused on a reference to Thomas. Thomas purportedly taught only client-centric systems: the clients gathered data and enforced rules. The server was used only to promulgate the rules to the clients. In contrast, like the claims here, Broch and Mahaffey both teach server-centric systems where the server itself gathers and evaluates data for purposes of managing mobile devices. Thus, Broch and Mahaffey are substantively different from the prosecution art and overcome any purported deficiencies identified with that art.

V. The '518 Patent

A. Overview

The '518 patent relates to a method/system “for ... providing remote management of mobile devices at a server using rules based actions.” (Ex. 1001, 1:15-18.) But, these mobile devices “may use several different platforms....” (*Id.*,

1:33-38.) This “make[s] it difficult to ensure security when users access” network “computing resources....” (*Id.*, 1:39-42.) Moreover, users may “utilize a single device for both business and pleasure.” (*Id.*, 1:42-44.) These “devices[] include[e] their own personal preferences, data, and applications” that “may be counter to the security or management goals of an organization.” (*Id.*, 1:45-48.)

The patent proposes a “server-based rules-based action framework.” (*Id.*, 2:6-8.) The server “gather[s] status and configuration information” from “mobile devices.” (*Id.*, 2:8-12.) Then, the server “monitor[s] status information and respond[s] automatically to changes, causing appropriate rules to be evaluated to determine if one is triggered.” (*Id.*, 2:12-15.) The rules are intended to “work across multiple platforms” thereby “simplifying the administrative process.” (*Id.*, 2:16-23.) The rules can specify an “action” to be taken, like “blocking access ... to a corporate resource” or “wiping” the device. (*Id.*, 2:36-43.)

Example mobile device “attribute[s]” that can be gathered include the device’s “operating system,” “battery condition,” whether the device is “jailbroken,” the device’s “password quality,” and the device’s “policy compliance status.” (*Id.*, 20:61-23:40.) Example “rules” include “minimum operating system version[s],” “applications compliance,” restrictions on jailbreaking, and “passcode” requirements. (*Id.*, 19:30-60; Ex. 1003, ¶¶ 50-60.)

B. Prosecution History

The '518 patent's original claims were rejected in December 2013 as either anticipated by or obvious over U.S. 2010/0242088 ("Thomas") alone or in combination with secondary references. (*See* Ex. 1002 at 166, 170-183.)

Patent Owner responded in May 2014. Claim 1 was amended to require evaluating mobile device compliance "at the server" using "administrator"-defined rules. (*See id.* at 331.) Patent Owner argued that in Thomas, "enforcement is generally performed locally on the clients." (*Id.* at 337.) Thus, Thomas purportedly did not teach that "status information" is "gathered and stored" at the server or "evaluating, by a server, a compliance state of mobile devices." (*Id.*)

The claims were rejected again over Thomas in September 2014. (*See id.* at 352-372.)

Patent Owner responded with another amendment in February 2015. Here, the claims were amended to require gathering status information "in a substantially real time manner," "formatting the status information" to allow rules to apply to devices with "a plurality of different mobile operating systems," and initiating actions "at the server." (*Id.* at 390.) Patent Owner reiterated that Thomas "does not handle mobile status information at a server in the claimed manner." (*Id.* at 395.) Instead, it uses "client facilities" "to implement and/or participate in the application of ... policies." (*Id.*) Thus, it does not "gather" status information "at a server." (*Id.*

at 396.) It also did not teach gathering device data from a “plurality” of sources. (*Id.*) Moreover, because data is not sent to a server, Thomas did not teach “formatting” that data as claimed to allow the server to apply a uniform set of rules. (*Id.*)

After this, the claims were allowed. (*See id.* at 465-470; Ex. 1003, ¶¶ 61-74.)

VI. Claim Construction

The Board applies the same *Phillips* claim construction standard used by district courts. *Phillips v. AWH Corp.*, 415 F.3d 1303 (Fed. Cir. 2005) (*en banc*). For purposes of this petition, Petitioner submits that the terms of the ’518 patent’s claims do not require further construction and can be afforded their plain and ordinary meaning. (*See* Ex. 1003, ¶¶ 76-79.)

During one of the co-pending district court actions involving the ’518 patent, certain terms were identified for possible construction: (1) “substantially real time” (claim 1) and “substantially real-time data” (claims 10, 17), and (2) “initiating ... at least one action” (claims 1, 10) and “initiate an action” (claim 17). (Ex. 1012 at 168-175, 184-191.)

Patent Owner argued that “substantially real time” means “without intentional delay, given the processing limitations of the system.” (*Id.* at 168.) Defendants argued that this term is indefinite. For purposes of this petition, Petitioner accepts and has applied Patent Owner’s construction. While the ’518 does not define “real

time,” it does discuss data gathering “in response to device attribute changes.” (Ex. 1001, 12:1-2.) A POSITA would have recognized that the plain and ordinary meaning of “substantially real time” embraces at least this: data gathering that occurs immediately when a device attribute changes. (See Ex. 1003, ¶¶ 80-85.)

Next, Patent Owner argued that “initiating ... at least one action” should be afforded its plain and ordinary meaning. (Ex. 1012 at 184.) Defendants argued that the term “initiating” means “automatically causing to begin in real time.” (*Id.*) For purposes of this petition, Petitioner accepts Patent Owner’s proposal. Regardless, even if Defendants’ construction is applied, the prior art would still render the claims obvious: Broch and Mahaffey teach servers that automatically triggers responsive action after the completion of an evaluation/analysis. (See Ex. 1003, ¶¶ 86-90.)

VII. Level of Ordinary Skill in the Art

A POSITA in the field of the ’518 patent would have had a degree in computer science, computer engineering, electrical engineering, or a similar discipline, along with 2 years of professional experience working with systems for managing mobile device security, or an equivalent level of skill, knowledge, and experience. (Ex. 1003, ¶¶ 43-45.) This POSITA would have been aware of and generally knowledgeable about the different types of mobile devices available to users and technology used to ensure device security. (See *id.*, ¶ 45.)

VIII. Prior Art Overview

The application underlying the '518 patent—U.S. App. 13/540,142—was filed July 2, 2012. The patent also makes a facial claim of priority back to a July 1, 2011 provisional. (Ex. 1001, Cover.) For purposes of this petition only, it is assumed that the '518 patent is entitled to the benefit of this July 2011 date.

A. **Broch**

Broch was filed November 19, 2010. (Ex. 1004, Cover.) It is prior art under at least § 102(e).

Broch relates to a “mobile device management system” allowing “network administrators” to manage mobile devices “in an enterprise network.” (*Id.*, [0016].) Specifically, a “[d]evice management system 102 selectively logs (e.g., tracks and/or stores) data from” multiple “mobile devices 104.” (*Id.*, [0056].)

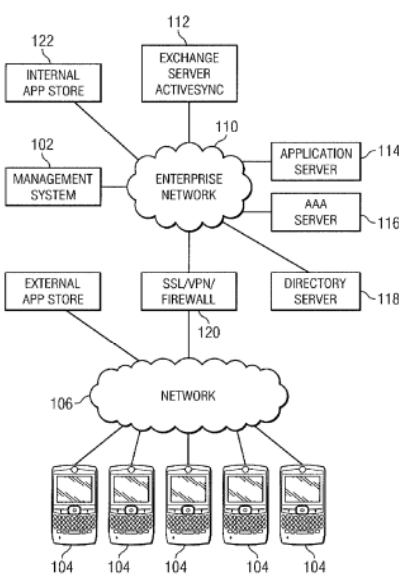


FIG. 1

(*Id.*, Fig. 1.) The devices can run different operating systems, like “Android” or

“Windows Mobile.” (*Id.*, [0023], [0030].) Information gathered from the devices includes “application usage data,” “applications installed,” “SIM ... card changes,” “configuration information,” “large file uploads,” “changes to the operating system files or behavior,” or “other activities ... relevant to a security profile.” (*Id.*, [0038], [0078].) Data can be sent to the management system 102 in “real-time” and is synchronized with a stored “mobile device profile object.” (*Id.*, [0038], [0043], [0047], [0079].)

Broch allows “[a] network administrator” to “create a number of policies.” (*Id.*, [0081].) The server uses the policies to evaluate and manage mobile devices. This can entail determining whether a mobile device improperly includes a blacklisted application or potential virus, includes required applications or upgrades, or is configured in a way that makes it unsecure. (*See id.*, [0078]-[0079], [0081]-[0082].) Based on this evaluation, the server triggers a response, like sending a device notification, device blacklisting, or wiping a managed device. (*See id.*, [0081]-[0082].) Figure 11 provides an example where the system receives status information, applies a policy, and then triggers a response:

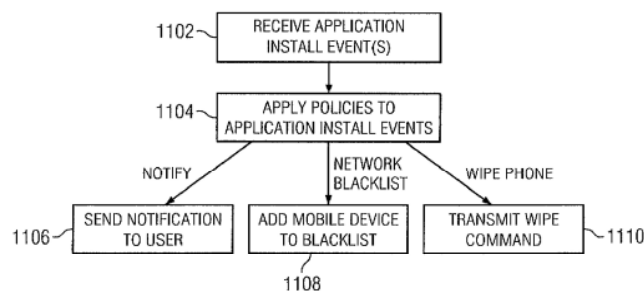


FIG. 11

(*Id.*, Fig. 11, [0082]; Ex. 1003, ¶¶ 93-104.)

B. Mahaffey

Mahaffey published February 24, 2011 and was filed August 25, 2010. (*See* Ex. 1005, Cover.) It is prior art under at least §§ 102(a) and (e).

Mahaffey teaches a system for managing “mobile communication devices.” (Ex. 1005, [0002].) A “server” monitors, “provide[s] protection from[,] and remov[es] ... undesired applications or other data objects that may affect” the “devices....” (*Id.*, [0021].) Multiple “mobile communication devices 101” send data to “servers 151” “over a ... network 121.” (*Id.*, [0025], [0055].)

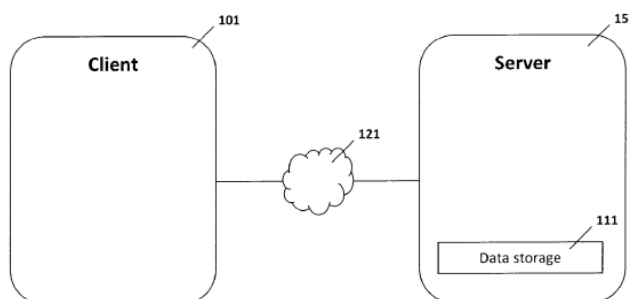


Figure 1

(*Id.*, Fig. 1.) The devices can run various operating systems, like “Apple iOS™” or “Android™ OS....” (*Id.*, [0022].) Transmitted data includes “device data” (operating system, firmware, and hardware information) and “application data” (like the presence of malware, software with coding flaws, software behavioral data, and metadata like file location and characteristics). (*Id.*, [0030]-[0033].)

Mahaffey's system employs a "consistent form of information between" mobile device "platforms" to allow the "server 151" to "include[e] some or all of the same fields in assessments" for all managed devices. (*Id.*, [0167].) To this end, a "cross-platform system" can also facilitate data communication from the different "device platforms." (*Id.*, [0028].) Each mobile device can send data to the server "whenever it encounters an undesirable application...." (*Id.*, [0135].)

Mahaffey allows an "administrator to set policy" for managed devices. (*Id.*, [0182].) For example, "[a] blacklist" of "applications ... explicitly denied from running" on the devices can be specified. (*Id.*) The "policies can be set for a specific application, even if the application is available on multiple platforms" (like the Apple or Android operating systems). (*Id.*, [0199].) Then, upon receipt of device data, Mahaffey's "server 151 ... analyze[s]" it "and provide[s] a security related assessment, response and/or other action" using the policies. (*Id.*, [0027]; *see also id.*, Fig. 3.) Responsive actions include "notif[ying] the ... device" or other "remediation...." (*Id.*, [0040], [0137], [0195]; Ex. 1003, ¶¶ 105-119.)

C. Stricklen

Stricklen published March 20, 2008. (*See* Ex. 1006, Cover.) It is prior art under at least § 102(b).

Stricklen relates to an "an enterprise level device management tool for various mobile device platforms (e.g., Microsoft™, Symbian™, and RIM™)." (*Id.*, [0006].)

It allows a single server to “uniformly manage a plurality of mobile devices of different types” without “know[ing] the manner and format or other platform-specific information” of the devices. (*Id.*, [0062].) “[A]n administrative user” is only required to “define a management task once for multiple mobile devices 119 of different types....” (*Id.*, [0071]; Ex. 1003, ¶¶ 120-124.)

IX. Ground 1: Claims 1-10, 12, 14-17, and 20-23 Are Obvious Over Broch and Stricklen

Broch (Ex. 1004) and Stricklen (Ex. 1006), when considered in view of the knowledge of a POSITA, render claims 1-10, 12, 14-17, and 20-23 obvious.

Broch’s server (management system 102) gathers formatted mobile device data, evaluates that data, and applies administrator-defined policies for purposes of managing the devices. The devices run different operating systems. Stricklen elaborates and expands on Broch, explaining that a system like Broch’s is desirably designed to allow a uniform set of rules to be applied to all the devices regardless of operating system.

A. Independent Claims

1. Claim 1

[1-p] “1. A method for providing device management services comprising steps of:”

To the extent the preamble is deemed limiting, Broch describes “methods, apparatuses and systems directed to managing ... mobile devices in an enterprise environment.” (Ex. 1004, [0015]). Broch further teaches a “mobile device

management system” and method that allows “network administrators to control” various mobile devices “in an enterprise network.” (*Id.*, [0001], [0016]; *see also* Ex. 1003, ¶¶ 128-129.)

[1-1(a)] “gathering, at a server, from a network, status information related to an operating state of each of a plurality of mobile devices,”

Broch’s system gathers “*status information related to an operating state of ... mobile devices.*” This includes “application usage data” from the device or “other data characterizing usage of, and/or the actions performed on, the mobile device.” (Ex. 1004, [0038].) “[D]ata that is relevant to a security profile” of managed devices is also gathered. (*Id.*, [0078].) This includes “applications installed,” “configuration information,” “SIM ... card changes,” “large file uploads to a memory card or a remote host,” “changes to the operating system files or behavior, [or] excessive or uncharacteristic out-of-network roaming.....” (*Id.*)

This is all “*status information related to*” a device’s “*operating state.*” Application and device usage data and configuration information like SIM card and operating system changes all relate to both the current status and operation of a mobile device. (See Ex. 1003, ¶¶ 130-136; Ex. 1004, [0038], [0078].)

The data Broch collects from a “*plurality of mobile devices*” is “*gather[ed]*” at a “*server*” over a “*network.*” “[A] plurality of mobile devices 104 ... communicate with device management system 102 via ... network links 106.” (Ex. 1004, [0018].) “Management system 102” can be a “computer server[.]....” (*Id.*,

[0021].) In operation, “[d]evice management system 102 selectively logs (e.g., tracks and/or stores) data from the mobile devices....” (*Id.*, [0056].) This allows the management system 102 to “maintain an accurate image of ... the mobile device 104, including the applications installed and the files stored on the mobile device.” (*Id.*, [0043]; *see also id.*, [0016] (explaining that the server tracks “the security state” of mobile devices).)

For instance, in Fig. 1 multiple mobile devices 104 (blue) upload data (orange) over network 106 to management system 102 (red):

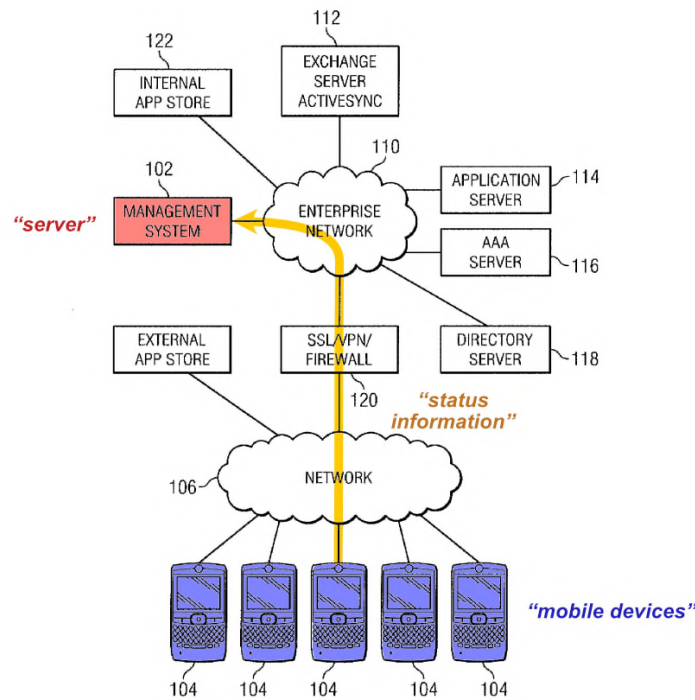


FIG. 1

(*Id.*, Fig. 1 (annotated); *see also* Ex. 1003, ¶¶ 137-141.)

[1-1(b)] “each mobile device utilizing one of a plurality of different mobile operating systems,”

Broch’s system gathers data from devices running *“a plurality of different mobile operating systems.”* Each “mobile device 104” can be “a smartphone” like the “iPhone,” “G1 based on the Android operating system, or Samsung BlackJack based on the Windows Mobile operating system...” (Ex. 1004, [0030]; *see also id.*, [0032] (explaining that the managed devices can employ various “suitable operating systems”); [0046] (explaining that the server collects “operating system type” information from devices).) Stricklen similarly confirms that the system accommodates mobile devices running different operating systems. Per Stricklen, “mobile devices utilized by employees or associates of an organization”—the types of devices Broch’s system is designed to monitor—“vary in their type, supporting platform, or vary in other ways.” (Ex. 1006, [0003].) Among other things, the devices routinely employ different “operating system.” (*Id.*, [0069]; *see also* Ex. 1003, ¶¶ 142-150.)

[1-1(c)] “wherein the status information for each mobile device is gathered from a plurality of sources including each mobile device in a substantially real time manner;”

Broch gathers status information *“from a plurality of sources including each mobile device.”* Broch’s system gathers a variety of status information from mobile devices. (*See supra* limitation [1-1(a)].) It also obtains information from other network sources.

For instance, the “mobile device management system 102 may include a component that interacts with an ActiveSync/Exchange server 112” to obtain “visibility into all devices attempting to connect to corporate emails.” (Ex. 1004, [0084].) This server 112 is connected to the same network as device management system 102. (*See id.*, Fig. 1.) And, the server “may be configured to transmit information on mobile devices 104 accessing the server.” (*Id.*; *see also* Ex. 1003, ¶¶ 151-156.)

The management system 102 is also connected to “authentication (AAA) servers 116.” (Ex. 1004, [0020]; Fig. 1.) Per Broch, the mobile device “profile object” stored and analyzed by system 102 can include “authentication information.” (*Id.*, [0047].) A POSITA would have understood that this information would come—at least in part—from the authentication servers. (*See* Ex. 1003, ¶¶ 157-160.)

Broch’s system also allows for “***substantially real time***” data gathering. A “control client [application] 308” running on each mobile device is responsible for collecting device data. (Ex. 1004, [0038].) This “application 308” can “operate to provide ... data” to the server “in real-time....” (*Id.*, [0043].) For instance, data transmitted “to device management system on a ... as-needed basis” including “when new applications are installed” on a device. (*Id.*, [0079].) In other words, data is sent to the management system as it is generated and without delay. (*See id.*; *see also* Ex. 1003, ¶¶ 161-165.)

[1-2] “formatting the status information for each of the mobile devices, such that rules can be substantially uniformly applied to the status information across the plurality of different mobile operating systems;”

Broch’s system “*format[s] the status information*” sent from mobile devices to its management system. Before transferring data, Broch’s mobile device “control client 308 creates a hash for the particular file and causes mobile device 104 to transmit the hash to device management system 102.” (Ex. 1004, [0057].) Then, when device data arrives at the management system server, it is “synchronize[d] ... with a corresponding mobile device profile object stored” by that server. (*Id.*, [0038].) By synchronizing the data its system transmits with a standardized server object, Broch “*format[s]*” the data as this limit requires. (See Ex. 1003, ¶¶ 166-170.)

Next, in Broch, “*rules*” are “*applied to the status information*” collected from mobile devices. “Device management system 102” can “apply a wide variety of policies.” (Ex. 1004, [0082].) These policies “manage[]” the “mobile devices 104” that transmit data to Broch’s system. (*Id.*, [0058].) The policies may determine if a “newly installed application has been placed on a blacklist” or is a “potential virus.” (*Id.*, [0082].) They can also determine “whether required applications (and upgrades thereto) have been installed....” (*Id.*, [0078].) Similarly, relayed device “configuration” information can be used to “determine whether any changes have occurred”—like “a SIM card change” or the device “being jailbroken or rooted”—

that “indicate a security issue.” (*Id.*, [0079]; *see also id.*, [0081]-[0082]; Ex. 1003, ¶¶ 171-174.)

A POSITA would also have understood that Broch’s rules are “***substantially uniformly applied ... across the plurality of different mobile operating systems.***” Again, Broch’s system monitors and applies policies to a “plurality of mobile devices 104.” (Ex. 1004, [0018].) And, these devices can run a variety of different operating systems. (*See id.*, [0023], [0032].) Moreover, Figure 13 of Broch “illustrates an example application management interface ... that allows a network administrator to configure one or more policies....” (*Id.*, [0086].) As shown, the policy can be designated as applicable to “All-Smartphones” (as opposed to just “Android” or “iPhone” devices):

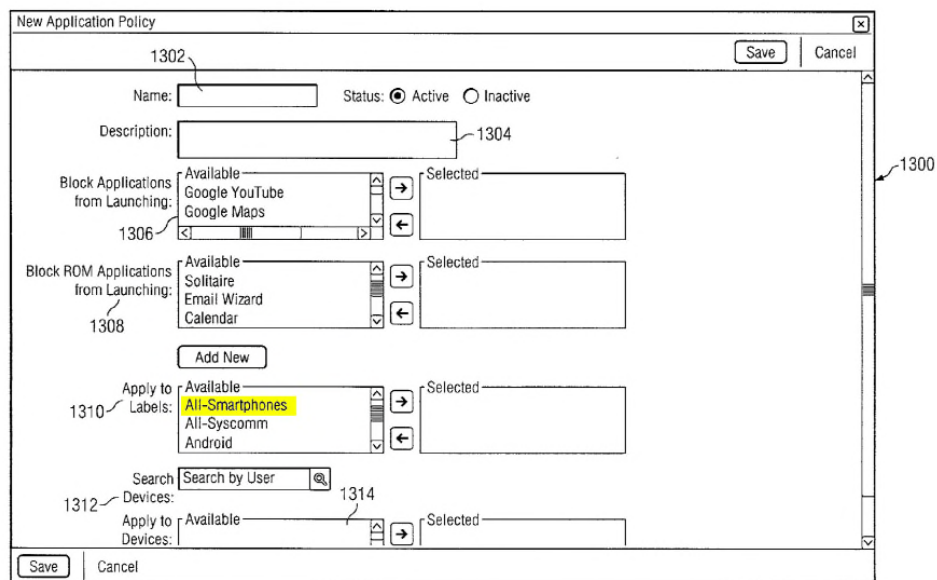


FIG. 13

(*Id.*, Fig. 13 (annotated); [0092]; *see also* Ex. 1003, ¶¶ 175-179.)

Stricklen similarly teaches applying server-applied policies like those taught by Broch uniformly regardless of device operating system. Stricklen “provides enterprises ... with an enterprise level device management tool for various mobile device platforms...” (Ex. 1006, [0006].) To allow these different mobile device platforms to communicate with a single server, “agents or modules native to the platforms themselves ... provide interaction with individual mobile devices.” (*Id.*, [0007].) An “abstraction layer” then allows “management tasks or other functions relating to mobile devices of different types” to be “generally defined” for the “different platforms.” (*Id.*, [0007].) Due to the use of the native agent and the abstraction layer, the managing server “need not know the manner and format or other platform-specific information to uniformly manage” the different devices. (*Id.*, [0062].) In other words, by using intermediate layers that translate communications, Stricklen teaches that a central server can “manage mobile devices of varying types and platforms, such that an administrative user need only define a management task once for multiple mobile devices 119 of different types....” (*Id.*, [0071]; *see also* Ex. 1003, ¶¶ 180-184.)

This is exactly how a POSITA would have understood Broch’s system operates. The data Broch collects would be processed through an abstraction layer and then stored as a device profile object at the management server to allow Broch’s server to apply a single, unified set of policies to all managed devices. (*See id.*, ¶¶

185-186.) Because device data is passed through this abstraction layer, and stored as a standardized profile object (instead of in the device’s native format), the policies can be applied to all the devices regardless of operating system. (*See id.*, ¶¶ 187-188.)

[1-3] “storing the status information in an electronic database accessible to the server;”

Broch’s “*server*” (its device management system 102) includes an “*electronic database*” that “*stor[es] the status information*” gathered from devices. Logged device data (the status information, *see* [1-1(a)]) is stored in a “database within or connected with device management system 102....” (Ex. 1004, [0057]; *see also* Ex. 1003, ¶¶ 189-192.)

[1-4] “evaluating, at the server, a first compliance state of each of the mobile devices from the status information using a plurality of administrator-defined rules, wherein each of the rules applies to mobile devices regardless of mobile operating system; and”

Broch’s system employs “*a plurality of administrator-defined rules.*” Per Broch, “[a] network administrator may ... create a number of policies.” (Ex. 1004, [0081].) As explained when discussing limitation [1-2], these policies can evaluate whether a mobile device improperly includes a blacklisted application or potential virus, includes required applications or upgrades, or is configured in a way (*i.e.*, has an unauthorized SIM card or is jailbroken/rooted) that makes it unsecure. (*See id.*, [0078], [0079], [0081]-[0082].) Moreover, in view of Stricklen, a POSITA would

have understood that Broch's system uniformly applies these rules to mobile devices ***“regardless of mobile operating system.”*** (See Ex. 1006, [0007], [0062], [0071]; see also *supra* limitation [1-2]; Ex. 1003, ¶¶ 193-197.)

Next, Broch's system uses administrator-defined rules to ***“evaluat[e], at the server, a first compliance state of each of the mobile devices from the status information.”*** Management system 102 “analyze[s]” “security or device state information” “to detect one or more device state changes.” (Ex. 1004, [0079].) As part of this analysis, the system uses administrator-defined policies to “identify which applications have been installed on the managed mobile devices 104, including whether required applications (and upgrades thereto) have been installed.” (*Id.*, [0078].) It can also “determine whether any changes have occurred ... that, for example, might indicate a security issue.” (*Id.*, [0079].) Similarly, “management system 102 may also determine from the device configuration whether certain functions are enabled or disabled,” or whether “the policy configuration of the mobile device 104 is up-to-date, such as minimum acceptable password attributes and the like. ...” (*Id.*)

Figure 11 provides an example. Here, the management system / server first “receive[s]” data from monitored mobile devices indicating “application install event(s).” (*Id.*, Fig. 11.) Then, the “system 102 ... applies one or more policies to the application install event” (yellow highlighting). (*Id.*, [0082].)

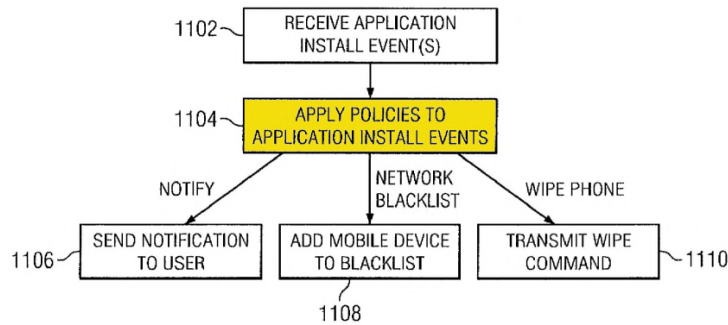


FIG. 11

(*Id.*, Fig. 11 (annotated); *see also* Ex. 1003, ¶¶ 198-208.)

[1-5] “initiating, at the server, at least one action defined by the administrator-defined rules in response to the step of evaluating,”

Broch’s “*server*” (device management system 102) “*initiate[s] ... at least one action defined by the administrator defined rules*” in response to its evaluation of mobile device data. “A network administrator can configure policies to respond to application install events in a variety of manners.” (Ex. 1004, [0082]; *see also id.*, [0081] (discussing “actions ... triggered” by analysis).) For instance, if it is determined that an application has been installed on a device that is “unauthorized, but do[es] not present security issues,” “system 102 may ... transmit a notification or warning to the user of the managed mobile device 104....” (*Id.*) Alternatively, if more severe security issues are detected like the presence of “a known virus or other malicious application,” “[a]nother policy can ... cause the mobile device management system 102 to transmit a wipe command” to the device. (*Id.*) Various other actions can also be initiated at the server, such as “lock[ing] down one or more of the input/output devices of the mobile device,” “push[ing] new policy

configurations to the mobile device,” or “block[ing] access to” email or other servers. (*Id.*, [0039], [0082]-[0083].)

Once again, Figure 11 provides an example. Here, after the server “appl[ies] policies to application install events,” it can trigger actions including “send[ing] notification to user,” “add[ing] mobile device to blacklist,” or “transmit[ting] wipe command” (all highlighted yellow):

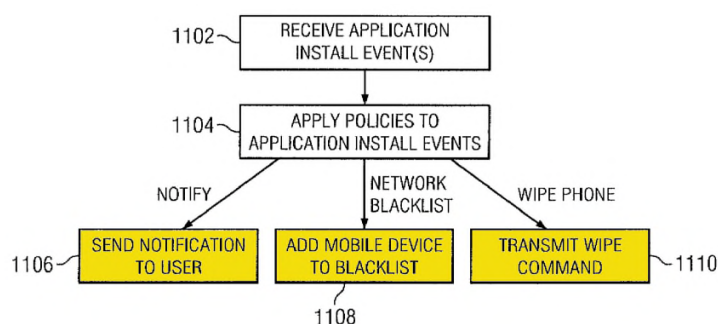


FIG. 11

(Fig. 11 (annotated).) As shown, the responses are automatically triggered by and occur immediately after the policy evaluation performed by the server. (*See id.*; *see also id.*, [0081] (server actions are “triggered” by evaluations); [0082] (policy evaluation causes the server to “respond”); Ex. 1003, ¶¶ 209-221.)

[1-6] “wherein the step of evaluating is performed by the server automatically, in response to changes in the status information in the electronic database.”

Again, Broch teaches that mobile devices transmit a variety of different types of data to its server for storage and evaluation, including data indicating application installs or security change events. (*See supra* limitation [1-1(a)].) Further, the evaluation process performed by “system 102 ... may be initiated in response to

individual events....” (Ex. 1004, [0082].) Figure 11 provides an overview. As shown, receipt of data from a mobile device at step 1102 results in the performance of evaluation step 1104:

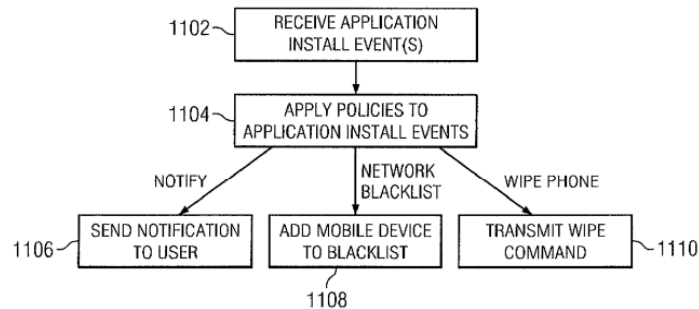


FIG. 11

(*Id.*, Fig. 11.) Because Broch both (1) teaches that evaluation can be “initiated” by the receipt and storage of data at the server, and (2) describes and depicts a process that begins this evaluation immediately upon data receipt/storage without any intervening steps or events, a POSITA would understand that Broch’s evaluation process is *“performed ... automatically” “in response to changes in the status information in the electronic database.”* (See Ex. 1003, ¶¶ 222-228.)

2. Claim 10

Claim 10 is another independent claim.

[10-p] “10. A method for providing device management services comprising steps of:”

Broch teaches a server-based system for managing mobile devices. (See *supra* limitation [1-p]; Ex. 1003, ¶¶ 230-231.)

[10-1(a)] “receiving, at a server, a plurality of attributes related to an operating state of each of a plurality of mobile devices,”

Broch’s system gathers—at a server—various “*attributes related to an operating state*” of mobile devices. This includes application and device usage data, the identity and update state of installed applications, configuration information, SIM card change data, data indicating unusual operating system or network behavior, and whether the device is jailbroken/rooted. (*E.g.*, Ex. 1004, [0038], [0078]-[0079], [0081]-0082]; *see also supra* limitation [1-1(a)]; *see also* Ex. 1003, ¶¶ 232-234.)

[10-1(b)] “each mobile device utilizing one of a plurality of different mobile operating systems,”

Broch’s system gathers and analyzes data from mobile devices running different operating systems. (*See supra* limitation [1-1(b)]; Ex. 1003, ¶¶ 235-236.)

[10-1(c)] “wherein the attributes for each mobile device are gathered from a plurality of sources including each mobile device;”

Broch’s system gathers data from and manages a plurality of mobile devices. It also gathers device data from other sources, including ActiveSync/Exchange server 112 and authentication servers 116. (*See supra* limitation [1-1(c)]; Ex. 1003, ¶¶ 237-239.)

[10-2] “making a determination that an attribute of a mobile device has changed;”

Broch “*determin[es]*” that device “*attribute[s]*” have “*changed.*” After the “control client 308” running on a mobile device “transmit[s] state information and

event-related data to device management system[,]" the management system "may analyze this data to detect one or more device state changes." (Ex. 1004, [0079].) For instance, the system detects "that a new application has been installed." (*Id.*, [0082].) It also determines that "SIM ... card changes" have occurred. (*Id.*, [0078].) And, "changes to the operating system files or behavior" are determined. (*Id.*; see also *id.*, [0081], [0083]; Ex. 1003, ¶¶ 240-246.)

[10-3] "evaluating the attribute, by a processor at the server in response to the determination, to determine if the attribute change triggers at least one of a plurality of administrator-defined rules, which indicates that the attribute is out of compliance, wherein each of the rules applies to mobile devices regardless of mobile operating system; and"

Broch and Stricklen teach a system that employs a "*plurality of administrator-defined rules*" applicable to monitored devices "*regardless of mobile operating system.*" (See *supra* limitation [1-4].) This includes policies prohibiting applications or requiring upgrades, preventing jailbreaking/rooting, or requiring security compliance. (See *id.*; Ex. 1003, ¶¶ 247-249.)

Broch "*determine[s] if the attribute change triggers at least one of a plurality of administrator-defined rules*" because "*the attribute is out of compliance.*" Broch's management system monitors for and analyzes changes in mobile device data. (*E.g.*, Ex. 1004, [0078]-[0079], [0081]-[0083].) And, the system allows for the specification of administrator-defined policies. (*Id.*, [0081]; see also *supra* limitation [1-4].) Broch's system uses these policies to "manage"

mobile devices and determine device compliance. (Ex. 1004, [0058].) This can entail determining if a device attribute change, like an “application install,” runs afoul of a policy. (*Id.*, [0081]-[0082]; Fig. 11.) Other changes that violate security policies—and thus indicate that the device is out of compliance—can also be determined: Broch can determine if “certain functions are enabled or disabled,” that a “SIM card change” has occurred, that the device has been “jailbroken or rooted,” or that a newly installed application is a “potential virus.” (*Id.*, [0079], [0081]-[0082]; *see also* Ex. 1003, ¶¶ 250-257.) This determination occurs automatically. (*See supra* limitation [1-6].)

[10-4] “automatically initiating, at the server, an action defined by the at least one of a plurality of administrator defined rules if the attribute is out of compliance,”

Broch’s system automatically initiates a response (like a notification, device blacklisting, or wiping) at the server in response to analysis. (*See supra* limitations [1-5]-[1-6]; Ex. 1003, ¶¶ 258-259.)

[10-5(a)] “wherein the attribute is stored in a data store accessible to the server, that includes substantially real-time data, gathered from the plurality of sources on a network and pertaining to a plurality of attributes of a plurality of mobile devices,”

Broch’s system includes the claimed “*data store accessible to the server*”: its “device management system 102” includes a “database” storing device data. (Ex. 1004, [0057]; *see also supra* Section limitation [1-3].) “*Substantially real-time data*” is stored: devices send data to Broch’s management system 102 “in real-time.”

(Ex. 1004, [0043]; *see also id.*, [0079].) And, Broch teaches both “***gather[ing]***” data “***from the plurality of sources***” and data “***pertaining to a plurality of attributes of a plurality of mobile devices***”: Broch’s system gathers device data both from the devices themselves and other external sources like ActiveSync/Exchange and authentication servers. (*See supra* limitation [1-1(c)]; Ex. 1003, ¶¶ 260-266.)

[10-5(b)] “wherein the substantially real-time data is formatted, such that the rules can be substantially uniformly applied to the plurality of attributes across the plurality of different mobile operating systems.”

Broch “***format[s]***” collected device data by hashing and then synchronizing it with a “mobile device profile object” stored on its management system. (Ex. 1004, [0038], [0057].) Broch and Stricklen together teach a system where rules can be “***substantially uniformly applied***” to data collected from devices running “***different mobile operating systems***.” Broch’s systems can manage mobile devices running different operating systems. (*See id.*, [0023], [0038].) It also allows rules to be defined that apply to all these devices regardless of operating system. (*See, e.g.*, Fig. 13.) Stricklen confirms that a system like that taught by Broch allows rules that uniformly “manage mobile devices of varying types and platforms” by processing information through an abstraction layer. (Ex. 1006, [0071]; *see also supra* limitation [1-2]; Ex. 1003, ¶¶ 267-271.)

3. Claim 17

Claim 17 is yet another independent claim.

[17-p] “17. A system for providing device management services comprising:”

This is taught by Broch for the reasons discussed for preamble elements [1-p] and [10-p]. (See Ex. 1003, ¶ 273.)

[17-1] “a data store configured to be accessible by a server;”

Broch’s management system 102 includes a database. (See *supra* limitations [1-3], [10-5(a)]; Ex. 1003, ¶ 274.)

[17-2] “a CPU of the server coupled to a memory configured to execute computer readable instructions;”

[17-3] “computer readable instructions for execution on the server comprising instructions configured to:”

Broch explains that its “management system 102” may be “one or more computer servers or systems.” (Ex. 1004, [0021].) “FIG. 2 illustrates an example” of one server/computer system. (*Id.*) The system includes both “[p]rocessor(s) 201” and “memory 203.” (*Id.*, [0026].) The memory can include “[s]oftware” for “implementing” Broch’s various functions. (*Id.*, [0023], [0026].)

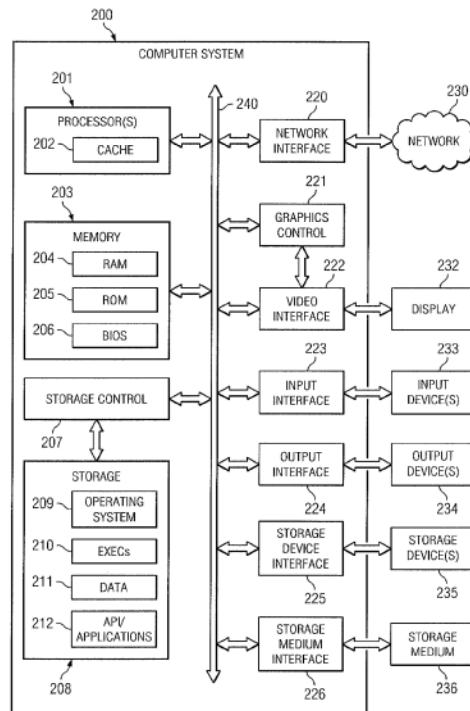


FIG. 2

(*Id.*, Fig. 2; see also Ex. 1003, ¶¶ 275-279.)

[17-3(a)] “automatically make a determination that an attribute of a mobile device stored in the data store has changed;”

Broch’s system is automated and monitors for mobile device attribute changes. (*See supra* limitations [1-1(c)], [1-6], [10-2]; Ex. 1003, ¶ 280.)

[17-3(b)] “evaluate the attribute, in response to the determination, to determine if the attribute change triggers at least one of a plurality of administrator-defined rules, which indicates that the attribute is out of compliance, wherein each of the rules applies to mobile devices regardless of mobile operating system; and”

Broch’s management system 102 evaluates device data using administrator-defined policies to identify device policy compliance and violations (like improper

applications, viruses, SIM card changes, jailbreaking, etc.). (See *supra* limitations [1-4], [10-3]; Ex. 1003, ¶ 281.)

[17-3(c)] “automatically initiate an action defined by the at least one of a plurality of administrator defined rules if the attribute is out of compliance,”

Broch’s system automatically initiates a response (like a notification, device blacklisting, or wiping) at the server in response to the performed analysis. (See *supra* limitations [1-5]-[1-6]; Ex. 1003, ¶ 282.)

[17-3(d)] “wherein the data store is configured to gather substantially real-time data pertaining to a plurality of attributes related to an operating state of each of the plurality of mobile devices, the attributes for each mobile device being gathered from a plurality of sources including each mobile device, wherein each mobile device utilizes one of a plurality of different mobile operating systems, and the substantially real-time data is formatted, such that the rules can be substantially uniformly applied to the plurality of attributes across the plurality of different mobile operating systems.”

Broch’s system gathers data in “*substantially real-time.*” (Ex. 1004, [0038], [0043], [0079] *see also supra* limitation [1-1(c)].) Next, the gathered “*attributes*” come from “*a plurality of sources including each mobile device*”: Broch collects device data, and data from other external sources. (See *supra id.*) Broch’s system is also designed to work with different mobile devices running “*different mobile operating systems.*” (See Ex. 1004, [0023], [0038]; *see also supra* limitation [1-1(b)].) Further, Broch “*format[s]*” data, and Broch and Stricklen both teach systems allowing “*rules*” to “*be substantially uniformly applied*”: Broch hashes collected data and synchronizes it with a standardized object stored on its server. (See *supra*

limitation [1-2].) An administrator can designate policies as applicable to all devices (regardless of operating system). (*See supra* limitation [1-4].) Stricklen confirms that a system like that taught by Broch that manages devices running different operating systems also obviously allows for uniformly and generally applicable rules and policies. (*See supra* limitation [1-2]; Ex. 1003, ¶¶ 283-290.)

B. Dependent Claims

1. Claim 2

Claims 2 depends on claims 1 and further requires that:

“wherein the at least one action is selectable from a plurality of actions, comprising blocking access by the at least one of the mobile devices to a corporate resource”

After evaluating device data, Broch’s server can “trigger[]” various “possible actions.” (Ex. 1004, [0081]; *see also* Fig. 11.) Among these, the server can “[b]lock[] access” to “[o]ne or more network application services...” (*Id.*, [0081].) This can include “[e]nterprise [i.e., corporate] network applications, such as an email application....” (*Id.*, [0016]; *see also* Ex. 1003, ¶¶ 291-295.)

2. Claim 3

Claims 3 depends on claims 1 and further requires that:

“wherein the at least one action is selectable from a plurality of actions, comprising blocking e-mail access for the at least one of the mobile devices.”

Again, Broch’s server can block device access to an “email application.” (Ex. 1004, [0016], [0081]; *see also* Ex. 1003, ¶¶ 296-298.)

3. Claim 4

Claims 4 depends on claims 1 and further requires that:

“wherein the at least one action is selectable from a plurality of actions, comprising wiping information from the at least one of the mobile devices.”

Broch’s server can also “initiate a wipe action” on a mobile device. (Ex. 1004, [0081]; Fig. 11; *see also* Ex. 1003, ¶¶ 299-301.)

4. Claim 5

Claims 5 depends on claims 1 and further requires that:

“wherein the at least one action is selectable from a plurality of actions, comprising warning a user of the at least one of the mobile devices that further action may be taken.”

Again, Broch teaches that its server can initiate a device wipe or block access to network resources. (*See supra* discussion of claims 3, 4.) Moreover, the server can also transmit “warning notifications to the user ... if an unauthorized application is detected on a mobile phone.” (Ex. 1004 [0081]; Fig. 11.) A POSITA would have understood or found it obvious that such “warning notifications” would include a notification of an impending wipe/resource block if the unauthorized application is not deleted. (*See* Ex. 1003, ¶¶ 302-306.)

5. Claims 6, 14, and 21

Claims 6, 14, and 21 respectively depend on claims 1, 10, and 17. They all similarly require:

“gathering [attribute] information from that mobile device and at least one other source on the network”

Broch’s system gathers a variety of status information from mobile devices and **“other source[s] on the network”** like an “ActiveSync/Exchange server 112” and “authentication (AAA) servers 116.” (*See supra* limitation [1-1(c)]; Ex. 1003, ¶¶ 307-309.)

6. Claims 7, 12, and 20

Claims 7, 12, and 20 respectively depend on claims 1, 10, and 17 and further require:

“the step of subsequently evaluating a second compliance state of at least one of the mobile devices to determine if the at least one action should be revoked” (claim 7)

“[the step of]/[wherein the computer readable instructions further comprise instructions configured to] automatically [revoking]/[revoke] the action if the attribute is brought back into compliance” (claims 12, 20)

Broch’s system “monitor[s] the security state” of mobile devices and blocks policy violating devices from accessing “network application services.” (Ex. 1004, [0016], [0081]; Fig. 11.) To do so, the management system “signal[s] the security state of a managed mobile device ... to ... enterprise services.” (*Id.*, [0083]; *see also id.*, Abstract, [0016], [0081].) A device can be both “admit[ted] or den[ied] access.” (*Id.*) Again, Broch evaluates and triggers responsive actions automatically. (*See id.*, [0082]; Fig. 11; *supra* limitation [1-6].)

Because Broch continuously monitors device security and provides enterprise services with both deny and admit access signals, a POSITA would have understood or found obvious that Broch not only denies access when a device initially falls out of compliance, but also re-admits that same device when it is later determined (in a second compliance state) to come back into compliance. (See Ex. 1003, ¶¶ 310-317.)

7. Claims 8, 15, and 22

Claims 8, 15, and 22 respectively depend on claims 1, 10, and 17. They all further requires that:

“the plurality of different mobile operating systems comprise at least operating systems compliant with Android and iOS specifications”

Broch teaches that its system can monitor and manage mobile devices such as “the iPhone” (which runs the iOS operating system) and “the G1 based on the Android operating system....” (Ex. 1004, [0030]; *see also* Ex. 1003, ¶¶ 318-320.)

8. Claims 9, 16, and 23

Claims 9, 16, and 23 respectively depend on claims 1, 10, and 17 and further relatedly require:

“wherein the [status information / attribute] comprises location information about [each / the] mobile device”

Broch explains that each mobile device can include “Global Positioning System modules.” (Ex. 1004, [0036].) Because of this, the “application data” collected by Broch’s system includes “context data” like “timestamps and location

data....” (*Id.*, [0038]; *see also id.*, [0049] (Broch’s system allows “locating” a user “device (if GPS enabled)”); Ex. 1003, ¶¶ 321-325.)

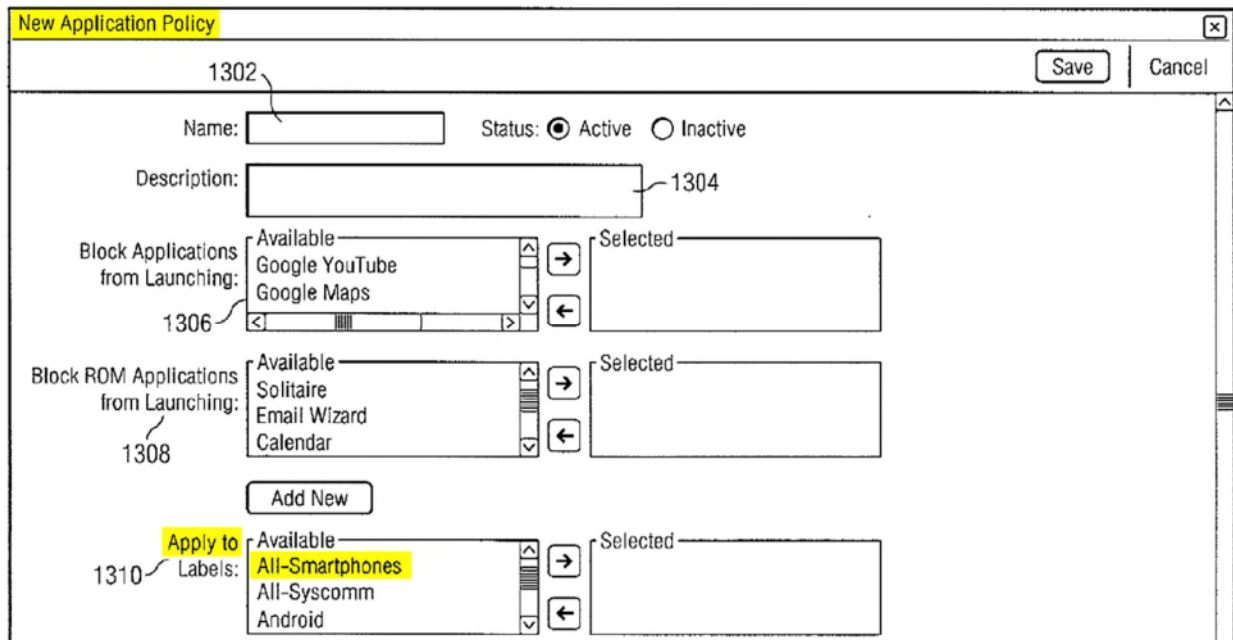
C. Motivation to Combine

A POSITA would have considered it obvious and would have been motivated to combine the teachings of Broch and Stricklen in the manner discussed above. (*See* Ex. 1003, ¶¶ 326-360.)

Broch teaches a system that employs real-time data collection, gathers various different types of device data, and to allows for a range of policies. (*See id.*, ¶ 327.) While Broch contemplates different data collection frequencies—such as collection on a “periodic basis” (*e.g.*, Ex. 1004, [0079], [0082])—it also specifically teaches and allows for “real-time” collection “when” events occur. (*Id.*, [0043], [0079].) Accordingly, a POSITA would have considered the use of real-time data collection to be nothing more than a straightforward design choice taught by Broch. (*See id.*, ¶¶ 330-331.) Moreover, certain types of evaluation performed by Broch (like malware detection) make real-time data collection important and would motivate its use. (*See id.*, ¶ 332.) And a POSITA would have recognized that many of the types of data Broch collects—like application installs and SIM card changes—occur less frequently and, accordingly, real-time monitoring would not impose a significant burden on either the managed devices or the managing server. (*See id.*, ¶¶ 333-334.)

Likewise, Broch teaches that the system collects and evaluates different types of device data. (*See id.*, ¶ 335.) While some examples in Broch more limited in scope—Fig. 11, for instance, monitors and assesses “application install event(s)” (Ex. 1004, Fig. 11)—Broch’s system is intended to collect and assess a range of status, operation, and attribute information from devices. (*See* Ex. 1003, ¶¶ 336-340.) Broch’s system is broadly intended to “allow network administrators” to “manag[e] and control[]” various aspects of “mobile devices in an enterprise environment.” (Ex. 1004, [0015]-[0016].) This includes not only “the deployment of applications,” but also device “security state.” (*Id.*) A POSITA would have understood that collecting more data and allowing greater policy flexibility would further these goals by allowing Broch’s system to detect (and address) a greater range of potential security issues. (*See* Ex. 1003, ¶¶ 340-341.)

Next, a POSITA would have been motivated to look to Stricklen when implementing Broch’s multi-platform functionality. Broch explains that its system intended to accommodate devices running different operating systems. (Ex. 1004, [0030], [0032].) An administrator can, for example, designate a policy as applicable to “All-Smartphones” regardless of operating system:



(*Id.*, Ex. 1004, Fig. 13 (cropped and annotated).) Accordingly, a POSITA would have been motivated to look to other available references teaching multi-platform systems (like Stricklen) when following these teachings and implementing Broch. (See Ex. 1003, ¶¶ 345-346.).

Stricklen, in turn, details how such multi-platform functionality (like that taught by Broch) can be achieved. (See *id.*) Specifically, Stricklen “enables device-independent consolidated asset inventory, configuration management, policy compliance, and security management through a single management application.” (Ex. 1006, [0006].) The system employs both an “agent[] or module” that is “native to the platforms themselves” and an “abstraction layer.” (*Id.*, [0007].) The native agent/module allows for “interaction with individual mobile devices” while the abstraction layer allows “management tasks or other functions relating to mobile

devices of different types” to be “generally defined and translated....” (*Id.*; *see also id.*, [0027].)

A POSITA would have understood that Stricklen operates just in the way Broch requires: devices running multiple operating systems are able to communicate with a single management server. (*See* Ex. 1003, ¶¶ 342-350). This in turn allows rules to be specified that apply to all devices regardless of operating system. (*See id.*) This similarity in purpose and operation would have provided a POSITA with a strong motivation to consider and use Stricklen when implementing Broch. (*See id.*, ¶¶ 351-352.) Put another way, a POSITA would have recognized that Stricklen provides a straightforward and highly obvious way to achieve the functionality already taught by Broch. (*See id.*) Indeed, a POSITA would have considered use of Stricklen with Broch to constitute nothing more than combining known and related prior art elements in their intended way to yield predictable results. (*See id.*, ¶¶ 345, 352.)

A POSITA also would have recognized that Stricklen’s system provides numerous other benefits and advantages further motivating its use when implementing Broch. (*See id.*, ¶ 354.)

For example, Stricklen’s system can be integrated with an “existing organizational structure” like an “enterprise directory.” (Ex. 1006, [0016].) It also includes an initial “bootstrapping” process that allows enterprise management to

dictate the initial configuration of devices. (Ex. 1006, [0059]; Ex. 1003, ¶¶ 354-355.) A POSITA would have understood that this would have improved Broch’s system for enterprise use by improving system security, ensuring uniformity, and using existing enterprise information to simplify the process of defining management policies. (See Ex. 1003, ¶¶ 354-358.).

Stricklen’s system also allows for the use of various “standards” compliant “components.” (Ex. 1006, [0006].) A POSITA would have recognized that implementing Broch’s system to be similarly standard compliant would have simplified its design and rendered it more compatible with a wider range of devices and networks and easier to configure. (See Ex. 1003, ¶¶ 359-360.)

A POSITA would also have had a more than reasonable expectation of success. (See *id.*, ¶¶ 361-364.) Again, Broch is already open to and designed to allow for real-time data gathering, the collection of many different types of device data, and the application of different policies. (See *id.*, ¶¶ 362-363.) Given this, POSITA would have readily understood that Broch’s system could be successfully implemented to include this collection of features. Moreover, given the similarities between Broch’s and Stricklen’s systems—both monitor and manage devices running different operating systems (*see supra* Sections VIII.A, C)—a POSITA would have every reason to believe and no reason to doubt that Stricklen’s teachings could be successfully used when implementing Broch. (See Ex. 1003, ¶ 364.)

X. Ground 2: Claims 1-10, 12, 14-17, and 20-23 Are Obvious Over Mahaffey

Mahaffey (Ex. 1005), when considered in view of the knowledge of a POSITA, renders 1-10, 12, 14-17, and 20-23 obvious.

Mahaffey teaches a system in which a server 151 gathers device data, analyzes the data, applies administrator-defined policies, and triggers responsive actions. A “cross-platform system” allows devices running different operating systems to communicate with the server. The server applies the same set of administrator-defined rules to the different devices.

A. Independent Claims

1. Claim 1

[1-p] “1. A method for providing device management services comprising steps of:”

To the extent limiting, Mahaffey teaches a system and method for “detecting and preventing data from adversely affecting ... mobile communication devices.” (Ex. 1005, [0002].) A “server” is used to “provide” this “protection from ... undesired applications or other data....” (*Id.*, [0021]; *see also* Ex. 1003, ¶¶ 368-370.)

[1-1(a)] “gathering, at a server, from a network, status information related to an operating state of each of a plurality of mobile devices,”

Mahaffey’s system “*gather[s] ... status information related to an operating state ... of mobile devices.*” Both “device data” and “application data” is collected.

(Ex. 1005, [0030]-[0031].) “[D]evice data” “pertains to the state capabilities, operating system, firmware version, memory capacity, available communication ports, battery limitations, hardware characteristics and other ‘baseline’ information” of a particular “mobile communication device.” (*Id.*, [0030].) “Application data includes data objects that are malware or spyware, and thereby can negatively affect a mobile communication device.” (*Id.*, [0032].) It can also include “data objects” that “have coding flaws or other issues that can negatively affect a device,” “compromise[] privacy, overtax[] a device’s battery or network connection, and/or has objectional content.” (*Id.*) “[A]pplication data” also includes “metadata” like “the location” in a “device’s filesystem where a data object is stored” and other “characteristics of how the data object is installed on or integrates with the mobile communication device’s operating system.” (*Id.*, [0033].) “[A]pplication data also includes behavioral data” like “how an application interacts with or uses” device “resources, such as memory usage, battery usage, storage usage, CPU usages, API usage, errors and crashes, network services connected to ..., and runtime library linkage.” (*Id.*, [0034].) Collectively, this data is “**status information**” relating to mobile device “**operating state**”: it indicates applications running on the device, the impact those applications have on device operation and behavior, and the device’s various hardware characteristics. (See Ex. 1003, ¶¶ 371-379.)

Next, in Mahaffey information from ***“each of a plurality of mobile devices”*** is gathered ***“at a server.”*** “FIG. 1” below shows “mobile communication devices 101” (blue) communicating with “servers 151” (red) “over a ... network 121.” (Ex. 1005, [0025].) Each “server 151” “may have access to a plurality of mobile communication devices....” (*Id.*, [0055].) These devices all “send[] application data” (orange) “to server 151....” (*Id.*, [0039]; *see also id.*, ([0075]-[0076] (discussing the “gather[ing]” of “device data”).)

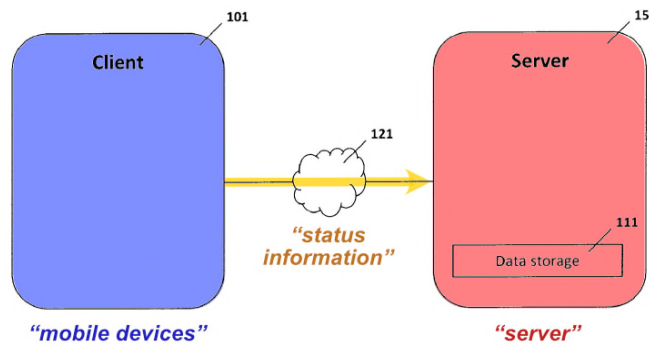


Figure 1

(*Id.*, Fig. 1 (annotated).) Figure 3 also shows the gathering and storage of device data at the server:

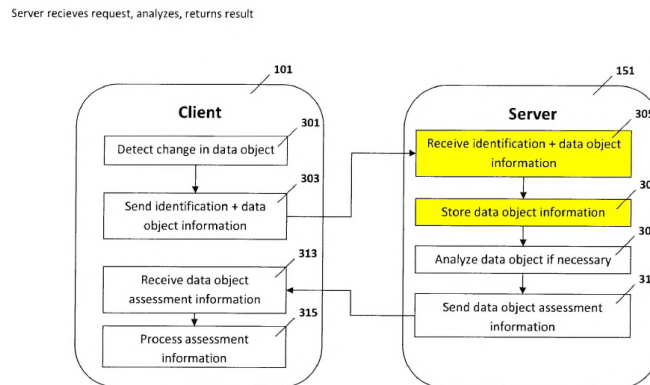


Figure 3

(*Id.*, Fig. 3 (annotated); *see also* Ex. 1003, ¶¶ 380-384.)

[1-1(b)] “each mobile device utilizing one of a plurality of different mobile operating systems,”

Mahaffey’s “mobile communication devices 101” employ **“a plurality of different mobile operating systems”**: the device can be a “handheld or mobile device[]” like a “smartphone[.]” (Ex. 1005, [0022].) This can include “any device running the Apple iOS™, Android™ OS, Google Chrome OS, Symbian OS®, Windows Mobile® OS, Pam OS® or Palm Web OS™.” (*Id.*; *see also id.*, [0199]; Fig. 11; Ex. 1003, ¶¶ 385-389.)

[1-1(c)] “wherein the status information for each mobile device is gathered from a plurality of sources including each mobile device in a substantially real time manner;”

Mahaffey gathers status information **“from a plurality of sources including each mobile device.”** Again, Mahaffey’s system gathers a variety of status information from mobile devices. (*See supra* limitation [1-1(a)].) It also obtains information from other network sources. For instance, “server 151” can “analyze[] user-generated ratings and comments for an application” on a device “such as those gathered by application marketplace data gathering system 1005.” (Ex. 1005, [0095]; *see also id.*, [0100] (explaining that “multiple” other “data sources” are “available to server 151” such as “characterization data, trust data, distribution data, and user-supplied data to determine if an application” on a device “is malicious.”);

Fig. 10 (showing the gathering of data from “disparate data sources”); *See* Ex. 1003, ¶¶ 390-396.)

Next, Mahaffey’s system allows device “status information” to be gathered in **“a substantially real time manner.”** Mahaffey’s system is “continuously updated to provide constant real-time protection” of devices. (Ex. 1005, [0004].) The system also “immediately” analyzes received data. (*Id.*, [0110].) Real-time protection—and immediate analysis—is only possible if data is gathered from devices in real-time as it is generated. (*See* Ex. 1003, ¶¶ 397-398.) Consistent with this, Mahaffey explains that each “mobile communication device 101” can “send[] events to server 151” “whenever it encounters an undesirable application....” (Ex. 1005, [0135]; *see also id.*, [0060].) Thus, data can be transmitted in a non-delayed fashion as it is generated. (*See id.*; *see also* Ex. 1003, ¶¶ 399-400.)

[1-2] “formatting the status information for each of the mobile devices, such that rules can be substantially uniformly applied to the status information across the plurality of different mobile operating systems;”

Mahaffey’s system employs **“rules”** that are **“applied to the status information”** from mobile devices. The system allows an “administrator to set policy [regarding] what applications are desirable to install on a device or group of devices.” (Ex. 1005, [0182].) For instance, “[a] blacklist” of “applications ... explicitly denied from running on a mobile communications device” can be specified. (*Id.*) Alternatively, rather than entirely banning applications, “a policy

may specify certain capabilities (e.g., sending a device's location to the internet) that are blacklisted....” (*Id.*; Ex. 1003, ¶¶ 401-406.)

Next, Mahaffey's rules are intended to be “***substantially uniformly applied ... across the plurality of different mobile operating systems.***” Mahaffey's system allows mobile device protection “regardless of the make or model of the mobile communication device(s).” (Ex. 1005, [0021].) In this regard, the system can perform a “cross-platform assessment” applicable to different systems, like “BlackBerry” and “Android....” (*Id.*, [0167].) “[P]olicies can be set for a specific application, even if the application is available on multiple platforms and has multiple versions.” (*Id.*, [0199].) “[I]f a certain application is known to be undesirable on” different “operating systems” like those employed by “iPhones, BlackBerry devices and Android devices,” “then the IT administrator can block all versions of the application from installation, regardless of platform.” (*Id.*; Ex. 1003, ¶¶ 407-410.)

To accomplish this ability to handle multiple different platforms, data transmitted from mobile communication devices to Mahaffey's server is “***formatt[ed].***” A “consistent form of information between platforms” is used so as to allow the “server 151” to “include[e] some or all of the same fields in assessments for data objects that run on different platforms....” (Ex. 1005, [0167].) This is exactly what this limitation requires: data from different devices running different

operating systems is consistently formatted to allow Mahaffey's server to analyze the data. (See Ex. 1003, ¶¶ 411-413.)

Mahaffey also provides examples of how this consistent information form is achieved: a "cross-platform system" can be employed to allow for use of the system "on a variety of mobile communication device platforms." (Ex. 1005, [0028]; Ex. 1003, ¶¶ 414-416.)

Mahaffey points to (and incorporates by reference) U.S. App. "12/255,626" as detailing one such cross-platform system. (See Ex. 1005, [0028].) This application published as the '591 application (Ex. 1007). The '591 application provides a "cross-platform system" that operates "on any mobile communication device, regardless of manufacture or service provider." (*Id.*, [0005].) The system includes "a platform-independent or 'core' component, a platform specific component, and a lightweight abstraction layer component...." (*Id.*, [0021]; *see also id.*, Fig. 1.) A "common API" "enable[s] communication" between different components "regardless of manufacture, service provider or operating system." (*Id.*, [0024]; *see also id.*, [0030].) Among other functionality, the "common API" includes an "XML parser or generator." (*Id.*, [0030].) Thus, via its incorporation of the '591 application, a POSITA would have understood Mahaffey to teach the use of a cross-platform system that translates platform-specific data into a platform-independent data format—like XML—using an API. (See Ex. 1003, ¶¶ 417-426;

see also Ex. 1008, 7:12-15 (“XML storage solution[s] allow[] data from any arbitrary relational database schema to be loaded, rearranged and retrieved by a data aggregator”); Ex. 1009, 1:38-43 (“XML” serves “as a ‘lingua franca’ ... mak[ing] it possible for data to be exchanged regardless of the platform on which it is stored or the data model in which it is represented”).)

[1-3] “storing the status information in an electronic database accessible to the server;”

Mahaffey’s “server[] 151” has an internal “data storage 111”—that can be a “database, data table, data structure, file system or other memory store”—for storing device information. (Ex. 1005, [0025].) Data storage 111—the claimed “**electronic database**”—is highlighted dark red below:

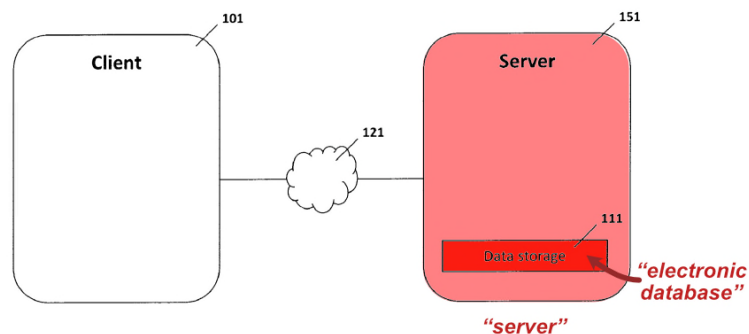


Figure 1

(*Id.*, Fig. 1 (annotated).)

Next, Mahaffey stores “**the status information**” its system previously gathered from mobile devices in data storage 111. As shown in Figure 11, “[i]n

block 1103, application data ... is stored on server 151 or data storage 111” while gathered “device data” is similarly stored at step 1107. (*Id.*, [0075]-[0076].)

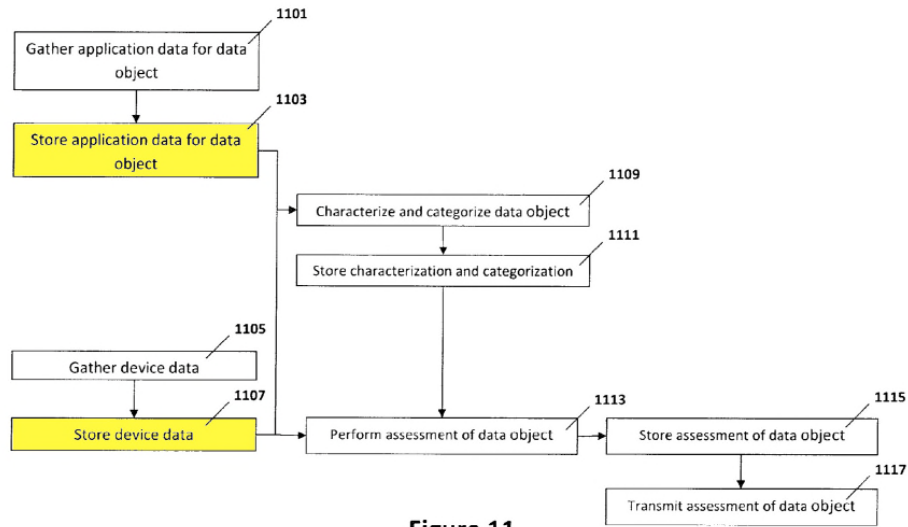


Figure 11

(*Id.*, Fig. 11 (annotated).) This server-storage is also shown in Figure 3:

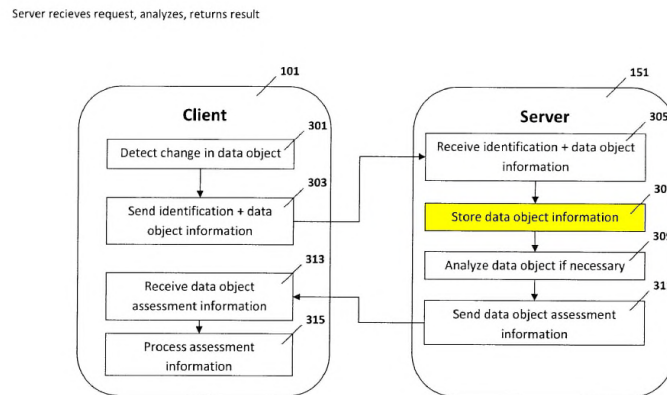


Figure 3

(*Id.*, Fig. 3 (annotated); *see also* Ex. 1003, ¶¶ 427-432.)

[1-4] “evaluating, at the server, a first compliance state of each of the mobile devices from the status information using a plurality of administrator-defined rules, wherein each of the rules applies to mobile devices regardless of mobile operating system; and”

Mahaffey’s system employs **“a plurality of administrator-defined rules”** that are used to assess data from mobile devices: “server 151 allows” an “administrator to set policy for a device or group of devices.” (Ex. 1005, [0186].) Then, “[w]hen a device 101 attempts to install an application, the device sends a request to server 151 for an assessment of the application. Based on policy configured on server 151, the assessment contains an indication of whether the application is allowed or disallowed....” (*Id.*; see also *id.*, [0182] (similarly referencing “administrator ... set policy”). “[S]erver 151” also “allows policy to be configured by assessment criteria....” (*Id.*, [0187].) Thus “an administrator may use server 151 to block all applications that are in a certain category such as social networking applications or all applications that access certain capabilities such as the ability to transmit files or other sensitive data from a device.” (*Id.*)

Next, Mahaffey explains that its administrator-defined and server applied rules apply **“regardless of mobile operating system.”** The policies allow for a “cross-platform assessment” applicable to different systems, like “BlackBerry” and “Android....” (*Id.*, [0167].) Thus, “policies can be set for a specific application, even if the application is available on multiple platforms and has multiple versions.”

(*Id.*, [0199] (explaining that applications running on different “operating systems” can all be blocked); *see also* Ex. 1003, ¶¶ 433-442.)

Finally, in operation, Mahaffey’s system employs these rules to ***“evaluat[e], at the server, a first compliance state of each of the mobile devices from the status information.”*** After each “mobile communication device 101 ... transmit[s] ... certain data to server 151,” the server “will analyze this data and provide a security related assessment, response and/or other action.” (Ex. 1005, [0027].) According to Mahaffey, “both device data and application data are useful for providing” this assessment. (*Id.*, [0035].) The data—and administrator defined rules—are used by the server to perform a variety of assessments. (*See id.*, [0075] (noting that the “server may use a variety of methods of analysis”); *see also* [0153] (similar).) For example, the server can determine if a “data object” running on a device is a “battery waster[.]” (*Id.*, [0086].) It may also “perform analysis” to “determine if” a device “application is considered to be malware or spyware....” (*Id.*, [0106].) Further, device “policy compliance checking can be performed by ... server 151....” (*Id.*, [0191].) A POSITA would understand this policy checking to include an assessment of whether a mobile device includes applications (or application functionality) prohibited by the administrator defined blacklists or other policies. (*See* Ex. 1003, ¶¶ 443-450.) The below annotated version of Figure 11 shows performance of this server assessment at step 1113:

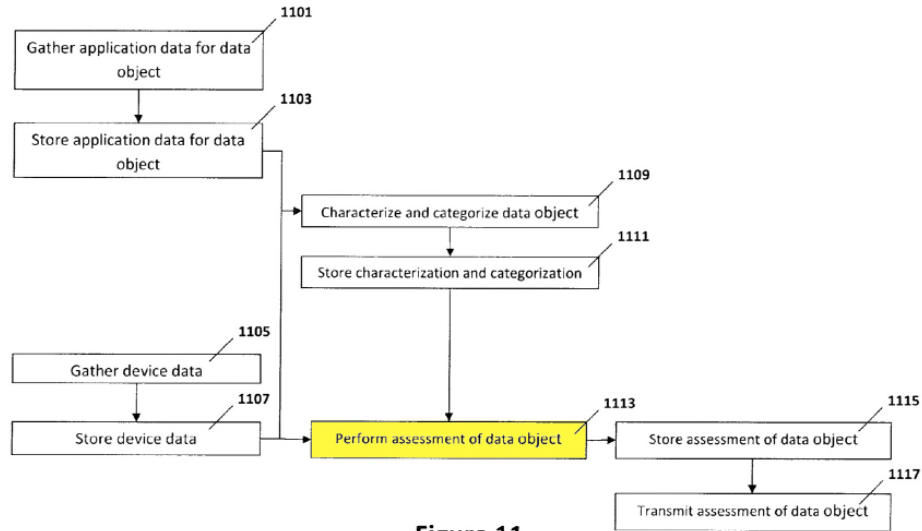


Figure 11

(Ex. 1005, Fig. 11 (annotated).) Figure 3 also shows Mahaffey’s server-based analysis step:

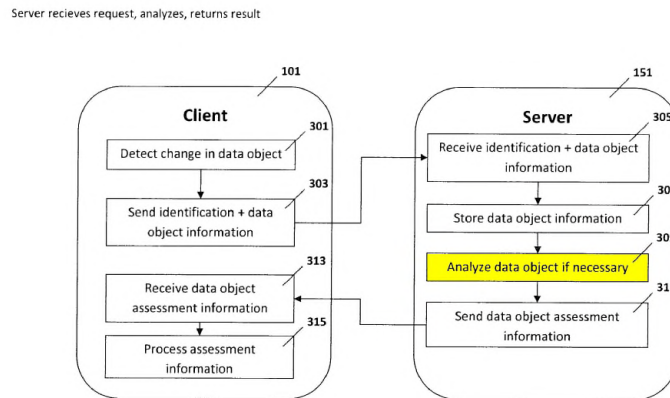


Figure 3

(Id., Fig. 3 (annotated); see also Ex. 1003, ¶¶ 451-452.)

[1-5] “initiating, at the server, at least one action defined by the administrator-defined rules in response to the step of evaluating,”

After performing its evaluation, Mahaffey’s server responsively *“initiat[es]* ... *at least one action defined by the administrator-defined rules.”* For example,

if Mahaffey’s server detects “malicious” data on a mobile device, it “may notify the ... device or instruct the device to reject the data object and not process it further.” (Ex. 1005, [0137].) Likewise, if the server determines that a “device is ... violating policy, a variety of remediation actions are possible”: “any violating applications may have their processes ended, may be uninstalled or isolated from accessing certain system functionality (e.g., internet, private data), or may be restricted from accessing certain networks (e.g., only allowed to access Wi-Fi, not the cellular network). It may also be desirable to isolate the whole device from accessing sensitive resources such as a corporate email or VPN server while it is out of compliance to prevent information leakage.” (*Id.*, [0195]; *see also id.*, [0040] (“server 151” can “send[] a notification” in the form of “an alert, a message, an instruction or other information”); [0048].) This responsive, server-performed action is shown as step 1117 in Figure 11:

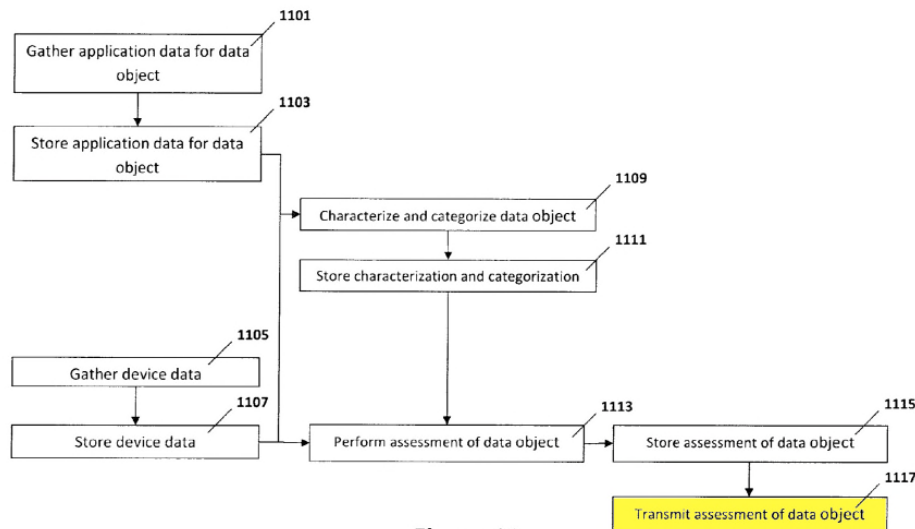


Figure 11

(*Id.*, Fig. 11 (annotated).) Figure 3 also shows an example server-performed notification step:

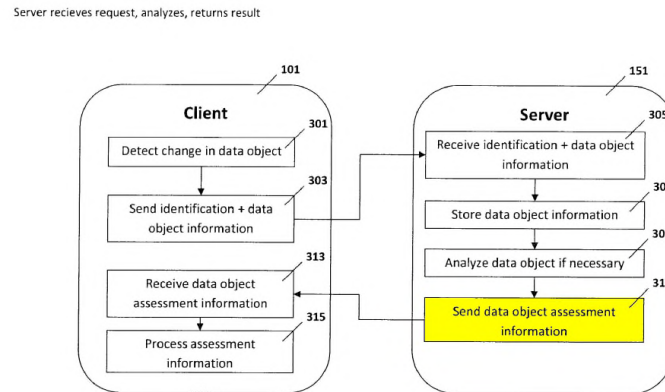


Figure 3

(*Id.*, Fig. 3 (annotated).) As shown, the responsive action performed by the server occurs immediately after the server’s completion of the assessment/analysis step. (See *id.*, Figs. 3, 11.) Consistent with this, Mahaffey explains that its system operates to provide “real-time protection” of devices. (*Id.*, [0004]; see also Ex. 1003, ¶¶ 453-461.)

[1-6] “wherein the step of evaluating is performed by the server automatically, in response to changes in the status information in the electronic database.”

Mahaffey’s system allows for “automated gathering of data object information” and operates as an “automated analysis system[.]...” (Ex. 1005, [0073], [0104].) Thus, “server 151 may analyze” a received “data object immediately....” (*Id.*, [0110]; see also Fig. 11 (showing the performance of assessment immediately after data gathering and storage).) Because it automatically and immediately analyzes received device data, Mahaffey’s server engages in

“evaluating” “automatically, in response to changes in the status information in the electronic database” as required. (See Ex. 1003, ¶¶ 462-466.)

2. Claim 10

Claim 10 is another independent claim.

[10-p] “10. A method for providing device management services comprising steps of:”

Mahaffey teaches a server-based system for managing and protecting mobile devices. (See *supra* preamble element [1-p]; Ex. 1003, ¶¶ 468-469.)

[10-1(a)] “receiving, at a server, a plurality of attributes related to an operating state of each of a plurality of mobile devices,

Mahaffey’s system gathers various *“attributes related to an operating state”* of mobile devices. This includes “device data” like state capabilities, operating system, firmware version, memory capacity, available communication ports, battery limitations, and hardware characteristics. (See *supra* limitation [1-1(a)].) It also includes “application data” like the presence of applications, malware, or spyware, along with flawed data objects, metadata, and device behavioral data. (See *id.*; see also Ex. 1003, ¶¶ 470-473.)

[10-1(b)] “each mobile device utilizing one of a plurality of different mobile operating systems,”

Mahaffey’s system manages devices running various different operating systems. (See *supra* limitation [1-1(b)]; Ex. 1003, ¶¶ 474-475.)

[10-1(c)] “wherein the attributes for each mobile device are gathered from a plurality of sources including each mobile device;”

Mahaffey gathers data not only from a plurality of mobile devices, but also from various other sources such as a “marketplace data gathering system 1005,” or external “characterization data, trust data, distribution data, and user supplied data.” (See *supra* limitation [1-1(c)]; Ex. 1003, ¶¶ 476-477.)

[10-2] “making a determination that an attribute of a mobile device has changed;”

Mahaffey “*determin[es]*” that device “*attribute[s]*” have “*changed.*” In Mahaffey’s system, “mobile communication device 101 detects a change in a specific data object” and communicates this to the server. (Ex. 1005, [0042]; *see also id.*, [0043] (“any time application data for a data object has not been transmitted from mobile communication device 101 and received by server 151, it is considered to be changed and needs to be transmitted”); [0050] (referencing the “detect[ing] that a data object has changed” and requires transmission to the server).) The server in turn monitors for and then evaluates this “change in a data object stored on mobile communication device 101.” (*Id.*, [0042], [0050], [0053]; Figs. 5-6.) This data gathering and analysis process can be “automated.” (*Id.*, [0073], [0104]; *see also* Ex. 1003, ¶¶ 478-486.) Figure 3 provides an example of automatic detection of data changes on a mobile device, along with the relaying of this detected change to the server for analysis:

Server receives request, analyzes, returns result

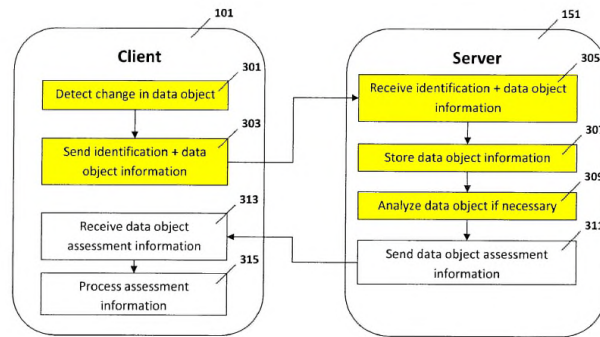


Figure 3

(Ex. 1005, Fig. 3 (annotated).)

[10-3] “evaluating the attribute, by a processor at the server in response to the determination, to determine if the attribute change triggers at least one of a plurality of administrator-defined rules, which indicates that the attribute is out of compliance, wherein each of the rules applies to mobile devices regardless of mobile operating system; and”

Mahaffey’s system employs a **“plurality of administrator-defined rules”** that apply to monitored devices **“regardless of mobile operating system.”** (See *supra* limitation [1-4].) This includes policies permitting or prohibiting certain applications or application functionality. (See *id.*)

Next, Mahaffey’s system also **“determine[s] if the attribute change triggers at least one of a plurality of administrator-defined rules”** because **“the attribute is out of compliance.”** Again, Mahaffey’s server monitors and analyzes changes in mobile device data. (E.g., Ex. 1005, [0042], [0050], [0053].) And, Mahaffey’s server uses administrator-defined rules to analyze this changed device data for purposes of “provid[ing] a security related assessment, response and/or other

action.” (*Id.*, [0027], [0035], [0075], [0153].) This allows the server to identify the presence of policy-violating data on mobile device—like blacklisted applications, battery wasting applications, or malware—or otherwise determine that the mobile device is out of “policy compliance.” (*Id.*, [0086], [0106], [0191]; *see also supra* limitation [1-4]; Ex. 1003, ¶¶ 487-493.)

[10-4] “automatically initiating, at the server, an action defined by the at least one of a plurality of administrator defined rules if the attribute is out of compliance,”

Mahaffey’s system automatically initiates a response (like a notification or remediation action) at the server in response to the performed analysis. (*See supra* limitations [1-5]-[1-6]; *see also* Ex. 1003, ¶¶ 494-495.)

[10-5(a)] “wherein the attribute is stored in a data store accessible to the server, that includes substantially real-time data, gathered from the plurality of sources on a network and pertaining to a plurality of attributes of a plurality of mobile devices,

To begin, Mahaffey’s system includes the claimed “***data store accessible to the server***”: the “server[] 151” has an internal “data storage 111” that stores transmitted device information. (Ex. 1005, [0025]; Fig. 1; *see also supra* limitation [1-3].) The stored data also includes “***substantially real-time data***.” Mahaffey’s system operates to “provide constant real-time protection” of devices and allows for data transmission and analysis “whenever it [is] encounter[ed]” by the system. (Ex. 1005, [0004]; [0135]; *see also supra* limitation [1-1(c)].) And, Mahaffey teaches both “***gather[ing]***” data “***from the plurality of sources***” and data “***pertaining to a***

plurality of attributes of a plurality of mobile devices”: the system gathers device/application data from many different mobile devices. (*See supra* limitation [1-1(c)].) It also gathers data from various external sources like marketplace, characterization, trust, distribution, and user supplied data. (*See id.*; Ex. 1003, ¶¶ 496-502.)

[10-5(b)] “wherein the substantially real-time data is formatted, such that the rules can be substantially uniformly applied to the plurality of attributes across the plurality of different mobile operating systems.”

Mahaffey’s server is provided with “a consistent form of information”: the system employs a “cross-platform system” that formats data generated by devices running different operating systems by, for instance, converting it to XML. (*See supra* limitation [1-2]; Ex. 1003, ¶¶ 503-505.)

3. Claim 17

Claim 17 is yet another independent claim.

[17-p] “17. A system for providing device management services comprising:”

This is taught by Mahaffey as discussed for preamble elements [1-p] and [10-p]. (*See* Ex. 1003, ¶ 507.)

[17-1] “a data store configured to be accessible by a server;”

Mahaffey’s server includes a data storage 111. (*See supra* limitations [1-3], [10-5(a)]; Ex. 1003, ¶¶ 508-509.)

[17-2] “a CPU of the server coupled to a memory configured to execute computer readable instructions;”

[17-3] “computer readable instructions for execution on the server comprising instructions configured to:”

Mahaffey’s server 151 “operate[]” as a typical “HTTP server.” (Ex. 1005, [0026].) The server can employ “computer readable storage medium”—like RAM, ROM, or a “persistent store”—that includes “computer readable instructions or computer program code.” (*Id.*, [0022]-[0023].) Moreover, the server engages in operations such as data “analy[sis],” and can generate “a security related assessment, response and/or other action.” (*E.g., id.*, [0027].) Because it is a typical server and engages in data analysis/assessment, a POSITA would have understood that Mahaffey’s server 151 includes a “**CPU**” and coupled “**memory**” with “**computer readable instructions.**” (See Ex. 1003, ¶¶ 510-517; see also Ex. 1010, [0025] (“[A] typical server includes a processing unit 101, a memory 102, a storage unit 103 and a network interface unit 104....”); Ex. 1011, 7:63-64 (“A typical server includes an IBM RISC/6000 with AIX operating system and server program”); Fig. 5 (showing a “server” with a “computer” and “application”).)

[17-3(a)] “automatically make a determination that an attribute of a mobile device stored in the data store has changed;”

Mahaffey’s system is automated, and monitors for mobile device attribute changes. (See *supra* limitations [1-6], [10-2]; see also Ex. 1003, ¶¶ 518-519.)

[17-3(b)] “evaluate the attribute, in response to the determination, to determine if the attribute change triggers at least one of a plurality of administrator-defined rules, which indicates that the attribute is out of compliance, wherein each of the rules applies to mobile devices regardless of mobile operating system; and”

Mahaffey’s server evaluates device/application data using administrator-defined rules to identify, among other things, device policy compliance. (*See supra* limitations [1-4], [10-3]; Ex. 1003, ¶¶ 520-521.)

[17-3(c)] “automatically initiate an action defined by the at least one of a plurality of administrator defined rules if the attribute is out of compliance,”

Mahaffey’s system automatically initiates a response (like a notification or remediation action) at the server in response to the performed analysis. (*See supra* limitations [1-5]-[1-6]; Ex. 1003, ¶¶ 552-523.)

[17-3(d)] “wherein the data store is configured to gather substantially real-time data pertaining to a plurality of attributes related to an operating state of each of the plurality of mobile devices, the attributes for each mobile device being gathered from a plurality of sources including each mobile device, wherein each mobile device utilizes one of a plurality of different mobile operating systems, and the substantially real-time data is formatted, such that the rules can be substantially uniformly applied to the plurality of attributes across the plurality of different mobile operating systems.”

Mahaffey’s system gathers data in **“substantially real-time”** as that data is generated. (Ex. 1005, [0004], [0060], [0135]; *see also supra* limitation [1-1(c)].)

Next, the gathered **“attributes”** come from **“a plurality of sources including each mobile device”**: as explained, Mahaffey collects both device/application data, and data from other external sources. (*See supra* limitation [1-1(c)].) Mahaffey’s system is also designed to work with different mobile devices running **“different mobile**

operating systems.” (See Ex. 1005, [0022]; see also supra limitation [1-1(b)].)

Further, Mahaffey **“format[s]”** data to allow **“rules”** to **“be substantially uniformly applied”**: Mahaffey’s servers provided with “a consistent form of information” and it makes use of a “cross-platform system” that can format data generated by devices running different operating systems by, for instance, converting it to XML. (See *supra* limitation [1-2]; Ex. 1003, ¶¶ 524-529.)

B. Dependent Claims

1. Claim 2

Claims 2 depends on claims 1 and further requires that:

“wherein the at least one action is selectable from a plurality of actions, comprising blocking access by the at least one of the mobile devices to a corporate resource”

After assessing device data, Mahaffey’s server can perform “a variety of remediation actions.” (Ex. 1005, [0195].) This includes “isolat[ing] the ... device from accessing ... corporate email or VPN server while it is out of compliance....” (*Id.*; see also Ex. 1003, ¶¶ 530-533.)

2. Claim 3

Claims 3 depends on claims 1 and further requires that:

“wherein the at least one action is selectable from a plurality of actions, comprising blocking e-mail access for the at least one of the mobile devices.”

Again, one of the actions Mahaffey’s server can take is blocking access to “corporate email.” (Ex. 1005, [0195]; see also Ex. 1003, ¶¶ 534-536.)

3. Claim 4

Claims 4 depends on claims 1 and further requires that:

“wherein the at least one action is selectable from a plurality of actions, comprising wiping information from the at least one of the mobile devices.”

Mahaffey’s server can also engage in “*wiping information*”: it causes “any violating applications” to be “uninstalled” from a managed device. (Ex. 1005, [0195]; *see also* Ex. 1003, ¶¶ 537-539.)

4. Claim 5

Claims 5 depends on claims 1 and further requires that:

“wherein the at least one action is selectable from a plurality of actions, comprising warning a user of the at least one of the mobile devices that further action may be taken.”

Mahaffey’s server can be configured to “transmit notifications or instructions to remediate data objects whose assessment has changed, thereby requiring action by” a mobile device. (Ex. 1005, [0047].) This notification can be in the form of an “alert, a message, [or] an instruction.” (*Id.*, [0040]; *see also id.*, [0131] (explaining that a server can provide a device with a “notification, a warning, a risk rating, or the like”).) Mahaffey’s system is also able to take other actions such as “isolat[ing]” the device from network resources, or “uninstall[ing]” applications. (*Id.*, [0195].) A POSITA would have understood that failure of a device to comply with a previously provided notification/instruction would obviously lead to Mahaffey’s server taking one of these more serious actions. (*See* Ex. 1003, ¶¶ 540-546.)

5. Claims 6, 14, and 21

Claims 6, 14, and 21 respectively depend on claims 1, 10, and 17. They all similarly require:

“gathering [attribute] information from that mobile device and at least one other source on the network”

As discussed, Mahaffey gathers information from both mobile devices and other network sources like, data from a “marketplace data gathering system 1005,” or external “characterization data, trust data, distribution data, and user supplied data.” (*See supra* limitation [1-1(c)]; *see also* Ex. 1003, ¶¶ 547-549.)

6. Claims 7, 12, and 20

Claims 7, 12, and 20 respectively depend on claims 1, 10, and 17 and further require:

“the step of subsequently evaluating a second compliance state of at least one of the mobile devices to determine if the at least one action should be revoked” (claim 7)

“[the step of]/[wherein the computer readable instructions further comprise instructions configured to] automatically [revoking]/[revoke] the action if the attribute is brought back into compliance” (claims 12, 20)

Mahaffey’s system can “isolat[e]” a monitored device “from accessing ... corporate email or VPN server while it is out of compliance....” (Ex. 1005, [0195].) This—and Mahaffey’s other responsive actions—are “automated.” (*See id.*, [0073], [0104].) Because isolation occurs only when the device is “out of compliance,” a POSITA would have understood that the monitored device would obviously be

automatically re-allowed to access these resources once it is determined to be compliant. (See Ex. 1003, ¶¶ 550-555.)

7. Claims 8, 15, and 22

Claims 8, 15, and 22 respectively depend on claims 1, 10, and 17. They all further requires that:

“the plurality of different mobile operating systems comprise at least operating systems compliant with Android and iOS specifications”

Mahaffey’s system applies to and can be used with devices running both “Apple iOS™” and “Android™ OS.” (Ex. 1005; [0022]; *see also* Ex. 1003, ¶¶ 556-558.)

8. Claims 9, 16, and 23

Claims 9, 16, and 23 respectively depend on claims 1, 10, and 17 and further relatedly require:

“wherein the [status information / attribute] comprises location information about [each / the] mobile device”

According to Mahaffey, its “server may use a device’s country ... or area code” when making an assessment. (Ex. 1005, [0117].) Both country and area code are examples of device **“location information.”** (See Ex. 1003, ¶¶ 559-562.)

C. Motivation to Combine

A POSITA would have considered it obvious and would have been motivated to apply Mahaffey’s teachings in the manner discussed above to arrive at the claimed subject matter. (See Ex. 1003, ¶¶ 563-580.)

As an initial matter, if it is somehow determined that Mahaffey did not properly incorporate the '591 application by reference, a POSITA would have still considered it obvious to apply its teachings to Mahaffey. Mahaffey explains that its system is intended to gather data from different devices running different operating systems. (See Ex. 1005, [0022].) It also notes that a “cross-platform system” can be used to facilitate this type of communication. (*Id.*, [0028].) And, Mahaffey points to the '591 application as providing an example of one such “cross-platform system.” (See *id.*) This explicit instruction would have provided a POSITA with a strong motivation to apply the '591 application to Mahaffey. (See Ex. 1003, ¶¶ 564-568.)

Further, the '591 application itself articulates numerous benefits deriving from the use of its cross-platform system. In particular, the '591 application's cross-platform system is “lightweight” “without sacrificing functionality or power.” (Ex. 1007, [0021].) Thus, it allows communication between different devices without consuming excessive system resources. (See *id.*, [0004], [0021].) Further, it allows for the use of “a common API” for communication instead of “different APIs as is common in the prior art.” (*Id.*, [0021].) This allows for the re-use of code and avoids the need for excessive and difficult customization. (See *id.*; see also Ex. 1003, ¶¶ 569-571.) This would also have motivated the application of the '591 application's cross-platform system (and data formatting procedures) to Mahaffey. Doing so

would have improved Mahaffey by reducing the resources consumed when communicating between different types of devices. (*See id.*, ¶¶ 572-573.) And, use of common APIs would have rendered the code underlying Mahaffey less complicated, more efficient, and less prone to error. (*See id.*, ¶ 574.)

Next, while Mahaffey discusses a number of more specific examples—like a system for collecting data (*e.g.*, Ex. 1005, [0069]), a server-side analysis system (*e.g.*, *id.*, [0075]), an anti-malware system (*e.g.*, *id.* [0106]), and an application assessment and advisement system (*e.g.*, *id.*, [0163])—a POSITA would have understood that these are simply meant to be representative of the overall operation and capabilities of Mahaffey’s system. (*See* Ex. 1003, ¶¶ 575-580.) None are meant to be used in isolation. (*See id.*) Mahaffey explains that its system is generally able to “provid[e] security to a plurality of mobile communication devices.” (Ex. 1005, [0021].) The system is designed to flexibly allow for the collection of many different types of device and application data, engage in a variety of different types of analyses, and generate various different responses. (*See id.*, [0030], [0075], [0195].) As a result, a POSITA would have been motivated to use the features of these various example systems together. (*See* Ex. 1003, ¶¶ 579-580.)

A POSITA would also have had a more than reasonable expectation of success. (*See* Ex. 1003, ¶¶ 581-584.) Again, Mahaffey explains that its system is designed for and meant to use the cross-platform system of the ’591 application.

(See Ex. 1005, [0028].) Moreover, Mahaffey itself notes that its various different embodiments are non-limiting, and are just examples of the overall capabilities of its system. (See *e.g.*, *id.*, [0210].) Given this, a POSITA would have every reason to believe—and no reason to doubt—that Mahaffey’s system could be successfully designed to not only employ the ’591 patent’s cross-platform system, but provide all the functionality set forth in Mahaffey. (See Ex. 1003, ¶ 584.)

XI. Conclusion

For the foregoing reasons, Petitioner respectfully requests institution of IPR and cancellation of claims 1-10, 12, 14-17, and 20-23.

Respectfully submitted,

ORRICK, HERRINGTON & SUTCLIFFE LLP

Dated: 10/31/2023

/K. Patrick Herman/

K. Patrick Herman, Reg. No. 75,018
Orrick, Herrington & Sutcliffe LLP
51 West 52nd Street
New York, NY 10019
Telephone: (212) 506-3596
Email: P52PTABDocket@orrick.com

Attorney for Petitioner

CERTIFICATE OF WORD COUNT

The undersigned certifies that the foregoing PETITION FOR *INTER PARTES* REVIEW complies with the type volume limitation in 37 C.F.R. § 42.24(a). According to the utilized word-processing system's word count, the petition—excluding the caption, table of contents, table of exhibits, mandatory notices, certificate of word count, and certificate of service—contains 13,975 words.

Dated: 10/31/2023

By: /K. Patrick Herman/

K. Patrick Herman, Reg. No. 75,018
Orrick, Herrington & Sutcliffe LLP
51 West 52nd Street
New York, NY 10019
Telephone: (212) 506-3596
Email: P52PTABDocket@orrick.com

Attorney for Petitioner

CERTIFICATE OF SERVICE

The undersigned hereby confirms that the foregoing PETITION FOR IPR and associated exhibits was caused to be served on October 31, 2023 via overnight courier upon the following counsel of record for Patent Owner:

INTERNATIONAL BUSINESS MACHINES
CORPORATION
Intellectual Property Law Department
T.J. Watson Research Center
1101 Kitchawan Road, Route 134, P.O. Box 218
YORKTOWN, NY 10598

Courtesy copies were also served via electronic mail on Patent Owner's counsel of record in a related district court litigation:

Alfred R. Fabricant (ffabricant@fabricantllp.com)
Joseph M. Mercadante (jmercadante@fabricantllp.com)
Julian G. Pymonto (jpymonto@fabricantllp.com)
Peter Lambrianakos (plambrianakos@fabricantllp.com)
Vincent J. Rubino, III (vrubino@fabricantllp.com)
FABRICANT LLP

/K. Patrick Herman/

K. Patrick Herman, Reg. No. 75,018
Orrick, Herrington & Sutcliffe LLP
51 West 52nd Street
New York, NY 10019
Telephone: (212) 506-3596
Email: P52PTABDocket@orrick.com