

Petition for *Inter Partes* Review of
U.S. Patent No. 9,542,571 B2

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

NINTENDO CO., LTD., and
NINTENDO OF AMERICA INC.,
Petitioners,

v.

MALIKIE INNOVATIONS LTD.,
Patent Owner.

Case IPR2026-00007
U.S. Patent No. 9,542,571 B2
Issue Date: January 10, 2017

Title: SYSTEM AND METHOD OF OWNER APPLICATION CONTROL OF
ELECTRONIC DEVICES

**PETITION FOR *INTER PARTES* REVIEW
OF U.S. PATENT NO. 9,542,571 B2**

Table of Contents

	Page
I. MANDATORY NOTICES UNDER §42.8(A)(1)	1
A. Real Party-In-Interest Under §42.8(b)(1).....	1
B. Related Matters Under §42.8(b)(2)	1
C. Lead and Back-Up Counsel under §42.8(b)(3).....	1
D. Service Information	2
E. Power of Attorney	3
II. FEE PAYMENT	3
III. REQUIREMENTS UNDER §§ 42.104 AND 42.108 AND CONSIDERATIONS UNDER §§ 314(A) AND 325(D).....	3
A. Grounds for Standing	3
B. Identification of Challenge and Statement of Precise Relief Requested	3
C. Considerations Under §§ 314(a) and 325(d).....	4
IV. '571 PATENT OVERVIEW	4
A. POSITA	4
B. Specification.....	5
V. CLAIM CONSTRUCTION	5
VI. PRIOR ART.....	5
A. Status	5
B. Overview of Grounds	6
VII. GROUNDS 1-2: CLAIMS 1-20 ARE OBVIOUS OVER LAMBERT (GROUND 1) OR OBVIOUS OVER LAMBERT IN VIEW OF FRANK, GOMBERG, AND WINDOWS GUIDELINES BOOK (GROUND 2).....	8
A. Claim 1:	8
1. “A method of application control on an electronic device, the method comprising:” (Claim 1[p]).....	8

Table of Contents

(continued)

	Page
2. “receiving, by the electronic device, owner information from an owner of the electronic device;” (Claim 1[a]) and “receiving, by the electronic device, owner control information that comprises:” (Claim 1[b])	9
(a) “a list of applications permitted for installation on the electronic device, and” (Claim 1[b][i]).....	12
(b) “permissions associated with each permitted application in the list of applications;” (Claim 1[b][ii])	15
3. “storing, by the electronic device, the owner control information in a protected data store to prevent alteration or deletion of the owner control information in response to authenticating the owner control information using the owner information;” (Claim 1[c]).....	18
4. “receiving, by the electronic device, from an application operating on the electronic device, a request to access local memory on the electronic device;” (Claim 1[d])	22
5. “determining whether the application is allowed to access the local memory in response to identifying a permission associated with the application in the owner control information; and” (Claim 1[e])	26
6. “controlling application access to the local memory based upon the determining.” (Claim 1[f])	29
B. Claim 2: “The method of claim 1, wherein the request to access the local memory comprises a request to erase data from the local memory.”	33
C. Claim 3: “The method of claim 1, wherein the request to access the local memory comprises a request to read data from the local memory.”	33
D. Claim 4: “The method of claim 1, further comprising restricting modification of the owner control information after the owner control information has been stored in the protected data store.”	34

Table of Contents
(continued)

	Page
E. Claim 5: “The method of claim 1, wherein the owner control information includes an authorization record and an application identifier for each of the applications, and the owner control information is provided by an external source through a wireless communication channel.”	35
F. Claim 6: “The method of claim 5, wherein the authorization record related to determining whether access to the local memory is allowed is provided by an external computer that is used to enforce policies for operating electronic devices within an organization.”	37
G. Claim 7: “The method of claim 1, wherein an update to the owner control information is provided by an external source for storage on the electronic device.”	37
H. Claim 8	38
1. “The method of claim 1, wherein a plurality of authorization records associated with a plurality of application identifiers are stored on the electronic device, the method further comprising:” (Claim 8[p]).....	38
2. “receiving requests to access the local memory from a plurality of applications operating on the electronic device;” (Claim 8[a])	39
3. “determining whether access to the local memory is allowed to be performed by their respective applications based upon the stored authorization records and the application identifiers that are respectively associated with the plurality of applications; and:” (Claim 8[b])	39
4. “controlling access by the plurality of applications to local memory based upon whether access to the local memory is allowed.” (Claim 8[c])	40
I. Claim 9: “The method of claim 8, wherein a group containing two or more of the applications is associated with one of the authorization records.”	40
J. Claim 10	42

Table of Contents

(continued)

	Page
1. “A device, comprising:” (Claim 10[p]); “a memory; and” (Claim 10[a]); “a processor coupled with the memory and configured by instructions stored in the memory to:” (Claim 10[b]).....	42
2. “receive owner information from an owner of the device;” (Claim 10[b][i]) and “receive owner control information that comprises:” (Claim 10[b][ii]); “a list of applications permitted for installation on the device, and” (Claim 10[b][ii][A]); “permissions associated with each permitted application in the list of applications;” (Claim 10[b][ii][B]); “store the owner control information in a protected data store to prevent alteration or deletion of the owner control information in response to authenticating the owner control information using the owner information” (Claim 10[b][iii]); “receive, from an application operating on the electronic device, a request to access local memory on the electronic device;” (Claim 10[b][iv]); “determine whether the application is allowed to access the local memory in response to identifying a permission associated with the application in the owner control information; and” (Claim 10[b][v]); “control application access to the local memory based upon the determining.” (Claim 10[b][vi]).....	44
K. Claim 11: “The device of claim 10, wherein the request to access the local memory comprises a request to erase data from the local memory.”	44
L. Claim 12: “The device of claim 10, wherein the request to access the local memory comprises a request to read data from the local memory.”	44
M. Claim 13: “The device of claim 10, wherein the processor is further configured to restrict modification of the owner control information after the owner control information has been stored on the electronic device.”	45

Table of Contents
(continued)

	Page
N. Claim 14: “The device of claim 10, wherein the owner control information includes an authorization record and an application identifier for each of the applications, and wherein the owner control information is provided by an external source through a wireless communication channel.”	45
O. Claim 15: “The device of claim 14, wherein the authorization record related to determining whether access to the local memory is allowed is provided by an external computer that is used to enforce policies for operating electronic devices within an organization.”	46
P. Claim 16: “The device of claim 10, wherein an update to the owner control information is provided by an external source for storage on the electronic device.”	46
Q. Claim 17	46
1. “A non-transitory computer-readable medium with instructions stored thereon, wherein the instructions are executable by a processor to cause the processor to perform the operations of:” (Claim 17[p]).....	46

Table of Contents

(continued)

	Page
2. “receiving owner information from an owner of an electronic device;” (Claim 17[a]) and “receiving owner control information that comprises:” (Claim 17[b]); “a list of applications permitted for installation on the device, and” (Claim 17[b][i]); “permissions associated with each permitted application in the list of applications;” (Claim 17[b][ii]); “storing the owner control information in a protected data store to prevent alteration or deletion of the owner control information in response to authenticating the owner control information using the owner information;” (Claim 17[c]); “receiving, from an application operating on the electronic device, a request to access local memory on the electronic device;” (Claim 17[d]); “determining whether the application is allowed to access the local memory in response to identifying a permission associated with the application in the owner control information; and” (Claim 17[e]); “controlling application access to the local memory based upon the determining.” (Claim 17[f])	49
R. Claim 18: “The non-transitory computer-readable medium of claim 17, wherein the owner control information includes an authentication record and an identifier for each application allowed to be executed on the electronic device.”	49
S. Claim 19: “The non-transitory computer-readable medium of claim 17, wherein the request to access the local memory comprises a request to erase data from the local memory.”	51
T. Claim 20: “The non-transitory computer-readable medium of claim 17, wherein the request to access the local memory comprises a request to read data from the local memory.”	51
VIII. GROUNDS 3-5: CLAIMS 1-8 AND 10-20 ARE OBVIOUS OVER HAYES IN VIEW OF SPRIGG (GROUND 3), OR OBVIOUS OVER GROUND 3 + FRANK AND GUEDALIA (GROUND 4); CLAIMS 9-20 ARE OBVIOUS OVER GROUND 4 + LAMBERT (GROUND 5)	51
A. Claim 1	51

Table of Contents

(continued)

	Page
1. “A method of application control on an electronic device, the method comprising:” (Claim 1[p]).....	51
2. “receiving, by the electronic device, owner information from an owner of the electronic device;” (Claim 1[a]).....	51
3. “receiving, by the electronic device, owner control information that comprises:” (Claim 1[b]) and “a list of applications permitted for installation on the electronic device, and” (Claim 1[b][i]).....	54
4. “permissions associated with each permitted application in the list of applications;” (Claim 1[b][ii])	58
5. “storing, by the electronic device, the owner control information in a protected data store to prevent alteration or deletion of the owner control information in response to authenticating the owner control information using the owner information;” (Claim 1[c])	62
6. “receiving, by the electronic device, from an application operating on the electronic device, a request to access local memory on the electronic device;” (Claim 1[d])	63
7. “determining whether the application is allowed to access the local memory in response to identifying a permission associated with the application in the owner control information; and” (Claim 1[e])	65
8. “controlling application access to the local memory based upon the determining.” (Claim 1[f])	67
B. Claim 2: “The method of claim 1, wherein the request to access the local memory comprises a request to erase data from the local memory.”	69
C. Claim 3: “The method of claim 1, wherein the request to access the local memory comprises a request to read data from the local memory.”	69
D. Claim 4: “The method of claim 1, further comprising restricting modification of the owner control information after the owner control information has been stored in the protected data store.”	69

Table of Contents
(continued)

	Page
E. Claim 5: “The method of claim 1, wherein the owner control information includes an authorization record and an application identifier for each of the applications, and the owner control information is provided by an external source through a wireless communication channel.”	71
F. Claim 6: “The method of claim 5, wherein the authorization record related to determining whether access to the local memory is allowed is provided by an external computer that is used to enforce policies for operating electronic devices within an organization.”	73
G. Claim 7: “The method of claim 1, wherein an update to the owner control information is provided by an external source for storage on the electronic device.”	74
H. Claim 8	75
1. “The method of claim 1, wherein a plurality of authorization records associated with a plurality of application identifiers are stored on the electronic device, the method further comprising:” (Claim 8[p]).....	75
2. “receiving requests to access the local memory from a plurality of applications operating on the electronic device;” (Claim 8[a])	75
3. “determining whether access to the local memory is allowed to be performed by their respective applications based upon the stored authorization records and the application identifiers that are respectively associated with the plurality of applications; and:” (Claim 8[b])	76
4. “controlling access by the plurality of applications to local memory based upon whether access to the local memory is allowed.” (Claim 8[c])	76
I. Claim 9: “The method of claim 8, wherein a group containing two or more of the applications is associated with one of the authorization records.”	77
J. Claim 10	78
1. “A device, comprising:” (Claim 10[p])	78

Table of Contents
(continued)

	Page
2. “a memory; and” (Claim 10[a])	78
3. “a processor coupled with the memory and configured by instructions stored in the memory to:” (Claim 10[b])	80
4. “receive owner information from an owner of the device;” (Claim 10[b][i]) and “receive owner control information that comprises:” (Claim 10[b][ii]); “a list of applications permitted for installation on the device, and” (Claim 10[b][ii][A]); “permissions associated with each permitted application in the list of applications;” (Claim 10[b][ii][B]); “store the owner control information in a protected data store to prevent alteration or deletion of the owner control information in response to authenticating the owner control information using the owner information” (Claim 10[b][iii]); “receive, from an application operating on the electronic device, a request to access local memory on the electronic device;” (Claim 10[b][iv]); “determine whether the application is allowed to access the local memory in response to identifying a permission associated with the application in the owner control information; and” (Claim 10[b][v]); “control application access to the local memory based upon the determining.” (Claim 10[b][vi]).....	82
K. Claim 11: “The device of claim 10, wherein the request to access the local memory comprises a request to erase data from the local memory.”	82
L. Claim 12: “The device of claim 10, wherein the request to access the local memory comprises a request to read data from the local memory.”	82
M. Claim 13: “The device of claim 10, wherein the processor is further configured to restrict modification of the owner control information after the owner control information has been stored on the electronic device.”	83

Table of Contents
(continued)

	Page
N. Claim 14: “The device of claim 10, wherein the owner control information includes an authorization record and an application identifier for each of the applications, and wherein the owner control information is provided by an external source through a wireless communication channel.”	83
O. Claim 15: “The device of claim 14, wherein the authorization record related to determining whether access to the local memory is allowed is provided by an external computer that is used to enforce policies for operating electronic devices within an organization.”	84
P. Claim 16: “The device of claim 10, wherein an update to the owner control information is provided by an external source for storage on the electronic device.”	84
Q. Claim 17	84
R. Claim 18	85
S. Claim 19	85
T. Claim 20	85
IX. NO SECONDARY CONSIDERATIONS OF NONOBVIOUSNESS.....	85
X. CONCLUSION.....	85
CERTIFICATE OF SERVICE	87

List of Exhibits

Exhibit No.	Description of Document
1001	U.S. Patent No. 9,542,571 B2 to Neil P. Adams, et al. (filed May 18, 2015, issued January 10, 2017) (“571” or “571 patent”)
1002	Declaration of Nathaniel Polish, Ph.D. (“Polish”)
1003	Prosecution history of U.S. Patent No. 9,542,571 (Parts 1 and 2)
1004	U.S. Patent Application No. 2002/0099952 A1 (filed June 8, 2001, published July 25, 2002) (“Lambert”)
1005	U.S. Patent Application No. 2002/0095572 A1 (filed January 12, 2001, published July 18, 2002) (“Frank”)
1006	Gomberg, M., Evard, R., and Stacey, C., “A Comparison of Large-Scale Software Installation Methods on NT and UNIX,” <i>Mathematics and Computer Science Division, Argonne National Laboratory, USENIX</i> (1998)
1007	Declaration of Ingrid Hsieh-Yee, Ph.D. (“Hsieh-Yee”)
1008	Argonne National Laboratory webpage, <i>About Argonne</i> , available at https://www.anl.gov/argonne-national-laboratory
1009	Department of Energy webpage, <i>Argonne National Laboratory</i> , available at https://www.energy.gov/ea/argonne-national-laboratory
1010	<i>The Windows Interface Guidelines for Software Design</i> , Microsoft Corporation (1995), pp. xvii-xviii, 272-275 (“Windows Guidelines Book”)
1011	U.S. Patent Application No. 2001/0011341 A1 (filed May 5, 1998, published August 2, 2001) (“Hayes”)
1012	U.S. Patent No. 7,743,407 B2 to Stephen A. Sprigg and Laurence Lundblade (filed August 13, 2002, issued June 22, 2010) (“Sprigg”)
1013	U.S. Patent No. 6,121,970 to Joshua Siegfried Guedalia (filed November 26, 1997, issued September 19, 2000) (“Guedalia”)
1014	Kumar, S. Rajesh and Aravazhi, R., “A Study on Ajax in Web Applications with Latest Trends,” <i>International Journal on Recent and</i>

List of Exhibits

Exhibit No.	Description of Document
	<i>Innovation Trends in Computing and Communication</i> , Vol. 1, Issue 6, pp. 563-568 (2013) (“Kumar”)
1015	Spencer, Donald D., <i>Spencer’s Illustrated Computer Dictionary</i> , Camelot Publishing Co., p. 231 (1995)
1016	K.F. Ibrahim, <i>Electronic Systems & Techniques</i> , 2nd Edition, p. 66 (1994)
1017	<i>Microsoft Computer Dictionary</i> , Microsoft Press, 5th Edition, pp. 94, 196, 254, 440, 445, 446, 572, 575 (2002)
1018	<i>Malikie, et al. v. Nintendo, et al.</i> , No. 2:24-cv-01490-JLR, D.I. 50, LPR 132 Joint Claim Construction and Prehearing Statement (Amended), (W.D. Wash.) (Filed 09/24/2025)
1019	<i>Malikie, et al. v. Nintendo, et al.</i> , No. 2:24-cv-01490-JLR, D.I. 15, Stipulated Motion Extending Time for Defendants to Respond to Complaint (W.D. Wash.) (Filed Oct. 21, 2024)
1020	Table comparing ’571 claim language

I. MANDATORY NOTICES UNDER §42.8(A)(1)

A. Real Party-In-Interest Under §42.8(b)(1)

Nintendo Co., Ltd. and Nintendo of America Inc. (collectively, “Petitioner”) are the real parties-in-interest to this IPR petition.

B. Related Matters Under §42.8(b)(2)

U.S. Patent No. 9,542,571 is the subject of the following pending litigation involving Petitioner: *Malikie Innovations Ltd. v. Nintendo Co.*, Case No. 2:24-cv-01490-JLR (W.D. Wash.). Petitioner was served on October 21, 2024. (EX1019 at 001.)

C. Lead and Back-Up Counsel under §42.8(b)(3)

Petitioner provides the following designation of counsel.

LEAD COUNSEL	BACK-UP COUNSEL
Matthew J. Brigham (Reg. No. 44,047) mbrigham@cooley.com COOLEY LLP ATTN: Patent Group 1299 Pennsylvania Ave. NW, Suite 700 Washington, DC 20004 Tel: (650) 843-5677 Fax: (650) 849-7400	Andrew C. Mace (Reg. No. 63,342) amace@cooley.com COOLEY LLP ATTN: Patent Group 1299 Pennsylvania Ave. NW, Suite 700 Washington D.C. 20004 Tel: (650) 843-5808 Fax: (650) 849-7400
	Patrick. W. Lauppe (Admission <i>pro hac vice</i> to be requested) plauppe@cooley.com COOLEY LLP ATTN: Patent Group 1299 Pennsylvania Ave. NW, Suite 700 Washington D.C. 20004 Tel: (650) 843-5226 Fax: (650) 849-7400

Petition for *Inter Partes* Review of
U.S. Patent No. 9,542,571 B2

LEAD COUNSEL	BACK-UP COUNSEL
	Dena Chen (Admission <i>pro hac vice</i> to be requested) dchen@cooley.com COOLEY LLP ATTN: Patent Group 1299 Pennsylvania Ave. NW, Suite 700 Washington D.C. 20004 Tel: (650) 843-5135 Fax: (650) 849-7400
	Mark R. Weinstein (Admission <i>pro hac vice</i> to be requested) mweinstein@cooley.com COOLEY LLP ATTN: Patent Group 1299 Pennsylvania Ave. NW, Suite 700 Washington D.C. 20004 Tel: (650) 843-5007 Fax: (650) 849-7400
	Stephen Smith (Admission <i>pro hac vice</i> to be requested) stephen.smith@cooley.com COOLEY LLP ATTN: Patent Group 1299 Pennsylvania Ave. NW, Suite 700 Washington D.C. 20004 Tel: (703) 456-8035 Fax: (202) 842-7899

D. Service Information

This Petition is being served by Federal Express to the attorney of record for the '571 patent, 82313 - CONLEY ROSE - BLACKBERRY FILES ATTN: J. ROBERT BROWN, 4965 PRESTON PARK BLVD, SUITE 195E PLANO, TX 75093. This Petition is also being served on litigation counsel identified in the

Petition for *Inter Partes* Review of

U.S. Patent No. 9,542,571 B2

Certificate of Service. Petitioner consents to electronic service at the addresses provided above for lead and back-up counsel.

E. Power of Attorney

Filed concurrently per 37 C.F.R. § 42.10(b).

II. FEE PAYMENT

Petitioner requests review of 20 claims, with a \$51,875 payment.

III. REQUIREMENTS UNDER §§ 42.104 AND 42.108 AND CONSIDERATIONS UNDER §§ 314(A) AND 325(D)

A. Grounds for Standing

Petitioner certifies that the '571 patent is available for IPR and that Petitioner is not barred or estopped.

B. Identification of Challenge and Statement of Precise Relief Requested

Ground	Claims	Basis for Challenge Under § 103
1	1-20	Lambert
2	1-20	Ground 1 + Frank, Gomberg, and Windows Guidelines Book
3	1-8, 10-20	Hayes in view of Sprigg
4	1-8, 10-20	Ground 3 + Frank and Guedalia
5	9-20	Ground 4 + Lambert

This Petition is supported by the Declaration of Nathaniel Polish (EX1002; “Polish”), a qualified expert. (Polish, ¶¶3-9, Appendix A.)

C. Considerations Under §§ 314(a) and 325(d)

Discretionary denial is not warranted. This is the first and only IPR petition filed by anyone against the '571 patent, and the validity or patentability of the '571 patent has not been adjudicated previously. None of the references relied on in Grounds 1-5 were identified during prosecution of the '571 patent. The pending litigation against Petitioner is in a relatively early stage. Petitioner intends to promptly move to stay the litigation pending IPR. Additionally, in the event an IPR is instituted (and the institution is not subsequently vacated, or a Final Written Decision in the IPR is not subsequently vacated or terminated), Petitioner will not pursue in district court any grounds of invalidity against the challenged claims that were, or reasonably could have been, asserted in IPR under §§ 102 or 103 on the basis of prior art patent or printed publications. Petitioner will further address discretionary denial considerations if Patent Owner seeks discretionary denial of the Petition.

IV. '571 PATENT OVERVIEW

A. POSITA

A POSITA would have possessed a Bachelor's degree in software engineering, electrical engineering, or a similar degree, with 2-3 years of practical experience designing computer systems. (Polish, ¶¶15-17.) More education and less practical experience, or vice versa, suffices. (*Id.*, ¶15.)

B. Specification

Titled “System and Method of Owner Application Control of Electronic Devices,” the ’571 patent explains that “corporate computer system administrators may be authorized to perform owner information insertion operations” to control devices. (’571, 6:44-50, Title.) Owner control information can include lists that provide “owner control of application installation and deletion.” (*Id.*, 4:34-56.)

V. CLAIM CONSTRUCTION

No claim construction rulings have issued in the pending litigation between Petitioner and Patent Owner. The parties dispute one term, “list of applications permitted for installation.” (EX1018, p.008.) The parties agreed the claim 1 preamble is limiting. (*Id.*, p.002.)

Petitioner does not believe any term requires express construction at this time because, as shown below, the claims are obvious under any reasonable construction.

VI. PRIOR ART

A. Status

Each relied-upon reference is prior art. Gomberg (EX1006), Windows Guidelines Book (EX1010), Hayes (EX1011), and Guedalia (EX1013) are prior art under at least §102(b). (EX1007, ¶¶24-52 (discussing public availability of Windows Guidelines Book), ¶¶53-82 (discussing public availability of Gomberg).)

Lambert (EX1004), Frank (EX1005), and Sprigg (EX1012) are prior art under at least §102(e).

B. Overview of Grounds

This Petition presents two separate sets of grounds that all challenged claims were obvious: (1) the Lambert-based grounds or (2) the Hayes-based grounds. Either set alone renders the claims unpatentable.

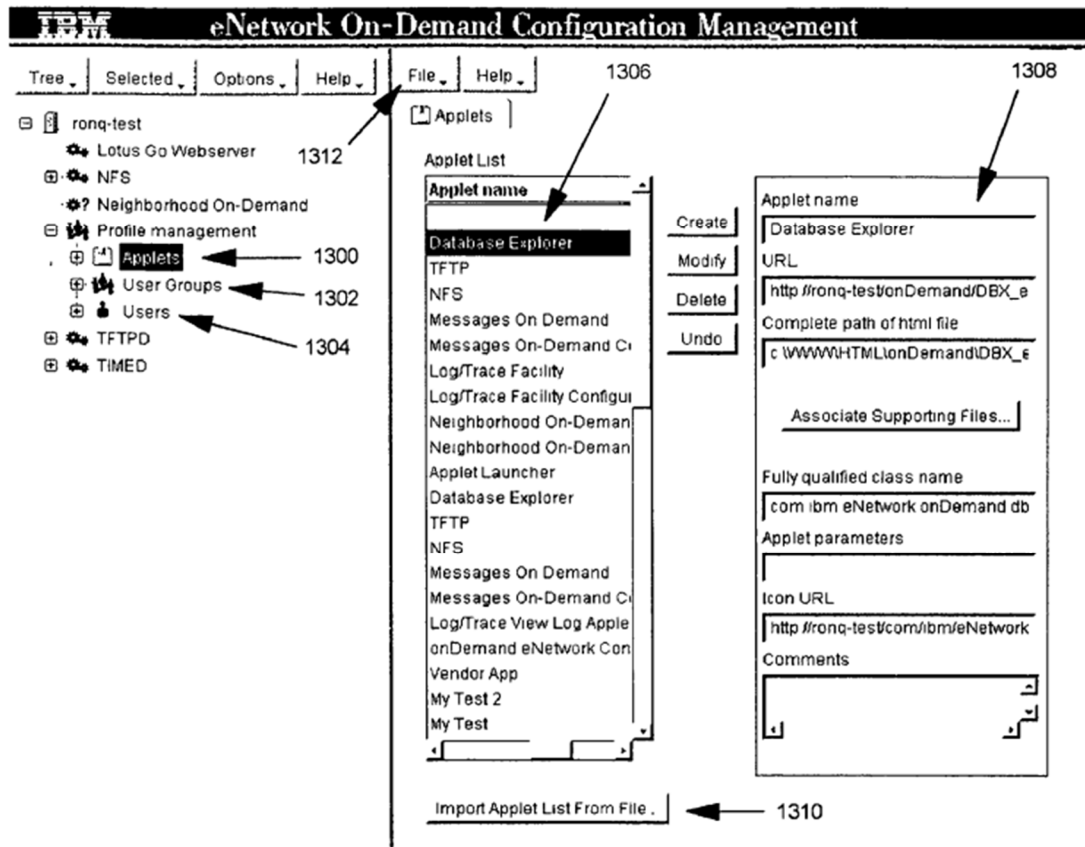
Lambert-Based Grounds

Lambert (EX1004) provides a system for managing applications on a computer using rules. (Lambert, [0052], [0012].) When “software is to be loaded,” rules are “used to determine whether the software can be loaded, and if so, to what extent any processes of the software will be restricted,” including by associating a “restricted access token with the software.” (*Id.*, [0052], [0014], [0070].) For detail on how an administrator could modify the rules on a user device, **Frank** (EX1005) teaches an administrator seeing security policy changes in real time. (Frank, [0040]-[0043].) For additional detail on how installation restrictions could be implemented, **Gomberg** (EX1006) describes providing a list of software for installation. (Gomberg, p.007.) For additional detail about how the rules are protected from modification, the **Windows Guidelines Book** (EX1010) describes using a registry. (Windows Guidelines Book, pp.006-007.)

Hayes-Based Grounds

Hayes (EX1011) provides a system for controlling application access.
(Hayes, [0001].) Administrators determine what applications users can download:

Fig. 13



(*Id.*, Fig. 13.) The server “build[s] a list of applications for which the user has access permission” and downloads the list. (*Id.*, [0013].) For details, Sprigg (EX1012) describes an administrator associating permission lists with applications. (Sprigg, 3:37-47.) Guedalia (EX1013) provides detail for the types of applications that can run in Hayes, including downloading and installing Java applets. (Guedalia, 1:57-

67.) **Frank** (EX1005) describes how an administrator can modify policies in real time (Frank, [0040]-[0043]), and **Lambert** (EX1004) provides implementation details. (Lambert, Fig. 1, [0082], [0063].)

VII. GROUNDS 1-2: CLAIMS 1-20 ARE OBVIOUS OVER LAMBERT (GROUND 1) OR OBVIOUS OVER LAMBERT IN VIEW OF FRANK, GOMBERG, AND WINDOWS GUIDELINES BOOK (GROUND 2)

A. Claim 1:

1. “A method of application control on an electronic device, the method comprising:” (Claim 1[p])

Lambert discloses claim 1[p] (Grounds 1-2). Lambert discloses restricting applications on a computer (“**application control on an electronic device**”). Lambert teaches “the present invention automatically controls whether content is executable, and if so, the extent (if any) to which it is restricted.” (Lambert, [0054],¹ [0101], [0050], [0052], [0022], [0058], [0008]; Polish, ¶¶84, 86.) Lambert discloses “**electronic device[s]**”. (Lambert, [0027] (“personal computers”); Polish, ¶83.) Lambert’s method operates “**on an electronic device**”—software controlling execution of applications on the computer. (Lambert, [0028], [0066], [0031]-[0032], [0038], [0052]; Polish, ¶85.)

¹ All emphasis added to quoted material unless noted.

2. **“receiving, by the electronic device, owner information from an owner of the electronic device;” (Claim 1[a]) and “receiving, by the electronic device, owner control information that comprises:” (Claim 1[b])**

Claims 1[a] and 1[b] are disclosed by or obvious over Lambert (Grounds 1-2). Lambert’s administrator credentials are **“owner information”**—*e.g.*, identifying credentials of an administrator of a corporate network operating for the **“owner”** (a company, like in the ’571). (Lambert, [0012], [0003]; ’571, 5:63-67 (controlled by a “corporate computer system administrator”), 6:44-50, 9:11-17, 11:37-40, 19:14-17, 8:7-10, 10:58-62; Polish, ¶¶88-89.) Lambert’s **“owner information”** (*i.e.*, administrator credentials) is **“from an owner of the electronic device”** (*i.e.*, the company). (Lambert, [0008]-[0009], [0003]-[0006], [0011]-[0012]; Polish, ¶¶88-89.)

Lambert teaches **“owner control information.”** Lambert discloses that an administrator provides a policy—*i.e.*, rules that control application execution on a computer. (Lambert, [0061]-[0065], [0080], [0054], [0101], [0050], [0052] (“administrator can efficiently control software in any environment”), [0022], [0058], [0008], [0056], [0068]-[0071]; Polish, ¶102.) Lambert’s rules constitute **“owner control information”** because they are provided by a network administrator on behalf of a corporate **“owner.”** (Lambert, [0012], [0003] (“system administrators”); ’571, 5:63-67, 6:44-50, 11:37-40, 19:14-17, 9:11-17, 8:7-10, 10:58-62; Polish, ¶103.)

Lambert's "**owner information**" (*i.e.*, administrator credentials) and "**owner control information**" (*i.e.*, administrator policy rules) are "**receiv[ed], by the electronic device.**" (Polish, ¶¶90, 104-105.) Lambert discloses that the user interface for making policy changes can be accessible on any computer in the network "by an administrator." (Lambert, [0081], [0100].) Administrator policy rules can be "set for a user or machine locally and maintained in the local system registry, e.g., under HKEY_LOCAL_MACHINE and HKEY_CURRENT_USER." (*Id.*, [0064], [0078]-[0079], Abstract, [0028].) Lambert discloses that administrator policy rules can be input by an administrator over a network. (*Id.*, [0064] ("distributable over a network"), Abstract, [0008], [0028].) An administrator can further input policy rules from any computer. (*Id.*, [0038], [0081], [0099], [0101], [0010], [0057].) Lambert thus teaches "**receiving, by the electronic device, owner information from an owner of the electronic device**" (*i.e.*, via an administrator logging into a network computer using network credentials) and "**receiving, by the electronic device, owner control information**" (*i.e.*, by the administrator providing policy information for the computer locally or distributed over a network). (Polish, ¶¶90, 104-105.)

Lambert in view of **Frank** (EX1005) also renders obvious "**receiving, by the electronic device, owner information**" (Ground 2). (Polish, ¶¶91-99.) Like Lambert, Frank discloses a computer security system managed by administrators to

control the activity of end users. (Frank, [0002], [0029] (“system administrator... authorized to set and/or change the security profile of the system”), [0005]-[0012].) Frank discloses a security application that an administrator can access on the device and apply security rules input by the administrator in real time. (*Id.*, [0043], [0040]-[0042], [0028]-[0030], [0011]-[0012].) It would have been obvious to implement Lambert’s system to allow an administrator to set-up or adjust the access restriction policy for a user’s computer (locally or on the server) by using the terminal itself. (Polish, ¶93.) In the combination, an administrator could log-in to a user’s terminal by providing “**owner information**” (administrator credentials) to the terminal and retrieve a security application that would allow the administrator to adjust the access restriction policy (stored on a server or in local memory) for the device. (*Id.*) In the combination, the user terminal would “**receiv[e], by the electronic device [i.e., the user terminal], owner information [i.e., administrator credentials] from an owner of the electronic device [i.e., the administrator using the terminal].**” (*Id.*)

Motivation to Combine: Lambert and Frank are analogous references to the ’571 patent. (Polish, ¶99.) Like the ’571, both Lambert and Frank relate to computer security policies by system administrators. (Frank, [0002], [0029], [0005]-[0012], [0036]; Lambert, [0078], [0002], [0080]-[0081], [0003], [0007]; ’571, 1:41-43, 6:44-50, 1:52-55, 1:63-2:5; Polish, ¶¶92, 99.) A POSITA would have had more than a reasonable expectation of success and could have implemented the proposed

combination using conventional techniques. (Polish, ¶99.) A POSITA would have been motivated to implement a security application on a user terminal on Lambert’s system to enable an administrator to easily see the results of changes to the security policy for several advantages. (*Id.*, ¶95.) Lambert discloses that a security policy might create “an error wherein the default rule is set to prevent (revoke) all software from running, which could then otherwise prevent the user interface program 600 from subsequently running to except itself from the default rule.” (Lambert, [0081].) Frank discloses that its system helps avoid such an error—*i.e.*, setting a policy that locks “even the system administrator out of the computer system and/or the security application[.]” (Frank, [0012], [0011], [0036].) A POSITA would have been motivated to implement a security application like Frank’s on a user terminal to make access restriction policy errors less likely. (Lambert, [0081], [0058]; Polish, ¶95.) A POSITA would have been further motivated to implement an administrator security application on a user terminal in Lambert’s system because formulating security policy for a network requires an iterative balancing approach. (Frank, [0008]; Polish, ¶97.) A POSITA would have further been motivated to combine Lambert and Frank to streamline the process of inputting security policies for a computer to decrease time and cost. (Lambert, [0007]; Polish, ¶96.)

(a) “a list of applications permitted for installation on the electronic device, and” (Claim 1[b][i])

Lambert renders claim 1[b][i] obvious (Grounds 1-2). The policy in

Lambert’s system is “**a list of applications permitted for installation on the electronic device,**” because the policy lists applications (like Internet Explorer or Windows Media Player) that can be installed “**on**” the electronic device. (Lambert, [0072], [0064]-[0065], [0080], [0068]-[0071]; Polish, ¶108.) As an example:

OnlySignedLockDown	HKCU	Disallowed	CP only
Path	%windir%	Unrestricted	All file types
Certificate	<publisher cert>	Unrestricted	
IexploreOnlyLockDown	HKCU	Disallowed	CP only
Path	%windir%	Unrestricted	All file types
Path	%programfiles%\Internet Explorer\	Unrestricted	All file types
WMPOnlyLockDown	HKCU	Disallowed	CP Only
Path	%windir%	Unrestricted	All file types
Path	%programfiles%\Windows Media Player	Unrestricted	All file types
PFNoMediaLockDown	HKCU	Disallowed	CP Only

(Lambert, [0082] (excerpt); Polish, ¶108.) Lambert discloses that the policy can restrict what is “**permitted for installation**”—*e.g.*, “software installer functions.” (Lambert, [0082]; Polish, ¶109.) Lambert’s policy is provided by an administrator who is acting on behalf of the “**owner**” (*e.g.*, the corporation). Thus, the list of applications in the software policy is “**permitted**” by the owner, *i.e.*, the administrator. (Polish, ¶110.) Lambert’s rules are a “**list of applications**” permitted by the owner “**for installation.**”

Alternatively, claim 1[b][i] is obvious over Lambert in view of **Gomberg** (EX1006) (Ground 2). Gomberg discloses that “the user can be presented with a list of software that can be installed at login.” (Gomberg, p.007; Polish, ¶113.) Lambert in view of Gomberg renders obvious the claimed “**list of applications permitted for installation on the electronic device.**” It would have been obvious in light of Gomberg to implement Lambert to provide users with a list of software permitted for installation at login. (Polish, ¶118.)

Motivation to Combine: Gomberg, like Lambert, is analogous to the ’571 patent because all are in the field of administered computer networks and using software. (Gomberg, pp.007, 002; Lambert, [0078], [0002], [0080]-[0081], [0003]; ’571, 1:52-55, 1:63-2:5; Polish, ¶115.) Both use the Windows operating system. (Lambert, [0037], [0038], [0082]; Gomberg, p.002.) A POSITA would have had more than a reasonable expectation of success and could have implemented the proposed combination using conventional techniques. (Polish, ¶119.) A POSITA would have been motivated to combine Lambert and Gomberg to “help[] with the problem of buying licenses that you don’t need.” (Gomberg, p.007.) A POSITA would have understood Gomberg’s licensing concerns to be relevant to Lambert’s corporate network. (Lambert, [0012], [0003]; Polish, ¶114.) Lambert discloses that disadvantages of the prior art include “increased support costs in system maintenance and helpdesk time[.]” (Lambert, [0007].) A POSITA would have

Petition for *Inter Partes* Review of
U.S. Patent No. 9,542,571 B2

considered licensing costs a cost to minimize. (Polish, ¶114.) A POSITA would have been motivated to combine Lambert and Gomberg to make Lambert’s policies (*e.g.*, installation restrictions) more user-friendly. (Lambert, [0072]; Polish, ¶116.) A POSITA would also have been motivated to look to Gomberg for detail on how to implement Lambert’s restriction of “software installer functions.” (Polish, ¶117; Gomberg, p.007.)

(b) “permissions associated with each permitted application in the list of applications;” (Claim 1[b][ii])

Lambert teaches or renders obvious claim 1[b][ii] in several ways (Grounds 1-2).

Unrestricted / Disallowed

Different applications are “Unrestricted” or “Disallowed”:

OnlySignedLockDown Path	HKCU %windir%	Disallowed Unrestricted	CP only All file types
Certificate IexploreOn- lyLockDown Path	<publisher cert> HKCU %windir%	Unrestricted Disallowed	CP only
Path	%programfiles%\Internet Explorer\	Unrestricted	All file types
WMPOnlyLockDown Path	HKCU %windir%	Disallowed Unrestricted	CP Only All file types
Path	%programfiles%\ Windows Media Player	Unrestricted	All file types
PFNoMediaLockDown	HKCU	Disallowed	CP Only

(Lambert, [0082] (excerpt, color added).) These are “**permissions**” associated with each application in the “**list of applications.**” (Polish, ¶123; Lambert, [0006] (“determine each user’s access to network resources based on permissions granted according to that user’s credentials”).) Each row is an item in the “**list.**” (Polish, ¶¶123-124.) “[S]ecurity levels for execution... may include normal (unrestricted), constrained (restricted) or disallowed (revoked)....” (Lambert, [0013], [0061].) Lambert’s rules monitor processes on the computer and provides an instruction on whether a given process can execute, or whether it can execute under certain restrictions. (*Id.*, [0068]-[0071], [0080]-[0082], Abstract, [0006], [0099], [0008]-

[0009], [0011], [0013], [0054], [0056], [0064]-[0065], [0072].) Lambert’s policy thus teaches “**permissions associated with each permitted application in the list of applications.**” (Polish, ¶124.)

Tokens

Lambert teaches that the enforcement mechanism can be instructed based on the administrator policy to restrict the activity of applications by providing a “restricted token.” (Lambert, [0070]-[0071]; Polish, ¶125.) Enforcement mechanism **518** “associate[s] a restricted token” with a software application. (Lambert, [0070].) The restricted token then “establishes a restricted execution environment/context for the processes” of the application, “thus providing the rule-determined secure execution environment.” (*Id.*, [0071].) The restricted token can limit the types of processes the application can do, including whether it has “read-write access to a file object.” (*Id.*, [0072], [0042]; Polish, ¶125.)

Lambert’s tokens are thus “**permissions associated with each permitted application in the list of applications.**” (Polish, ¶126.) The tokens—either restricted or unrestricted “parent tokens”—are associated with software applications. (Lambert, [0042], [0047], [0048], [0066], [0070]-[0073].) The tokens also provide “**permissions**” because they indicate what processes can and cannot execute. (*Id.*, [0043], [0047], [0071]-[0073], [0013]-[0014], [0037], [0042]; Polish, ¶126.)

3. **“storing, by the electronic device, the owner control information in a protected data store to prevent alteration or deletion of the owner control information in response to authenticating the owner control information using the owner information;” (Claim 1[c])**

Claim 1[c] is disclosed by or obvious over Lambert (Grounds 1-2). Lambert discloses **“storing, by the electronic device, the owner control information.”** Lambert discloses storing application restriction policy information on a user’s computer, “locally maintained and/or in a group policy object distributable over a network.” (Lambert, Abstract, [0056], [0079], [0064], [0008], [0055]; Polish, ¶¶127-129.)

Lambert further discloses storing the owner control information **“in a protected data store to prevent alteration or deletion of the owner control information in response to authenticating the owner control information using the owner information.”** As explained above, Lambert discloses that a user interface for adjusting policy settings can be accessed *only* with administrator privileges. (Lambert, [0081], [0078], [0055], [0056]; Polish, ¶130.) Lambert’s system is hierarchical, with policies from administrators overriding other (*e.g.*, user-set) policies. (Lambert, [0007], [0057], [0100], [0099], [0039], [0080].) Thus, Lambert discloses **“storing, by the electronic device, the owner control information in a protected data store [*i.e.*, one that can only be accessed or altered using administrative credentials] to prevent alteration or deletion of the owner**

control information in response to authenticating the owner control information using the owner information.” (Polish, ¶¶130-131.) Lambert’s teachings are consistent with the ’571 patent—*e.g.*, the “protected memory areas” where only the administrator-accessible “insertion module 44 has write and erase access[.]” (’571, 10:58-62, 13:46-62, 9:34-36; Polish, ¶132.)

Lambert also teaches “**prevent[ing] alteration or deletion**” to policy settings stored locally. (Polish, ¶¶131, 133.) Lambert teaches that policy information (“**owner control information**”) can be stored in a user device’s “registry,” which is a central depository for storing configuration information, including in Windows-based devices. (Lambert, [0064] (“maintained in the local system registry, *e.g.*, under HKEY_LOCAL_MACHINE and HKEY_CURRENT_USER”), claim 37; EX1017, pp.0011-0012 (defining “registry”), p.009 (“HKEY”); Polish, ¶133.) The information in the registry can be edited using the “registry editor,” also known as “REGEDIT.” (EX1017, p.0012.)

The registry is a “**protected data store.**” Lambert’s registry editor provides a controlled way to access and edit registry data. (Polish, ¶133.) Use of the registry editor can be restricted, thereby preventing access and protecting its contents. (*Id.*; Lambert, [0062], [0082] (“Disallowed” for the path “%WINDIR%\system32\regedit.exe”).) By preventing the registry editor (REGEDIT / regedit.exe) from running, alteration or deletion of the “**owner control**

information” is prevented based on authenticating the administrator’s credentials—
thereby disclosing **“storing, by the electronic device, the owner control
information in a protected data store to prevent alteration or deletion of the
owner control information in response to authenticating the owner control
information using the owner information.”** (Polish, ¶133.)

Claim 1[c] also is obvious over Lambert in view of the **Windows Guidelines
Book** (EX1010; Ground 2). (Polish, ¶¶134-140.) The Windows Guidelines Book
discloses that the HKEY_LOCAL_MACHINE register disclosed by Lambert can be
used to “add your service to the Passwords object[.]” (Windows Guidelines Book,
p.007.) The Book discloses the following structure:

```
HKEY_LOCAL_MACHINE
  System
    CurrentControlSet
      Control
        PwdProvider
          Provider Name Value Name = Value
```

(*Id.*) Adding a service to the Passwords object leads to that service being password-
protected, enabling the user to “change the password for the service by selecting the
name and filling in the resulting dialog box.” (*Id.*, pp.006-007.) It thus would have
been obvious for an administrator to store Lambert’s restriction policy rules in the
Passwords object of HKEY_LOCAL_MACHINE to prevent alteration or deletion
of the policy rule information by anyone other than an administrator. (Polish, ¶136.)

The combined system would “**stor[e], by the electronic device, the owner control information** [*i.e.*, restriction policy rules] **in a protected data store** [*i.e.*, the Passwords object in HKEY_LOCAL_MACHINE] **to prevent alteration or deletion of the owner control information in response to authenticating the owner control information using the owner information** [*i.e.*, administrator credentials].” (*Id.*)

Motivation to Combine: The Windows Guidelines Book, like Lambert, is analogous to the '571 patent, since both relate to user access restrictions and preventing users from circumventing such restrictions, including with Microsoft software. ('571, 1:41-43, 2:54-61, 1:67-2:4; Windows Guidelines Book, pp.006-007, 003; Lambert, [0037]-[0039], [0082], [0002], [0006], [0048], [0100]; Polish, ¶¶137-138.) A POSITA would have been motivated to combine Lambert with the Windows Guidelines Book for details on implementing Lambert's restriction policy rules that can be changed only by administrators. (Polish, ¶139.) A POSITA would have had more than a reasonable expectation of success and could have implemented the proposed combination using conventional techniques. (*Id.*, ¶140.) Lambert discloses that its restriction policy rules are stored on user terminals and can only be modified by certain users—*e.g.*, administrators. (Lambert, Abstract, [0055]-[0057], [0079], [0064], [0008], [0100], [0080], [0099], [0006].) The Windows Guidelines Book provides a design for storing restriction policy rules

on a user computer while restricting their modification—*i.e.*, password-protecting information stored in the HKEY_LOCAL_MACHINE register. (Polish, ¶¶139-140.)

4. **“receiving, by the electronic device, from an application operating on the electronic device, a request to access local memory on the electronic device;” (Claim 1[d])**

Claim 1[d] is disclosed by Lambert (Grounds 1-2). Lambert discloses receiving “**request[s] to access the local memory**” from applications running “**on an electronic device.**” Lambert teaches applications “requesting that an attachment be opened;” requesting “read/write access to a file object;” and requesting permission for “a create process function, a load library..., shell execute function, an object creation function (e.g., CoCreateInstance), script running functions and software installer functions.” (Lambert, [0010], [0042], [0072], [0014], [0038], [0056], [0043], [0046], [0048], [0066], [0070], [0076]; Polish, ¶142.) These are requests to “**access local memory,**” since they involve accessing items stored in the local memory. (*Id.*, [0082], [0063], Fig. 1; Polish, ¶142.) Lambert discloses that application operation involves accessing local random access memory. (*Id.*, [0031].)

Figure 5A

Lambert discloses receiving requests to use computer resources—including opening, loading, or executing an application **510** located in memory. (Lambert,



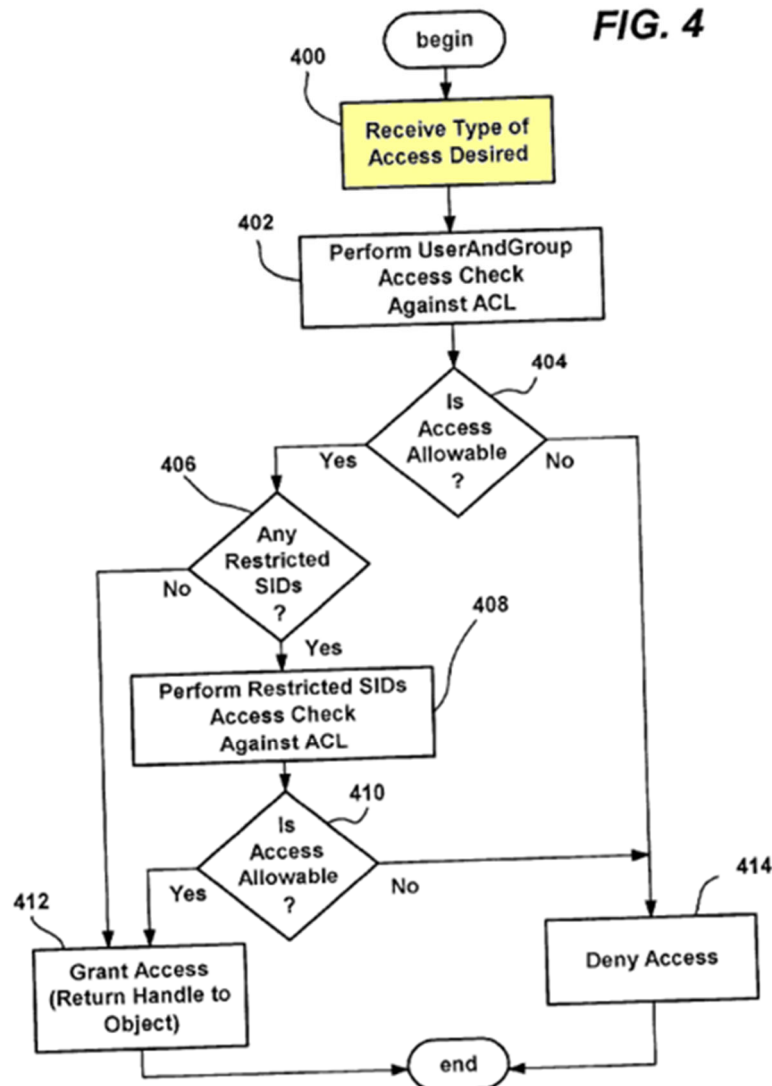
-23-

software functions, including loading or executing software via the local memory].”

(Polish, ¶143.)

Restricted Token

Lambert also discloses that the enforcement mechanism discussed above can associate a restricted access token with an application. (Lambert, [0070], [0071], [0074], [0100], [0013]-[0014], Figs. 2, 3, 9, [0037]-[0043].) When the application seeks to execute a process (*e.g.*, gain read or write access to another file), the application may “**request to access local memory on the electronic device**”—*e.g.*, comparing permissions in the restricted token against permissions associated with a target file. (*Id.*, [0048], [0042]; Polish, ¶144.) Lambert discloses that step **400** in Figure 4 represents “requesting access to the object **302**,” which can be a “file object” in local memory. (Lambert, [0048], [0042].)



(*Id.*, Fig. 4 (highlighting added); Polish, ¶144.) Step 400 constitutes “receiving, by the electronic device, from an application operating on the electronic device [*i.e.*, an application associated with a restricted token], a request to access local memory on the electronic device [*e.g.*, a request for read/write access to a file object].” (Polish, ¶144.)

5. “determining whether the application is allowed to access the local memory in response to identifying a permission associated with the application in the owner control information; and” (Claim 1[e])

Claim 1[e] is disclosed by or obvious over Lambert (Grounds 1-2). As explained above, Lambert teaches the claimed “**owner control information**” and “**permission associated with the application in the owner control information.**” (Polish, ¶¶145, 102-103, 122-126.) Lambert discloses determining application access based on the policy rules discussed above. Lambert teaches that “software that possibly includes executable code is identified and classified, and then a rule corresponding to the classification (or if none correspond, a default rule) automatically and transparently determines the software’s ability to execute.” (Lambert, [0008], [0052], [0067], [0013], [0014]; Polish, ¶146.) Since Lambert discloses applications can be stored and executed via local memory, this “**determining**” process represents whether “**access [to] the local memory**” is allowed. (*Id.*, [0082] (showing applications in local “windir” and “programfiles” directories), [0031] (“RAM **132** typically contains data and/or program modules that are ... presently being operated on by processing unit **120**.”), [0063], Fig. 1; Polish, ¶146.)

Figure 5A

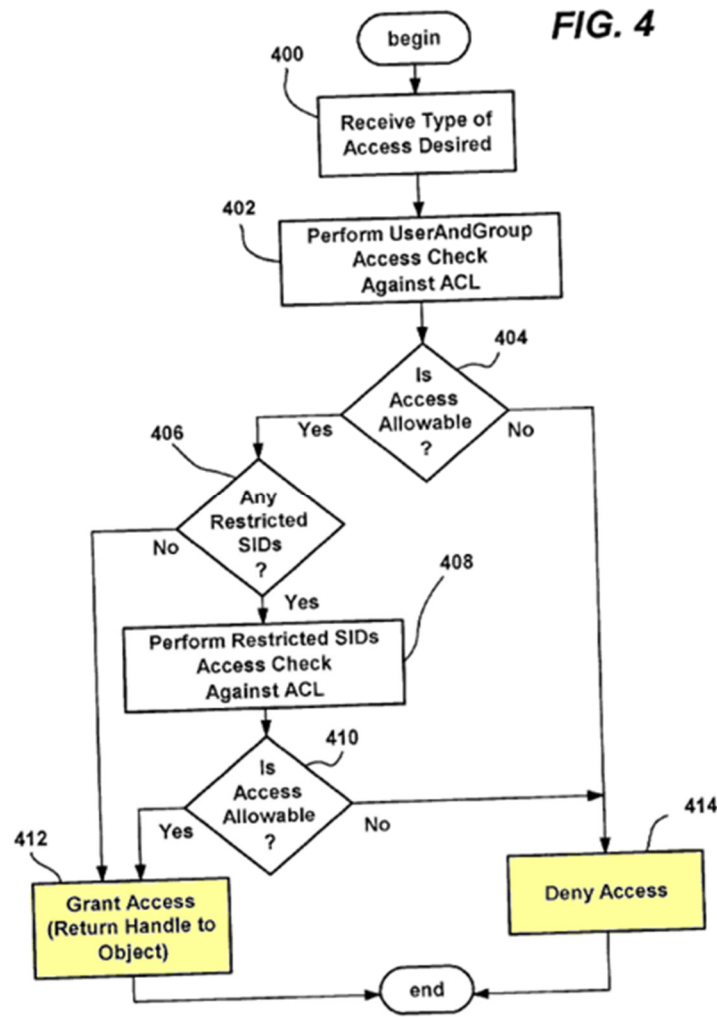
Lambert discloses that an access request from an application **510** leads to a query to an “enforcement mechanism **518**,” which “obtains a security level



-27-

Restricted Token

Lambert discloses that its enforcement mechanism can associate a restricted access token with an application. (Polish, ¶149.) When the application then requests access to local memory (*e.g.*, read/write access to a file object), Lambert's system determines if such access is allowed based on the restricted token (Step 404). (Lambert, [0070], [0071], [0074], [0100], [0013]-[0014], Figs. 2, 3, 9, [0037]-[0043].)



(*Id.*, Fig. 4 (color added), [0049], [0048]; Polish, ¶149.) Lambert thus discloses

“determining whether the application is allowed to access the local memory [e.g., read/write access to a file object in the local memory] in response to identifying a permission associated with the application in the owner control information [e.g., a restricted access token derived by the enforcement mechanism from administrator policy rules].” (Polish, ¶¶149-150.)

6. “controlling application access to the local memory based upon the determining.” (Claim 1[f])

Claim 1[f] is disclosed by or obvious over Lambert (Grounds 1-2). As explained above, Lambert teaches **“controlling application access.”** (Lambert, [0054], [0101], [0050], [0052], [0022], [0058], [0008]; Polish, ¶152.) Since Lambert discloses applications can be stored and executed via local memory, this **“controlling”** process represents controlling whether **“access to the local memory”** is allowed. (*Id.*, [0082] (showing applications in local “windir” and “programfiles” directories), [0031] (“RAM 132 typically contains data and/or program modules that are ... presently being operated on by processing unit 120.”), [0063], Fig. 1; Polish, ¶152.)

Figure 5A

Lambert discloses, based on the above-discussed determining step in Figure 5A, its system **“control[s] application access to the local memory based upon the determining.”** (Polish, ¶¶84-86.) In step (9), the enforcement mechanism

518 “completes the call and returns information back to the software function **516** that called it, the returned information essentially including an instruction on whether to open the file, and if so, the token to associate with the file.” (Lambert, [0071].) In step (10), “[t]he returned token... is maintained in a set of process information **528** (e.g., in a protected location) associated with that file[.]” (*Id.*)

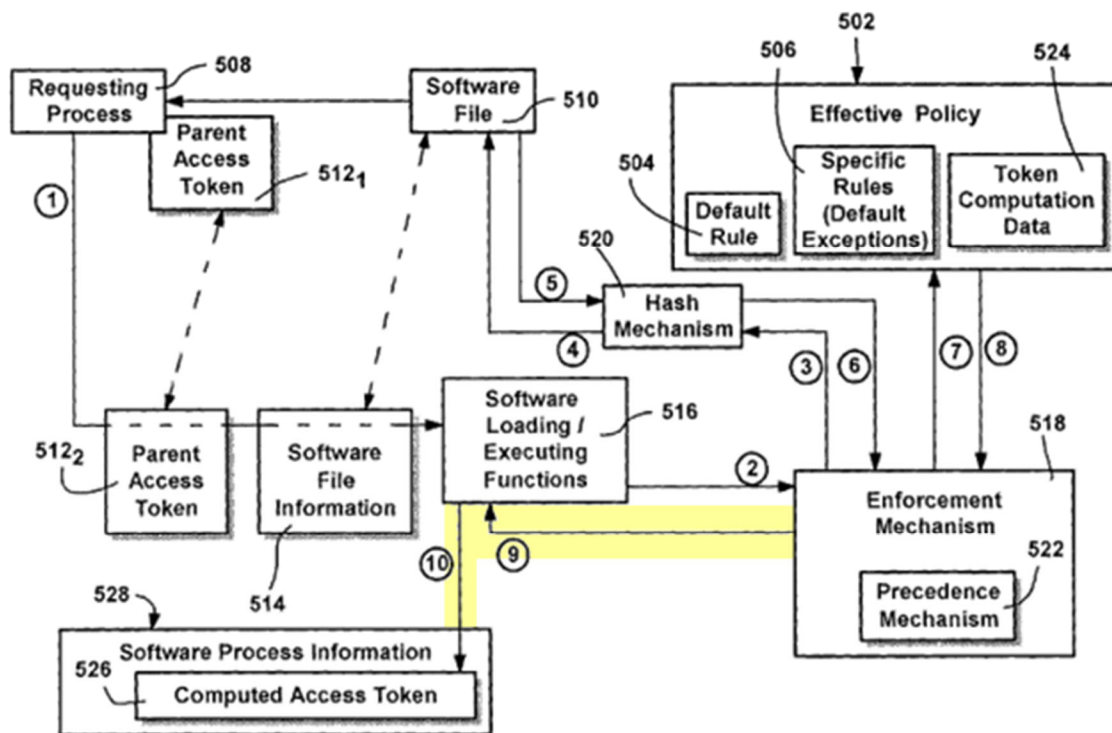
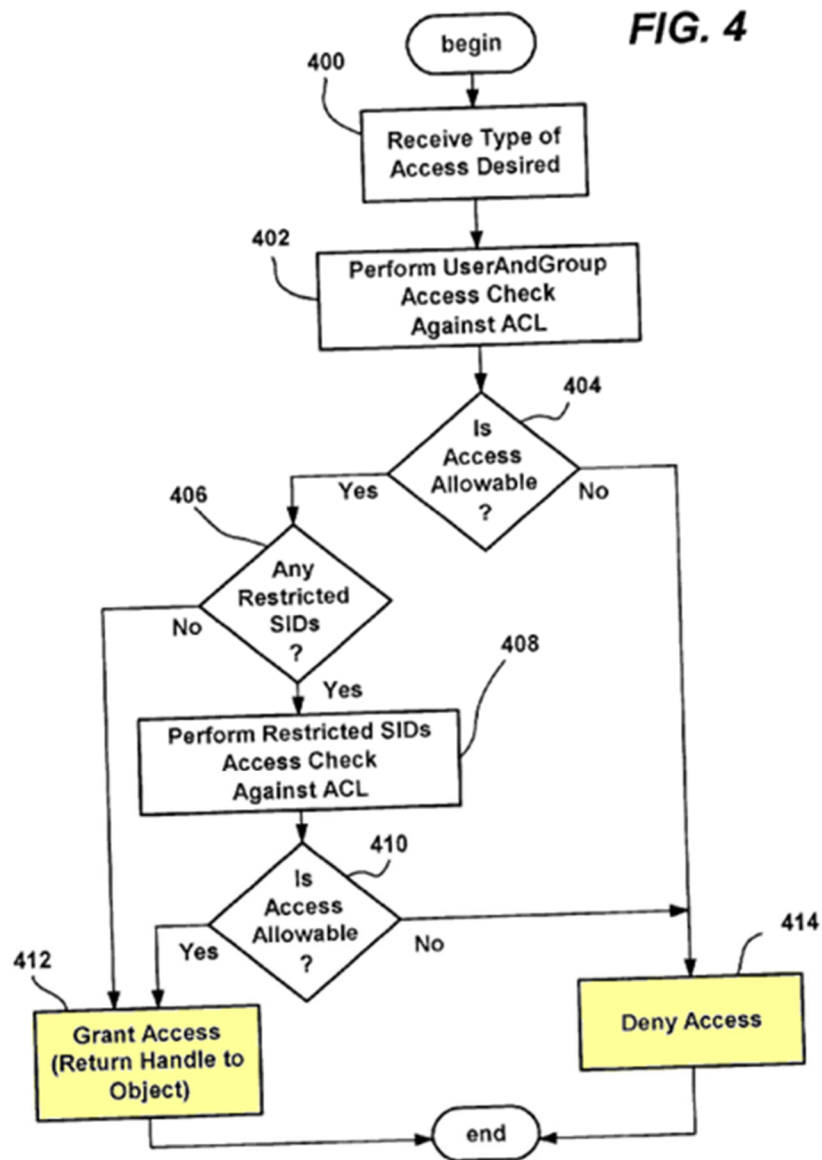


FIG. 5A

(*Id.*, Fig. 5A (highlighting added), [0071]; Polish, ¶153.) Lambert thus teaches or renders obvious “**controlling application access to the local memory based upon the determining**” step—e.g., controlling whether an application can run (e.g., an application stored and executing via local memory) and, if so, in what restricted capacity. (Polish, ¶153.)

Restricted Token

Lambert discloses controlling application access to the local memory based on the restricted token associated with the application. (Lambert, [0071].) Lambert discloses that based on the determining step discussed above related to the restricted token control process, access to a target object (*e.g.*, read/write access to a file object) is either granted or denied. (*Id.*, [0043]-[0049].)



(*Id.*, Fig. 4 (color added); Polish, ¶154.) Thus, Lambert discloses a further level at which its system “**control[s] application access to the local memory [e.g., read/write access to a file stored in local memory] based upon the determining.**” (Polish, ¶¶154-155.)

B. Claim 2: “The method of claim 1, wherein the request to access the local memory comprises a request to erase data from the local memory.”

Lambert renders claim 2 obvious (Grounds 1-2). As discussed for claim 1[d], the “**request to access the local memory**” in Lambert includes a request by an application running on the device to access an object in local memory, such as a file. (Lambert, [0042], [0066]; Polish, ¶159.) With respect to a request “**compris[ing] a request to erase data from the local memory,**” Lambert discloses the request by the application can include seeking to “obtain read/write access” to the file. (Lambert, [0042]-[0043], [0061], [0014], [0065], claim 39; Polish, ¶¶160-161.) A request for write access would be “**a request to erase data,**” because the ability to write data includes the ability to erase data, since writing (or overwriting) data necessarily involves erasing previously stored data. (Polish, ¶¶160-61; EX1017, pp.008, 0014 (“write mode”), 0010 (“read-only”).)

C. Claim 3: “The method of claim 1, wherein the request to access the local memory comprises a request to read data from the local memory.”

Lambert renders claim 3 obvious (Grounds 1-2). As discussed for claims 1[d] and 2, Lambert discloses that its system can receive a request from a process “desir[ing] access to an object **302,**” including “the type of access it desires (e.g., obtain read/write access to a file object)[.]” (Lambert, [0042], [0043].) Because Lambert discloses that an application can request “read” access for a file, Lambert

discloses a request “**to access the local memory comprises a request to read data from the local memory**”—*e.g.*, to read a file stored in local memory. (Polish, ¶¶164-166; EX1017, pp.0010, 0014.)

D. Claim 4: “The method of claim 1, further comprising restricting modification of the owner control information after the owner control information has been stored in the protected data store.”

Lambert (Grounds 1-2) and/or Windows Guidelines Book (Ground 2) renders claim 4 obvious. The claimed “**owner control information**” of Lambert and/or Windows Guidelines Book is discussed for claim 1[c]. (Polish, ¶¶169, 102-103.)

Lambert discloses a user interface program for adjusting policy rules that can only be accessed with administrator privileges, thereby “**restricting modification.**” (Lambert, [0081], [0078], [0039], [0057], [0100], [0080], [0099], [0064]; Polish, ¶169.) Lambert discloses that locally stored policy settings could be stored in the “registry,” which constitutes a “**protected data store**” for which modification can be “**restrict[ed].**” (Lambert, [0064], claim 37, [0062], [0082]; EX1017, pp.0011-0012, 009; Polish, ¶169.)

It would have been obvious to store policy information (“**owner control information**”) in the HKEY_LOCAL_MACHINE register and password protect it such that only an administrator could access or change it, including based on the Windows Guidelines Book. (Polish, ¶170.) Thus, “**restricting modification** [*i.e.*, preventing non-administrator alteration] **of the owner control information after**

the owner control information has been stored in the protected data store” is obvious for the reasons explained above. (*Id.*)

E. Claim 5: “The method of claim 1, wherein the owner control information includes an authorization record and an application identifier for each of the applications, and the owner control information is provided by an external source through a wireless communication channel.”

Lambert renders claim 5 obvious (Grounds 1-2). Lambert teaches example policy rules:

OnlySignedLockDown Path	HKCU %windir%	Disallowed Unrestricted	CP only All file types
Certificate IexploreOn- lyLockDown Path	<publisher cert> HKCU %windir%	Unrestricted Disallowed	CP only
Path	%programfiles%\Internet Explorer\	Unrestricted	All file types
WMPOnlyLockDown Path	HKCU %windir%	Disallowed Unrestricted	CP Only All file types
Path	%programfiles%\ Windows Media Player	Unrestricted	All file types
PFNoMediaLockDown	HKCU	Disallowed	CP Only

(Lambert, [0082] (excerpt).) Each associated rule for an application (*e.g.*, “Disallowed,” “Unrestricted,” “All file types,” and “CP only”) is an “**authorization record**” because it specifies a level of authorization (*e.g.*, permitted tasks) associated with “**each**” application. (*Id.*, [0079], [0053]-[0054], [0058]-

[0063], [0068]; Polish, ¶173.) The associated path or hash for the application is an “**application identifier**” because it identifies the application. (Lambert, [0053], [0058], [0056], [0055], [0082]; Polish, ¶¶173-174.)

Lambert discloses that the “**authorization record**” (“**owner control information includes an authorization record**”) is “**provided by an external source.**” (Polish, ¶¶175, 101-106.) Lambert discloses that restriction policy information can be provided “within one or more group policy objects distributable over a network.” (Lambert, [0064], [0012], [0008]; ’571, 18:57-61 (“maintained remotely or on an external memory device”).)

Lambert’s system provides policy information (“**owner control information**”) “**through a wireless communication channel.**” Lambert discloses computers connected to a “local area network (LAN) 171 and a wide area network (WAN) 173[.]” (Lambert, [0034], [0035]; EX1017, pp.0013 (“wireless LAN”), 007 (“channel”); Polish, ¶¶176, 101-106.) The devices communicate via “wireless media such as acoustic, RF, infrared and other wireless media.” (Lambert, [0030].) Lambert’s policy information (“**owner control information**”) is provided from a remote/external source “**through a wireless communication channel.**” (Polish, ¶176.)

F. Claim 6: “The method of claim 5, wherein the authorization record related to determining whether access to the local memory is allowed is provided by an external computer that is used to enforce policies for operating electronic devices within an organization.”

Lambert renders claim 6 obvious (Grounds 1-2). As discussed for claim 5, Lambert discloses providing restriction policy rules “within one or more group policy objects distributable over a network”—*i.e.*, “**provided by an external computer.**” (Lambert, [0064], [0008], [0012]; Polish, ¶179.) Lambert discloses that the “**authorization record[s]**” in policy information (“**owner control information**”) are “**related to determining whether access to the local memory is allowed.**” As discussed for claim 1[e], the rules indicate whether a software can execute at all or what processes applications can execute. (Lambert, [0082], [0079], [0072], [0042], [0043], Figs. 1, 3; Polish, ¶180.)

Lambert discloses that the external computer is “**used to enforce policies for operating electronic devices within an organization**”—*i.e.*, a business. Lambert’s system is used “[i]n an enterprise environment” to set an access policy “for sites, domains, organizational units, groups and/or users via group policy technology.” (Lambert, [0012], [0003], [0007]; Polish, ¶181.)

G. Claim 7: “The method of claim 1, wherein an update to the owner control information is provided by an external source for storage on the electronic device.”

Lambert renders claim 7 obvious (Grounds 1-2). As discussed for claims 5 and 6, Lambert discloses providing policy information (“**owner control**

information”) from an external source. (Lambert, [0064], [0008], [0012]; Polish, ¶184.) Lambert further discloses that its system can “automatically generate and propagate rules... by dynamically accessing a database to find new software that has been identified as having a virus, and dynamically generating and distributing new policy including a rule that prevents that software from running.” (Lambert, [0012], [0056].) This “distributing” of new policy information indicates local storage on user devices. (*Id.*, [0004] (“distributed to various computers”), [0056], Abstract; Polish, ¶185.) A POSITA would have understood that the generation and distribution of new rules constitutes providing “**an update to the owner control information is provided by an external source for storage on the electronic device[.]**” (Polish, ¶185.)

H. Claim 8

1. **“The method of claim 1, wherein a plurality of authorization records associated with a plurality of application identifiers are stored on the electronic device, the method further comprising:” (Claim 8[p])**

To the extent limiting, claim 8[p] is disclosed by or obvious over Lambert (Grounds 1-2). With respect to “**a plurality of authorization records associated with a plurality of application identifiers,**” this limitation is disclosed by Lambert as discussed for claim 5. (Polish, ¶187.) Lambert discloses that policy information includes entries for applications that identify the application (such as by file path or hash) (“**application identifier**”) and have an associated rule (“**authorization**

record”). (Lambert, [0082], [0079], [0053]-[0056], [0058]-[0063], [0068].)

Lambert also discloses the policy information is **“stored on the electronic device.”** (Lambert, Abstract, [0064]; Polish, ¶188.)

2. “receiving requests to access the local memory from a plurality of applications operating on the electronic device;” (Claim 8[a])

As discussed for claim 1[d], this element is disclosed by or obvious over Lambert (Grounds 1-2). (Polish, ¶¶189-190, 141-144.) Claim 8[a] additionally requires receiving requests from a **“plurality of applications,”** which Lambert discloses—*e.g.*, the applications identified in the policy rules, or the various applications in the zone of a “zone rule” or at the filepath of a “path rule.” (Lambert, [0082], [0052], [0058]-[0063]; Polish, ¶190.)

3. “determining whether access to the local memory is allowed to be performed by their respective applications based upon the stored authorization records and the application identifiers that are respectively associated with the plurality of applications; and:” (Claim 8[b])

As explained for claims 1[e] and 6, Lambert discloses determining local memory access based on policy rules, which (as discussed for claim 8[p]) constitute **“stored authorization records and the application identifiers that are respectively associated with the plurality of applications.”** Thus, claim 8[b] is disclosed by or obvious in light of Lambert (Grounds 1-2). (Polish, ¶¶191-192, 145-150.)

4. “controlling access by the plurality of applications to local memory based upon whether access to the local memory is allowed.” (Claim 8[c])

Claim 8[c] is disclosed by or obvious over Lambert (Grounds 1-2). Claim 8[c] additionally requires controlling access by a **“plurality of applications.”** As discussed for claim 1[f], Lambert discloses that its system controls local memory access for various applications (**“a plurality of applications”**), including the applications identified in the policy rules, or the various applications in the zone of a “zone rule” or at the filepath of a “path rule.” (Lambert, [0082], [0052], [0058]-[0063]; Polish, ¶¶193-194.)

I. Claim 9: “The method of claim 8, wherein a group containing two or more of the applications is associated with one of the authorization records.”

Claim 9 is disclosed by or obvious over Lambert (Grounds 1-2). Lambert discloses that an access restriction can be associated with a filepath, such that any applications at that filepath have the same restrictions. (Lambert, [0058]-[0062], [0082], [0068], [0066].) A filepath-based access restriction can therefore comprise **“a group containing two or more of the applications”**—such as a folder containing two or more applications—associated with that restriction (**“one of the authorization records”**). (Polish, ¶197.)

Lambert discloses that an access restriction can be set using a “zone rule,” wherein “the zone identifies software according to the source (e.g., the URL zone or

the like) from which it is downloaded or otherwise installed.” (Lambert, [0063].)

Such “zone rules” represent another way Lambert discloses “**a group containing two or more of the applications**” associated with a single restriction level (“**with one of the authorization records**”), so that “an administrator can set different rules for software files classified according to their source.” (*Id.*; Polish, ¶198.)

Lambert also discloses an example access restriction policy in which multiple listings are associated with the same restriction level. (Lambert, [0082].) The below shows that multiple entries are associated with the same “Disallowed” or “Unrestricted” level.

OnlySignedLockDown	HKCU	Disallowed	CP only
Path	%windir%	Unrestricted	All file types
Certificate	<publisher cert>	Unrestricted	
IexploreOn-lyLockDown	HKCU	Disallowed	CP only
Path	%windir%	Unrestricted	All file types
Path	%programfiles%\Internet Explorer\	Unrestricted	All file types
WMPOnlyLockDown	HKCU	Disallowed	CP Only
Path	%windir%	Unrestricted	All file types
Path	%programfiles%\Windows Media Player	Unrestricted	All file types
PFNoMediaLockDown	HKCU	Disallowed	CP Only
Path	%windir%	Unrestricted	All file types
Path	%programfiles%\	Unrestricted	All file types
Path	%programfiles%\Windows Media Player	Disallowed	All file types

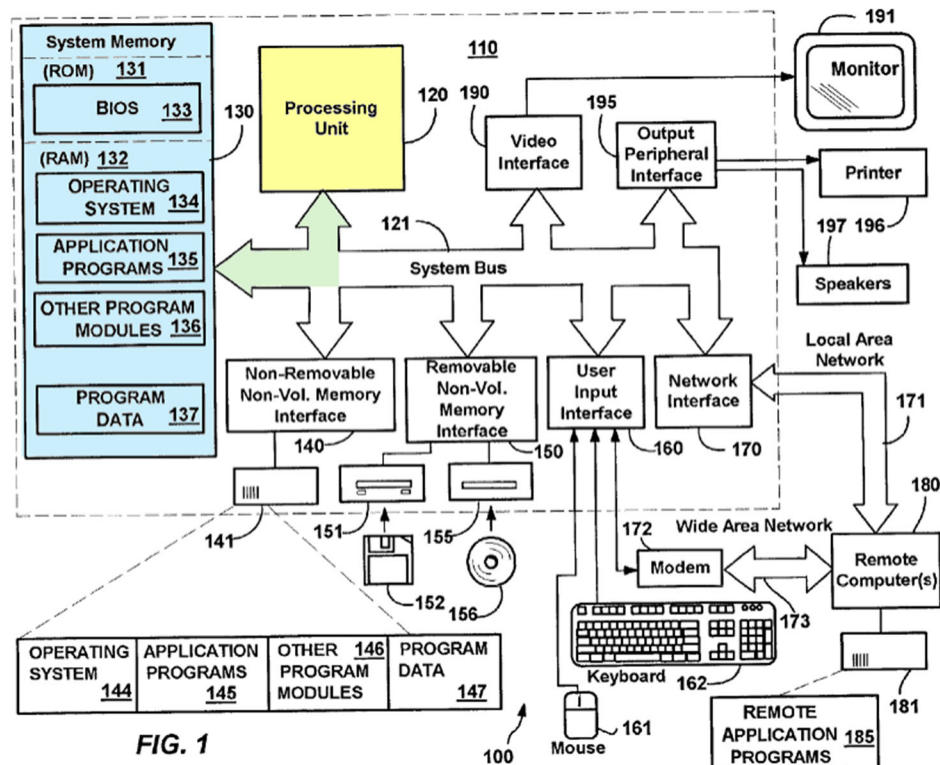
(*Id.*, [0082], [0079]; Polish, ¶199.) Each of these “code authorization levels” can be shared by a group of applications. (Lambert, [0082].) Lambert thus teaches the claimed **“group containing two or more of the applications is associated with one of the authorization records.”** (Polish, ¶199.)

J. Claim 10

1. **“A device, comprising:” (Claim 10[p]); “a memory; and” (Claim 10[a]); “a processor coupled with the memory and configured by instructions stored in the memory to:” (Claim 10[b])**

Claims 10[p], 10[a], 10[b] are disclosed by Lambert (Grounds 1-2). Lambert discloses controlling execution of applications on a computer (“a device”).

(Lambert, [0026], Fig. 1, [0027] (“personal computers, server computers, hand-held or laptop devices....”); Polish, ¶398; *see supra* claim 1[p].) Lambert’s system is implemented on a device with “**processor coupled with the memory.**” Lambert teaches: “**processing unit 120**” (“**processor**”) coupled to the “**system memory 130**” (“**memory**”):



(Lambert, Fig. 1 (highlighting added); Polish, ¶¶200-205.) Lambert discloses that the processor is “**configured by instructions stored in the memory.**” (Lambert, [0031]; Polish, ¶205.)

2. “receive owner information from an owner of the device;” (Claim 10[b][i]) and “receive owner control information that comprises:” (Claim 10[b][ii]); “a list of applications permitted for installation on the device, and” (Claim 10[b][ii][A]); “permissions associated with each permitted application in the list of applications;” (Claim 10[b][ii][B]); “store the owner control information in a protected data store to prevent alteration or deletion of the owner control information in response to authenticating the owner control information using the owner information” (Claim 10[b][iii]); “receive, from an application operating on the electronic device, a request to access local memory on the electronic device;” (Claim 10[b][iv]); “determine whether the application is allowed to access the local memory in response to identifying a permission associated with the application in the owner control information; and” (Claim 10[b][v]); “control application access to the local memory based upon the determining.” (Claim 10[b][vi])

These elements are obvious for the same reasons explained for claims 1[a]-1[f]. (Polish, ¶¶206-213; EX1020.)

- K. **Claim 11:** “The device of claim 10, wherein the request to access the local memory comprises a request to erase data from the local memory.”

This element is obvious for the same reasons explained for claim 2.
(Polish, ¶215; EX1020.)

- L. **Claim 12:** “The device of claim 10, wherein the request to access the local memory comprises a request to read data from the local memory.”

This element is obvious for the same reasons explained for claim 3.
(Polish, ¶217; EX1020.)

- M. Claim 13: “The device of claim 10, wherein the processor is further configured to restrict modification of the owner control information after the owner control information has been stored on the electronic device.”**

Claim 13 is substantially the same as claim 4 (Grounds 1-2). Claim 13 additionally recites that “**the processor is further configured**” to restrict modification. As discussed for claim 10[b], Lambert discloses that the computing device includes a processor, and as explained for claim 10, that processor would have been “**configured**” to perform the claimed operations. (Polish, ¶219.) The processor, in carrying out policy-based controls, would further be configured to restrict modification. (*Id.*) Claim 13 also recites restricting modification of the owner control information after it “**has been stored on the electronic device.**” This element is obvious for the same reasons explained for claims 1[c] and 4. (Polish, ¶219.)

- N. Claim 14: “The device of claim 10, wherein the owner control information includes an authorization record and an application identifier for each of the applications, and wherein the owner control information is provided by an external source through a wireless communication channel.”**

This element is obvious for the same reasons explained for claim 5. (Polish, ¶221; EX1020.)

- O. Claim 15: “The device of claim 14, wherein the authorization record related to determining whether access to the local memory is allowed is provided by an external computer that is used to enforce policies for operating electronic devices within an organization.”**

This element is obvious for the same reasons explained for claim 6.

(Polish, ¶223; EX1020.)

- P. Claim 16: “The device of claim 10, wherein an update to the owner control information is provided by an external source for storage on the electronic device.”**

This element is obvious for the same reasons explained for claim 7.

(Polish, ¶224; EX1020.)

- Q. Claim 17**

- 1. “A non-transitory computer-readable medium with instructions stored thereon, wherein the instructions are executable by a processor to cause the processor to perform the operations of:” (Claim 17[p])**

To the extent limiting, claim 17[p] is disclosed by Lambert (Grounds 1-2).

Lambert is implemented on a computer with a “**non-transitory computer-readable medium with instructions stored thereon.**” Lambert discloses **system memory 130** which “includes computer storage media in the form of volatile and/or nonvolatile memory such as read only memory (ROM) 131 and random access memory (RAM) **132.**” (Lambert, [0031].)

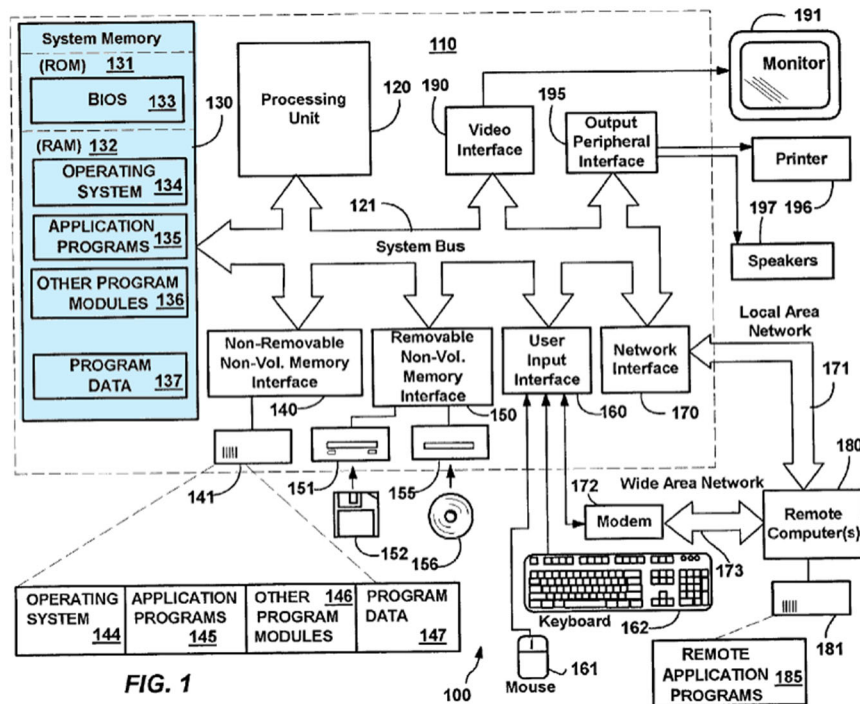
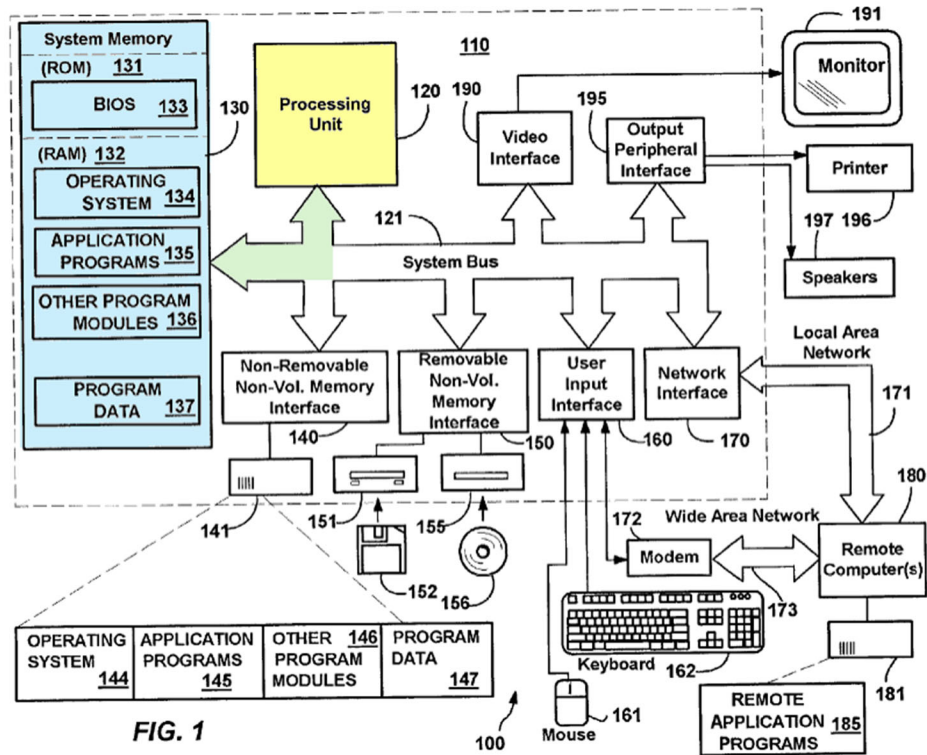


FIG. 1

(*Id.*, Fig. 1 (highlighting added); Polish, ¶227.) Lambert teaches the memory contains “the basic routines that help to transfer information between elements within computer 110” and RAM 132 contains “program modules that are immediately accessible to and/or presently being operated on by processing unit 120”—*i.e.*, “instructions.” (Lambert, [0031]; Polish, ¶227.)

Lambert discloses “a processor”—*i.e.*, processing unit 120:



(Lambert, Fig. 1 (highlighting added), [0031]; Polish, ¶228.) As discussed above for claim 10[b], Lambert discloses that processing unit **120** executes instructions to perform the claimed operations. (Lambert, [0031]; Polish, ¶228.)

2. **“receiving owner information from an owner of an electronic device;” (Claim 17[a]) and “receiving owner control information that comprises:” (Claim 17[b]); “a list of applications permitted for installation on the device, and” (Claim 17[b][i]); “permissions associated with each permitted application in the list of applications;” (Claim 17[b][ii]); “storing the owner control information in a protected data store to prevent alteration or deletion of the owner control information in response to authenticating the owner control information using the owner information;” (Claim 17[c]); “receiving, from an application operating on the electronic device, a request to access local memory on the electronic device;” (Claim 17[d]); “determining whether the application is allowed to access the local memory in response to identifying a permission associated with the application in the owner control information; and” (Claim 17[e]); “controlling application access to the local memory based upon the determining.” (Claim 17[f])**

These elements are obvious for the same reasons explained for claims 1[a]-1[f]. (Polish, ¶¶229-236; EX1020.)

- R. **Claim 18: “The non-transitory computer-readable medium of claim 17, wherein the owner control information includes an authentication record and an identifier for each application allowed to be executed on the electronic device.”**

Lambert renders claim 18 obvious (Grounds 1-2). As discussed for claim 1[b], Lambert discloses **“owner control information”** in the form of policy rules for application execution. As discussed for claim 5, that information includes for an application **“an authentication record”**—*i.e.*, the associated rule (*e.g.*, “Disallowed,” “Unrestricted,” etc.)—and an **“identifier”**—*i.e.*, filepath or hash value. (Lambert, [0082], [0079], [0053]-[0054], [0058]-[0063], [0068]; ’571, 7:46-

53; Polish, ¶239.) Because Lambert relates to controlling application operation on a computing device, a POSITA would have understood the policy to include an authentication record and an identifier **“for each application allowed to be executed on the electronic device”** based on Lambert. (Lambert, [0011]; Polish, ¶239.)

Lambert discloses that policy rules can use a “software publisher digital signature (e.g., certificate)” indicating the source of an application. (Lambert, [0057], [0084].) This digital signature information also constitutes an **“authentication record.”** (’571, 7:35-39, 9:52-58, 10:18-21, 23:21-23; Polish, ¶240.) Lambert renders obvious associating **“an authentication record and an identifier for each application allowed to be executed on the electronic device.”** (Polish, ¶241.) Lambert discloses that a rule could include both digital signature and identifying information. (Lambert, [0057], claims 5, 6, 18, [0068], [0066].) Lambert teaches that the policy rules can include “a multi-part rule that checks for allowed/restricted/disallowed signatures and allowed/restricted/disallowed control identifiers, and taking the most restrictive result from the evaluated parts of the rule[.]” (*Id.*, [0057].) A POSITA would have been motivated to use digital signature rules, since such rules can ensure that applications have a trusted source. (*Id.*, [0057], [0011]; Polish, ¶241.)

- S. Claim 19: “The non-transitory computer-readable medium of claim 17, wherein the request to access the local memory comprises a request to erase data from the local memory.”**

This element is obvious for the same reasons explained for claim 2.

(Polish, ¶243; EX1020.)

- T. Claim 20: “The non-transitory computer-readable medium of claim 17, wherein the request to access the local memory comprises a request to read data from the local memory.”**

This element is obvious for the same reasons explained for claim 3.

(Polish, ¶244; EX1020.)

VIII. GROUNDS 3-5: CLAIMS 1-8 AND 10-20 ARE OBVIOUS OVER HAYES IN VIEW OF SPRIGG (GROUND 3), OR OBVIOUS OVER GROUND 3 + FRANK AND GUEDALIA (GROUND 4); CLAIMS 9-20 ARE OBVIOUS OVER GROUND 4 + LAMBERT (GROUND 5)

A. Claim 1

- 1. “A method of application control on an electronic device, the method comprising:” (Claim 1[p])**

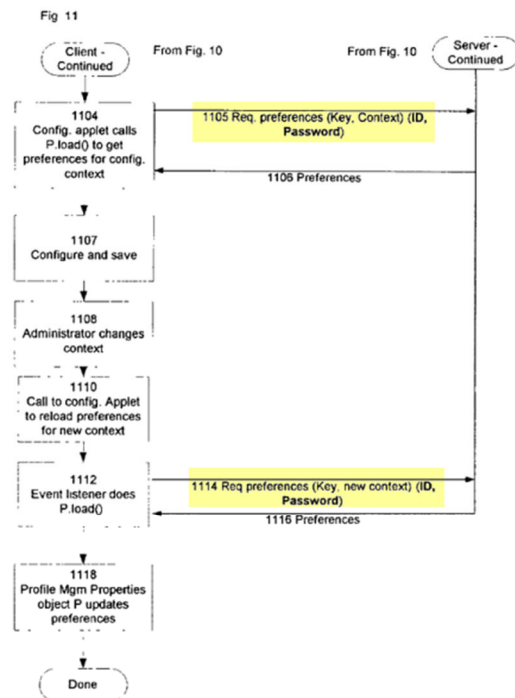
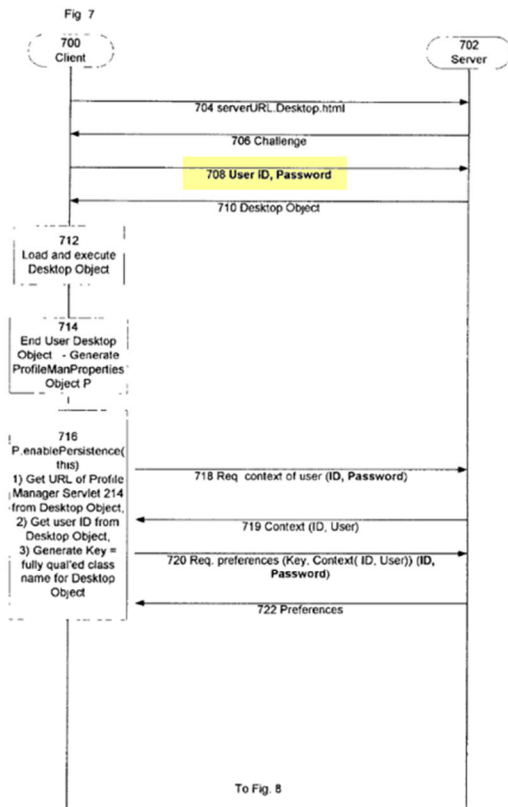
Hayes discloses administrators controlling what applications users can download (Grounds 3-4). (Hayes, [0008], [0012], [0013], Figs. 8, 13, 17, 24, [0044]-[0046], Fig. 3.) Hayes thus discloses “[a] **method of application control.**” (Polish, ¶260.) Hayes discloses application control “**on an electronic device.**” (Hayes, [0037] (“personal computers”), [0011], Fig. 1; Polish, ¶261.)

- 2. “receiving, by the electronic device, owner information from an owner of the electronic device;” (Claim 1[a])**

Claim 1[a] is disclosed by or obvious over Hayes (Grounds 3-4).

Hayes teaches the claimed “**owner information**,” such as the ID and password information provided by the administrator. (Hayes, [0075]-[0077], [0080], [0082], Figs. 7-11; ’571, 5:25-28, 7:35-39; Polish, ¶263.) Hayes’s ID and password information constitutes “**owner information from an owner of the electronic device**,” because like the ’571 patent, Hayes provides that the ID and password are input by authorized administrators of a network “such as a corporate intranet[.]” (Hayes, [0037], [0001]-[0010], [0073], [0039], [0036], [0004], [0025]; ’571, 5:63-67, 6:44-50, 11:37-40, 19:14-17, 9:11-17, 8:7-10, 10:58-62; Polish, ¶263.) The information provided by these administrators is “**from an owner of the electronic device**”—*i.e.*, the owner (*e.g.*, corporation). (Hayes, [0073], [0037]; Polish, ¶¶264-266.)

Hayes discloses that the owner information is “**receiv[ed], by the electronic device**.” For an administrator to run the profile manager applet, the administrator must provide “a user ID and password.” (Hayes, [0073]; [0011]; Polish, ¶267.) If the administrator “has permission to run the profile manager,” the profile manager applet is downloaded to the administrator’s terminal. (Hayes, [0073].) Hayes also discloses that the administrator submits a user ID and password at various steps of the process of loading and executing the profile manager applet—highlighted below. (*Id.*, [0075]-[0077], [0080], [0082], Figs. 7-11.)



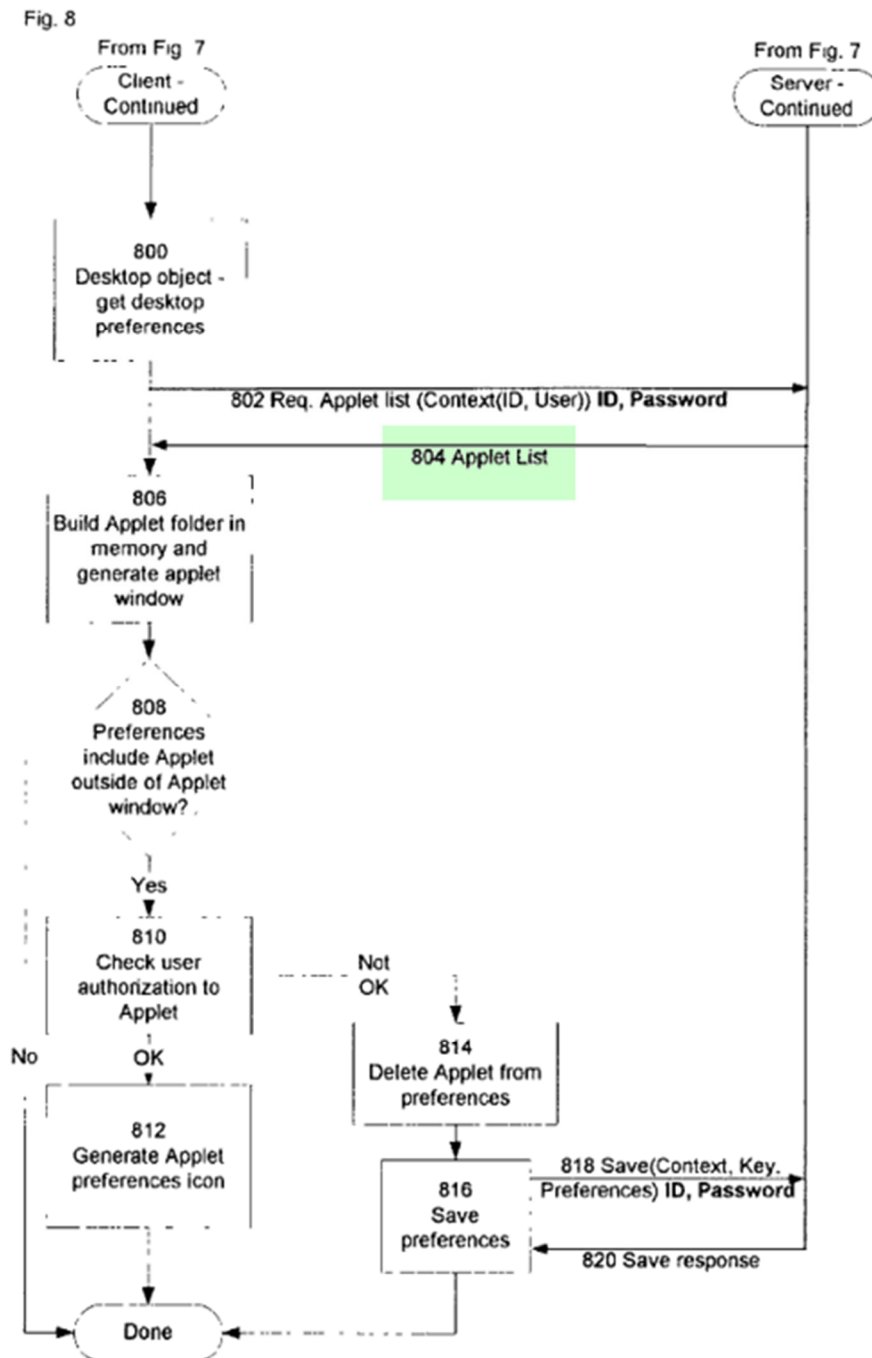
(*Id.*, Figs. 7, 11 (highlighting added).) Hayes thus teaches “**receiving, by the electronic device, owner information from an owner of the electronic device.**” (Polish, ¶267.)

It also would have been obvious based on Hayes that an administrator could log in to any terminal using administrative privileges. (Hayes, [0011], [0073]; Polish, ¶268.) A POSITA would have been motivated to implement Hayes’s system to allow “roam[ing]” administrators that could adjust preferences for users on the same device where those preferences would be applied or enforced. (Hayes, [0011]; Polish, ¶268.) Allowing a user to log in on any workstation would provide for further

“administrative flexibility of operation” by allowing an administrator to see how the set preferences work on a device. (Hayes, [0007]; Polish, ¶268.) As discussed for claim 1[b][ii] below, such administrator use of a user terminal would further have been obvious in view of Sprigg and Frank. (Polish, ¶269)

3. **“receiving, by the electronic device, owner control information that comprises:” (Claim 1[b]) and “a list of applications permitted for installation on the electronic device, and” (Claim 1[b][i])**

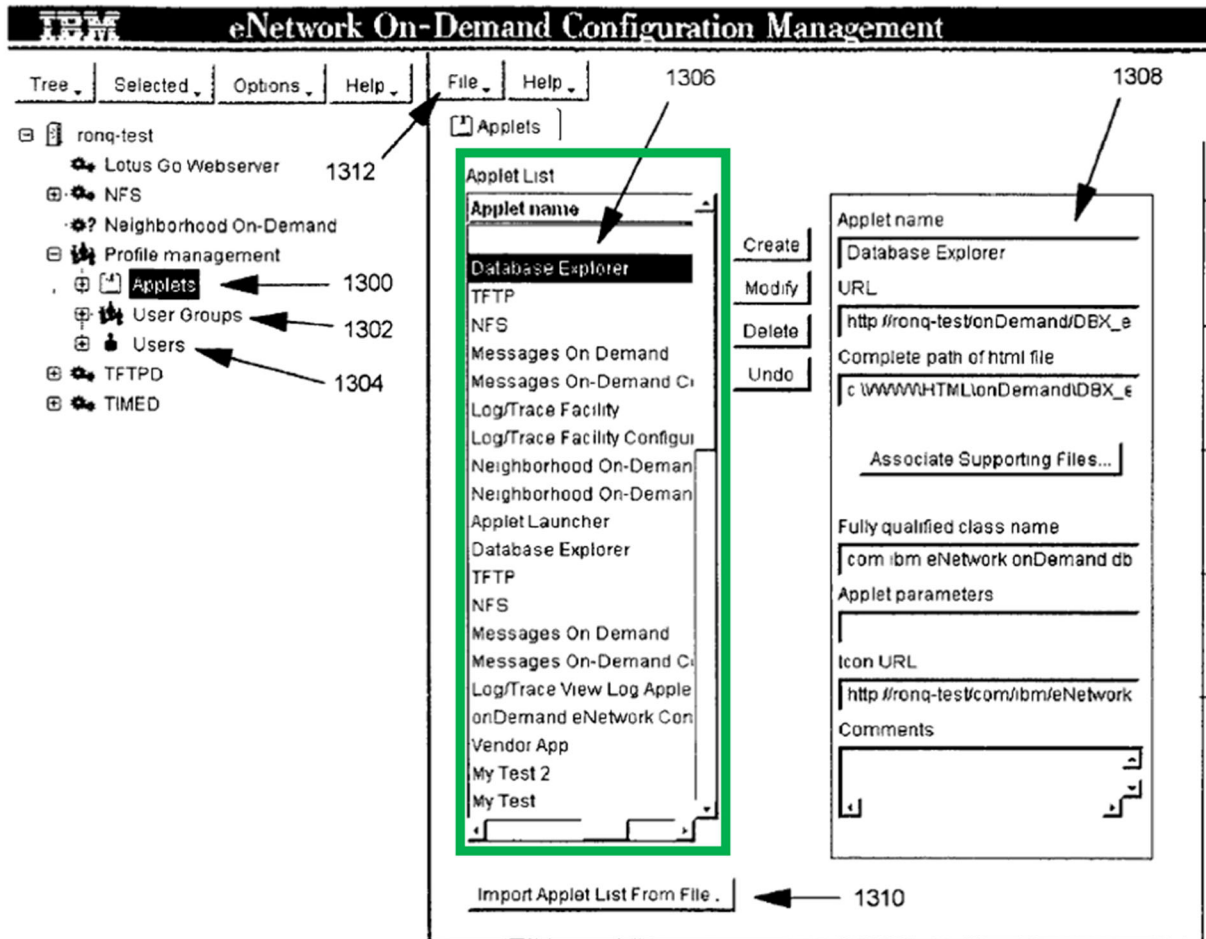
Claim 1[b] and claim 1[b][i] are disclosed by or obvious over Hayes (Grounds 3-4). Hayes discloses receiving **“a list of applications permitted for installation on the electronic device”** set by an administrator, such as on a corporate intranet system. Hayes teaches that when a user brings up the desktop on a terminal, the “desktop object calls a method... to get a list of the software to which the user has access permission.” (Hayes, [0071]; Polish, ¶¶271-276.) “[T]he server stores a plurality of user applications for downloading to user stations and further stores access permissions for the applications.” (Hayes, [0013], claim 1, [0012].) The computer’s receipt of the list of allowed software from the server (**highlighted**) is receiving **“by the electronic device, ... a list of applications permitted for installation on the electronic device.”**



(Hayes, Fig. 8 (highlighting added), [0025]; Polish, ¶276.)

Hayes also teaches an “Applet List” with options for an administrator to “Create” or “Delete” applets:

Fig. 13



(Hayes, Fig. 13 (highlighting added), [0085], Figs. 24 & 17, [0093], [0088]; Polish, ¶277.) Hayes thus teaches “owner control information” that comprises a “list of applications permitted for installation on the electronic device.” (Polish, ¶277.) Hayes’s list of allowed applications is “owner control information,” because it controls what applications an end user can download on his or her terminal. (*Id.*, ¶271.) Such information is “owner...information,” because it is provided by an administrator authorized by an owner (e.g., the corporation). (Hayes, [0037], [0001],

[0002]-[0010], [0073]; '571, 5:63-67, 6:44-50, 11:37-41, 19:14-17, 9:11-17, 8:7-10, 10:58-62; Polish, ¶¶271, 278.) The allowed applications are permitted by the owner, since they are permitted by the administrator. (Hayes, [0037], [0001]-[0010], [0073]; Polish, ¶278.)

Claim 1[b][i] would have further been obvious over Hayes in view of **Guedalia** (EX1013; Ground 4). (Polish, ¶¶280-288.) Guedalia discloses that Java applets “automatically download from the server and install themselves, triggered by the HTML page[.]” (Guedalia, 1:63-66, 1:5-17, 1:57-63.) Hayes discloses that its system can use Java applets, where the applets constitute the applications in the “**list of applications permitted for installation.**” (Hayes, [0011], [0025], Fig. 8, [0026], [0013], [0012]; Polish, ¶282.) Guedalia supports that Hayes discloses or renders obvious a “**list of applications permitted for installation.**” (Polish, ¶¶272-277.) Guedalia indicates that a POSITA would have understood that Hayes’s Java applets are both downloaded and installed on a user terminal. (Guedalia, 1:63-66; Polish, ¶284.)

Motivation to Combine: Both Guedalia and Hayes are analogous art to the '571 patent. (Polish, ¶285; Guedalia, 1:5-17; '571, 1:41-43, 6:44-50, 1:67-2:4; Hayes, [0005], [0006].) Both relate to providing applications in a server-client architecture. (Hayes, [0001], [0002]; Guedalia, 1:5-6, 1:57-2:3, 2:40-54.) Both use Java applets. (Hayes, [0011], [0025], Fig. 8, [0026], [0013], [0012]; Guedalia, 1:57-

2:3, 5:18-20, 12:21-30, 13:8-17; Polish, ¶¶285, 289.) A POSITA would have been motivated to combine Hayes and Guedalia in determining how to implement Hayes's system, since Guedalia discusses the use of various types of software to provide applications in a client-server architecture and the relative advantages of the various types. (Guedalia, 1:57-2:3; Hayes, [0026]; Polish, ¶286.) A POSITA would have been motivated to look to Guedalia for implementation details on Java applets. (Guedalia, 1:60-67; Polish, ¶287.) A POSITA would have had more than a reasonable expectation of success and could have implemented the proposed combination using conventional techniques. (Polish, ¶288; EX1014, p.002.)

4. “permissions associated with each permitted application in the list of applications;” (Claim 1[b][ii])

Hayes discloses that an administrator can input application preferences for specific users or groups. (Hayes, [0044]-[0046], Fig. 3; Polish, ¶¶291-292.) Hayes does not explicitly disclose an administrator using its system for providing application “**permissions**,” but this would have been obvious over **Sprigg** (EX1012; Grounds 3-4.) Claim 1[b][ii] is obvious over Hayes and Sprigg.

Sprigg discloses a computer network where a wireless device **300** contains “applications **310** each with a permission list **320**.” (Sprigg, 5:41-43; Polish, ¶¶252-254, 292-301.) These lists “contain[] a listing of the resources **315** and an indication whether the application has permission to access any of the specific resources **315** on the device.” (Sprigg, 5:63-66, 4:14-21.) The permission lists can be “created by

an intermediate authority,” including “determination of the application’s access to the device’s resources by those authorities, entities, or parties that may be affected by the misuse of a device’s resource.” (*Id.*, 6:12-17.) In the combination, an administrator would provide “**permissions associated with each permitted application in the list of applications.**” (Polish, ¶293.) For each allowed application on Hayes’s network, an administrator could input a set of permissions for that application. (Hayes, Figs. 12-24; Polish, ¶293.)

Motivation to Combine: Hayes and Sprigg are analogous to the ’571 patent, because each relates to management of applications by administrators. (Hayes, [0001]; Sprigg, 1:13-15, 3:36-41; ’571, 1:41-43; Polish, ¶¶296, 301.) Like the ’571 patent, Hayes and Sprigg relate to computer security. (Hayes, [0005], [0006]; Sprigg, 1:31-36; ’571, 1:67-2:4.) It would have been obvious to implement Sprigg’s permission lists in the network control system disclosed by Hayes. (Polish, ¶294.) In such a system, in addition to providing a list of allowed applications and preferences like Hayes discloses, an administrator would also provide a permission list for each allowed application indicating whether the application has permission to access resources on the device, including memory. (*Id.*, ¶295.) A POSITA would have been motivated to combine Hayes with Sprigg to allow for more fine-grained management of application use. (*Id.*, ¶297; Hayes, [0013]; Sprigg, 5:63-66, 4:14-21, 6:12-17, 2:11-15, 1:61-2:10.) A POSITA would also have been motivated to

Petition for *Inter Partes* Review of
U.S. Patent No. 9,542,571 B2

improve computer security in Hayes's system. (Sprigg, 1:45-52, 2:19-22; Polish, ¶298.) A POSITA would have been further motivated based on Hayes's disclosures regarding administrators setting preferences for applications. (Hayes, [0044]-[0046]; Polish, ¶299.) A POSITA would have had more than a reasonable expectation of success and could have implemented the combination using conventional techniques. (Polish, ¶300; '571, 1:63-67.)

Claim 1[b][ii] would have further been obvious based on Hayes in light of Sprigg and **Frank** (EX1005; Ground 4). Like Hayes and Sprigg, Frank discloses a computer security system managed by administrators to control user activity. (Frank, [0002], [0029], [0005]-[0012]; Polish, ¶303.) In Frank, an administrator can input and apply security rules on a user device. (Frank, [0043], [0040]-[0042], [0028]-[0030], [0011], [0012].) It would have been obvious in light of Frank to implement the combined system to enable the administrator to log-in to a terminal using administrator credentials ("**owner information**") and adjust the "**owner control information**" for that computer (allowed applications and application resource restrictions)—all on the user terminal where that "**owner control information**" would subsequently be enforced for a non-administrator user. (Polish, ¶304.) This combination would allow an administrator to view the impact of any security policy setting (e.g., application permissions) in real time. (*Id.*)

Motivation to Combine: Frank is analogous to the '571 patent, as with Hayes and Sprigg (as explained above). (Polish, ¶¶305, 311.) Each reference relates to administrators setting computer security policy for users. (Frank, [0002], [0029], [0005]-[0012]; Hayes, [0002], [0010], [0027], [0001], [0004]; Sprigg, 3:37-41.) A POSITA would have been motivated to implement a security application on a user terminal to enable an administrator to easily see the results of changes to security policy—similar to Hayes’s disclosure of viewing applications from the perspective of a particular user. (Polish, ¶306; Hayes, [0088], [0007], [0093], [0094], Figs. 17, 24, [0007].) A POSITA would have further been motivated to combine Hayes, Sprigg, and Frank to streamline the process of inputting security policy. (Polish, ¶307.) If an administrator can easily see the results of a security policy in real time on a user terminal, a POSITA would understand that this would make it less likely that the security policy would need to go through multiple iterations. (*Id.*, ¶308; Hayes, [0010]; Sprigg, 1:45-52; Frank, [0012], [0011], [0036] (“better decisions when defining a security profile”).) A POSITA would have been further motivated to implement an administrator security application on a user terminal in the combination because formulating security policy for a computer network requires an iterative balancing approach. (Polish, ¶309; Frank, [0008].) A POSITA would have had more than a reasonable expectation of success and could have implemented the proposed combination using conventional techniques. (Polish, ¶310.)

5. **“storing, by the electronic device, the owner control information in a protected data store to prevent alteration or deletion of the owner control information in response to authenticating the owner control information using the owner information;” (Claim 1[c])**

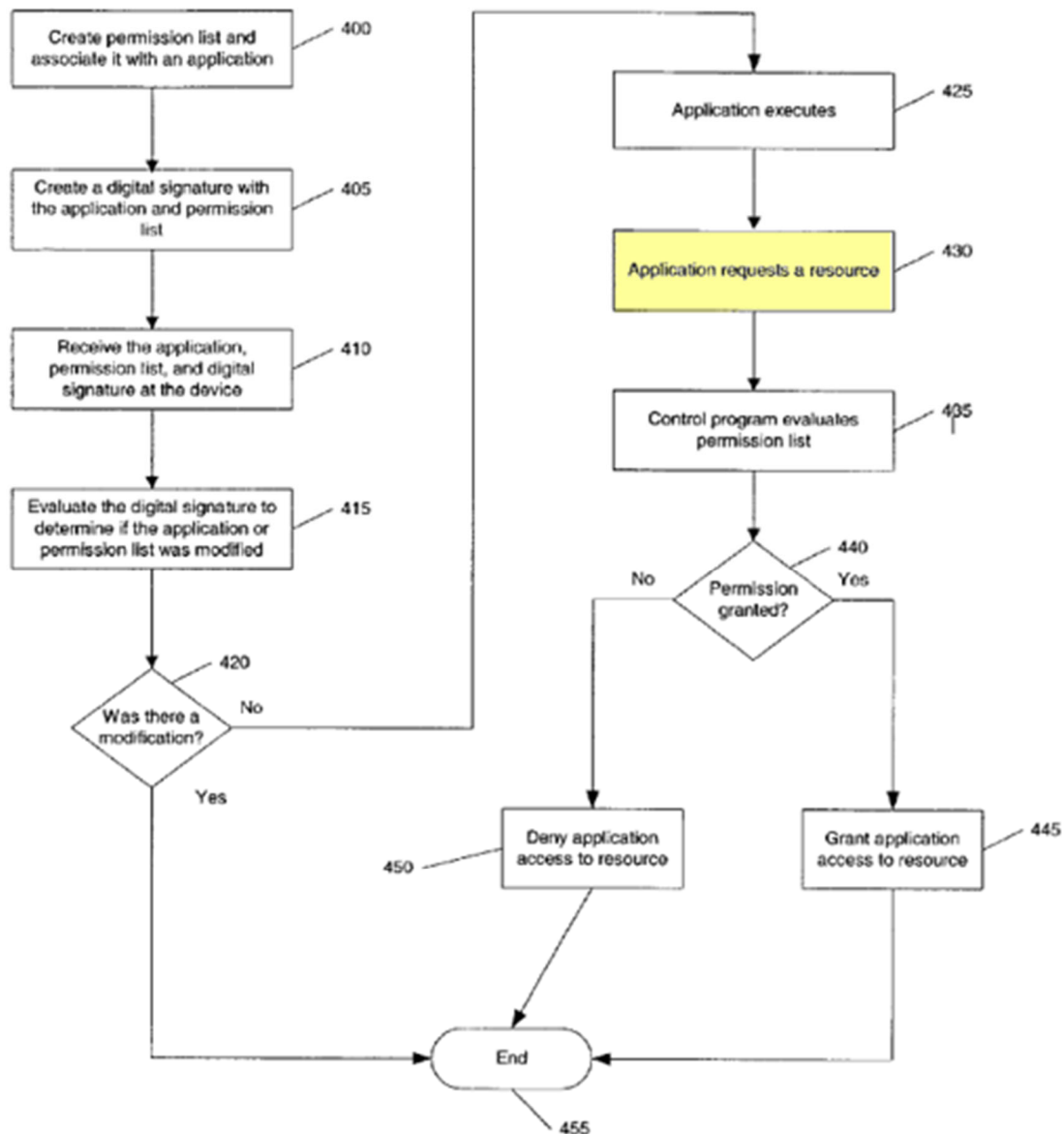
Claim 1[c] is disclosed by or would have been obvious based on Hayes, or based on Hayes and Sprigg (Grounds 3-4). Hayes discloses that the administrator-provided information, including allowed applications and application preferences, are provided to a password-protected server database. (Hayes, [0085], Figs. 17, 24, [0042] (“permanent storage”), [0055], [0060], [0065], [0067], [0081], Figs. 5, 6, 11; Polish, ¶313.) Since the server database is password protected, saving the owner control information via the profile manager application to the server database is **“storing, by the electronic device, the owner control information in a protected data store to prevent alteration or deletion of the owner control information.”** (Polish, ¶¶314-315; ’571, 10:47-51, 13:46-51, 20:3-10, 21:8-10, 22:59-60.) Alteration or deletion is prevented for users (*e.g.*, end users) with no administrative credentials. (Polish, ¶314.) This is **“in response to authenticating the owner control information using the owner information,”** because as discussed above, the administrator cannot access the profile manager for modifying owner control information unless administrative credentials (user ID and password) are provided. (Hayes, [0073], [0075], [0076], [0080], [0082], Figs. 7-11; Polish, ¶¶314, 260-271.)

It would have been obvious based on Hayes in view of Sprigg that **“owner**

control information” would be stored in a **“protected data store to prevent alteration or deletion.”** Sprigg discloses that its system checks whether permission information was modified before executing an application being checked against the permission information. (Sprigg, 1:45-52, Fig. 4; *id.*, 4:33-37, 5:25-38, 6:22-33, 6:51-7:23.) If it is determined that the permission information was modified (*e.g.*, digital signature), then Sprigg discloses preventing the application from executing. (*Id.*) Thus, the combined system could store **“owner control information”** in a **“protected data store to prevent alteration or deletion of the owner control information”**—*i.e.*, prevent alteration by preventing applications from executing if alteration is detected. (Polish, ¶316.)

6. “receiving, by the electronic device, from an application operating on the electronic device, a request to access local memory on the electronic device;” (Claim 1[d])

Claim 1[d] is obvious over Hayes and Sprigg (Grounds 3-4). As discussed above, Sprigg discloses that in enforcing application permissions, a computer **“receiv[es], by the electronic device, from an application operating on the electronic device, a request to access local memory on the electronic device.”** (Polish, ¶318.) Sprigg discloses that an “application executes on [a wireless device] (Step 425).” (Sprigg, 7:20-23.) In doing so, the application “requests a resource on the device (Step 430)”—*i.e.*, the device **“receiv[es], by the electronic device, from an application operating on the electronic device, a request to access”** a resource.



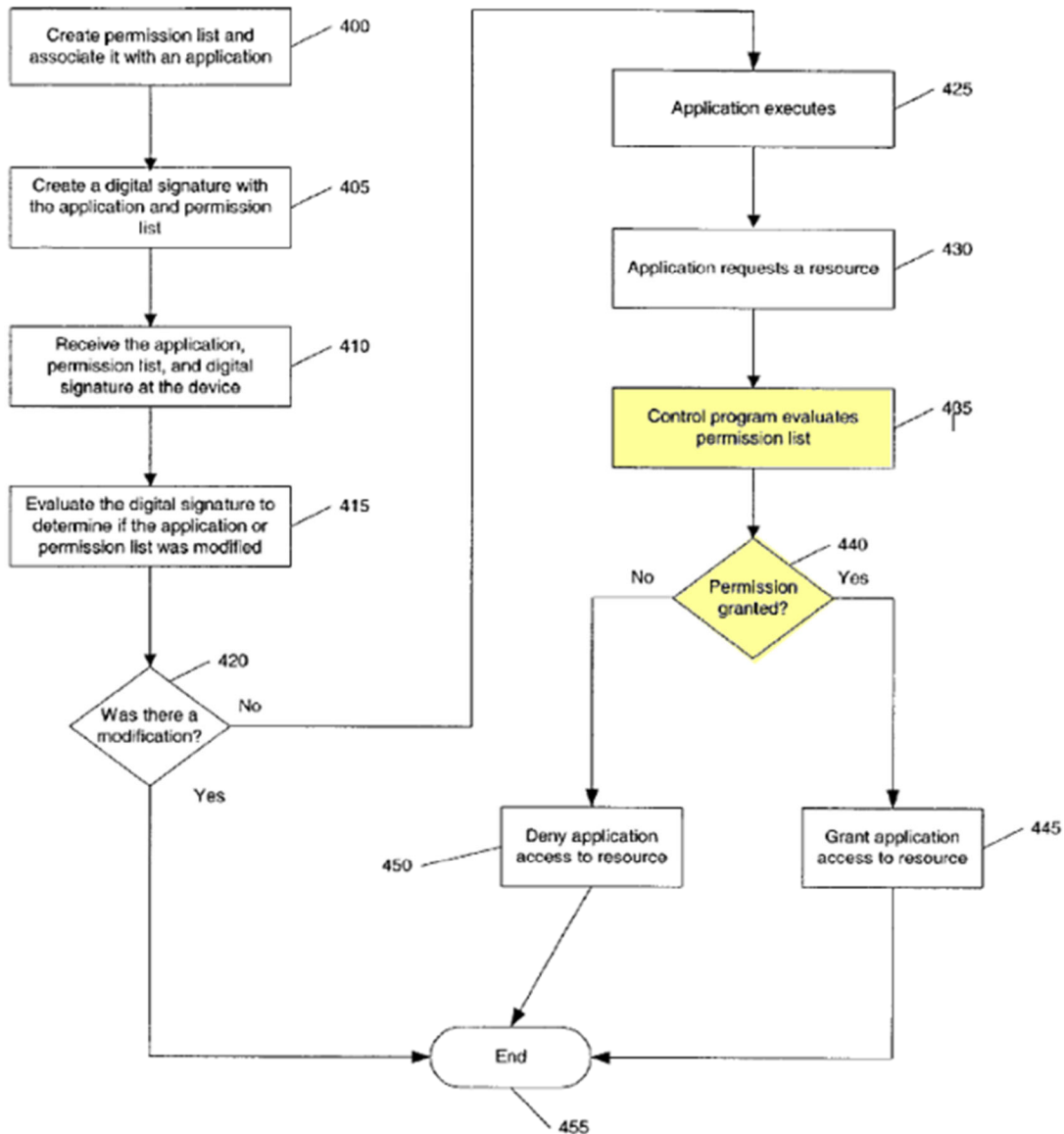
(*Id.*, Fig. 4 (highlighting added); Polish, ¶¶317-320.) Sprigg discloses that the resource request can be to the local memory of the device—*i.e.*, “**a request to access local memory on the electronic device.**” (Sprigg, 4:25-32, 4:14-21 (“memory, including primary and secondary storage”), 1:37-41; Polish, ¶319.)

As explained above, it would have been obvious to implement Hayes's system according to Sprigg, such that a device receives administrator-set application permissions that are associated with applications and enforced on a user device. (Polish, ¶320.)

7. **“determining whether the application is allowed to access the local memory in response to identifying a permission associated with the application in the owner control information; and” (Claim 1[e])**

Claim 1[e] is obvious over Hayes and Sprigg (Grounds 3-4). As discussed above, Sprigg discloses that in response to an application resource request (including a local memory request), a control program on a device “evaluates the permission list associated with the application” and determines “[i]f the permission list indicates that the application has the privilege to access the requested resource (Step 440)” or not. (Sprigg, 7:30-44, 4:25-32 (“application may request access to memory”).) Sprigg thus discloses **“determining whether the application is allowed to access the local memory in response to identifying a permission associated with the application in the owner control information[.]”** (Polish, ¶¶322, 291-301.)

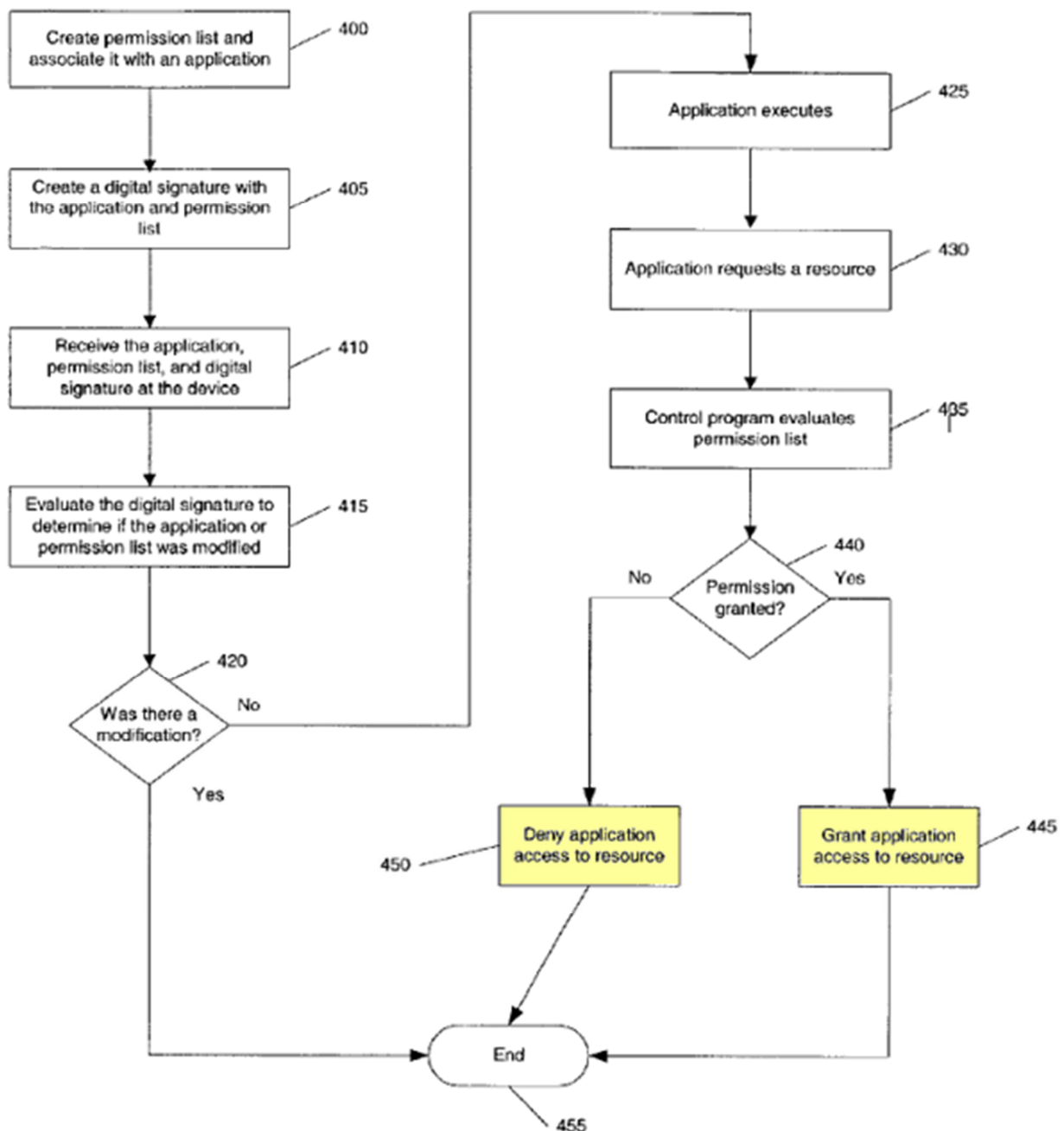
Figure 4 depicts:



(Sprigg, Fig. 4 (highlighting added); Polish, ¶322.) As explained above, it would have been obvious to implement Hayes's system according to Sprigg, such that a device receives administrator-set application permissions that are associated with applications and enforced on a user device. (Polish, ¶¶323, 299-301.)

8. “controlling application access to the local memory based upon the determining.” (Claim 1[f])

Claim 1[f] is obvious over Hayes and Sprigg (Grounds 3-4). As discussed above, Sprigg discloses that after determining whether an application’s access to a resource (including local memory) is allowed based on the relevant permissions list, the wireless device either prevents or allows the application to access the resource—thereby **“controlling application access to the local memory based upon the determining.”** (Polish, ¶325.) Sprigg discloses that “[i]f the permission list indicates that the application has the privilege to access the requested resource (Step **440**), the application will be granted access to the resource (Step **445**).” (Sprigg, 7:30-34.) Otherwise, “the request to access the resource is denied (Step **450**).” (*Id.*, 7:35-44.) Such control is based upon the **“determining”** of application access from permissions:



(*Id.*, Fig. 4 (highlighting added), 7:30-44; Polish, ¶325.) As discussed above, it would have been obvious to implement Hayes's system according to Sprigg. (Polish, ¶326.)

B. Claim 2: “The method of claim 1, wherein the request to access the local memory comprises a request to erase data from the local memory.”

Claim 2 is obvious over Hayes and Sprigg (Grounds 3-4). As discussed for claim 1[d], Sprigg discloses that the resources for which access can be controlled includes “files stored in memory, including the read, write and modify operations on such files[.]” (Sprigg, 4:14-21, 4:25-32, 4:34-42, 7:24-43, Figs. 1, 4; *see supra* claim 1[d].) Since “write and modify operations” on a file entail “eras[ing] data” (*i.e.*, erasing the data overwritten by the write or modify operation), Sprigg discloses “**the request to access the local memory comprises a request to erase data from the local memory.**” (Polish, ¶¶330-331.)

C. Claim 3: “The method of claim 1, wherein the request to access the local memory comprises a request to read data from the local memory.”

Claim 3 is obvious over Hayes and Sprigg (Grounds 3-4). As discussed for claims 1[d] and 2, Sprigg discloses managing the “read” operations on files stored in memory. (Sprigg, 4:14-21, 4:25-32, 4:34-42, 7:24-43, Figs. 1, 4.) Accordingly, Sprigg discloses and renders obvious “**the request to access the local memory comprises a request to read data from the local memory.**” (Polish, ¶¶334-335.)

D. Claim 4: “The method of claim 1, further comprising restricting modification of the owner control information after the owner control information has been stored in the protected data store.”

Claim 4 is disclosed by or obvious over Hayes (Grounds 3-4). Claim 4 is also obvious over Hayes and Sprigg. As explained for claim 1[c], Hayes teaches the

claimed “**protected data store.**” (Polish, ¶¶314-315.) Hayes discloses that the administrator-provided information, including allowed applications and application preferences, is provided to a password-protected server. (Hayes, [0085], [0042] (“permanent storage”), [0055], [0060], [0065], [0067], [0081], Figs. 5, 6, 11, 17, 24.) Since the server database is password protected, Hayes’s system “**restrict[s] modification of the owner control information after the owner control information has been stored in the protected data store[.]**” (Polish, ¶339.) Modification is prevented for non-administrator users (*e.g.*, end users) who do not have administrative credentials. (Hayes, [0073], [0075], [0076], [0080], [0082], Figs. 7-11; Polish, ¶339.)

Further, it would have been obvious based on Hayes in view of Sprigg that once “**owner control information**” is stored on a user device, modification of that information would be “**restrict[ed].**” Sprigg discloses that its system checks whether permission information was modified before executing an application being checked against the permission information. (Sprigg, 1:45-52, Fig. 4, 4:33-37, 5:25-38, 6:22-33, 6:51-7:23.) If it is determined that the permission information was modified (for example, on the basis of a digital signature), then Sprigg discloses preventing the application from executing. (*Id.*) Thus, the combined system “**restrict[s] modification of the owner control information** [*i.e.*, prevents modification without consequences to the execution of applications] **after the**

owner control information has been stored in the protected data store[.]”

(Polish, ¶339.)

- E. Claim 5: “The method of claim 1, wherein the owner control information includes an authorization record and an application identifier for each of the applications, and the owner control information is provided by an external source through a wireless communication channel.”**

Claim 5 is obvious over Hayes and Sprigg (Grounds 3-4). As explained above, Hayes discloses the claimed **“owner control information”** and Hayes teaches a server (including servlet **214**) that provides the **“list of applications permitted for installation.”** (Hayes, [0071], Fig. 8; Polish, ¶342.) Hayes discloses that **“the owner control information includes... an application identifier for each of the applications”** and the **“owner control information is provided by an external source [e.g., a server].”** (Hayes, Fig. 13, [0085], Figs. 24 & 17, [0093], [0088], [0071], [0013], claim 1; Polish, ¶¶342, 260-311.) Hayes discloses that the **“owner control information is provided... through a wireless communication channel”**—e.g., a local area network (LAN) or wide area network (WAN), which a POSITA would have understood to include wireless communication channels like Wi-Fi. (Hayes, Fig. 1, [0037]; EX1015 (Spencer), p.006 (“wide area network”); Polish, ¶343.) A POSITA would have been motivated to use a **“wireless communication channel”** to enable Hayes’s network for more devices. (Hayes, [0037]; Polish, ¶343.) As explained above, the claimed **“owner control**

information includes an authorization record and an application identifier for each of the applications” would have been obvious over Hayes in view of Sprigg.

Sprigg discloses a computer network where a wireless device **300** contains “applications **310** each with a permission list **320**” for controlling application access. (Sprigg, 5:41-43, 5:45-48, 5:63-66, 4:14-21.) Thus, Sprigg discloses that “**the owner control information includes an authorization record.**” As explained above, it would have been obvious to modify Hayes’s administrator interface program (*e.g.*, Figs. 13, 17, and 24) in light of Sprigg to allow an administrator to provide application resource restrictions. (Polish, ¶¶344-348.) This would have led to the server providing a client device “**owner control information**” including “**an authorization record and an application identifier for each of the applications.**” (*Id.*, ¶¶349-350.) Hayes discloses that a user terminal receives from the server a “list of the software to which the user has access permission” including “a user friendly name, the applet’s URL, the URL of an icon for the applet, etc. (information that is required for the desktop to represent the applet on the desktop and to load and launch it).” (Hayes, [0071].) This information includes various “**application identifier[s]**”—including the name and URL of the applet. (*Id.*; Sprigg, 5:25-28, Fig. 3; Polish, ¶351.) In the combination, the “**application identifier**” information would be paired with an “**authorization record**”—*e.g.*, the resource access restriction information disclosed by Sprigg. (Sprigg, 5:41-48, 5:63-66; Polish,

¶351.) The claimed **“through a wireless communication channel”** would also have been obvious based on Sprigg, which discloses the use of “wireless devices” and “a wireless RF network.” (Sprigg, 4:46-5:53, Figs. 1-3; Polish, ¶343.) Motivations to combine are above. (Polish, ¶¶294-301.)

F. Claim 6: “The method of claim 5, wherein the authorization record related to determining whether access to the local memory is allowed is provided by an external computer that is used to enforce policies for operating electronic devices within an organization.”

Claim 6 is obvious over Hayes and Sprigg (Grounds 3-4). As explained above in claim 1[e], it would have been obvious for Hayes’s system to provide **“owner control information”** including an **“authorization record related to determining whether access to the local memory is allowed.”** Sprigg discloses providing permission lists for applications that can control access to “memory, including primary and secondary storage” and “files stored in memory, including the read, write and modify operations on such files.” (Sprigg, 4:14-21.) It would have been obvious for the server in Hayes’s system to provide such permission lists per application, thereby providing an **“authorization record related to determining whether access to the local memory is allowed.”** (Polish, ¶354.) Hayes discloses that the **“owner control information”** is **“provided by an external computer.”** (*Id.*, ¶355.) Hayes discloses that a server (including servlet **214**) provides the owner control information. (Hayes, [0071], Fig. 8.) Thus, Hayes discloses that the **“owner control information is provided by an external computer”** (*e.g.*, a server).

Hayes discloses that the external computer is “**used to enforce policies for operating electronic devices within an organization.**” (Polish, ¶356.)

Hayes discloses enforcing policies on a “corporate intranet.” (Hayes, [0001], [0002], [0059]; Polish, ¶¶294-301.)

G. Claim 7: “The method of claim 1, wherein an update to the owner control information is provided by an external source for storage on the electronic device.”

Claim 7 is disclosed by Hayes (Grounds 3-4). As explained for claim 1[b], Hayes teaches the claimed “**owner control information.**” (Polish, ¶271.) Hayes discloses when a user logs onto the system, the server is checked for the “**list of applications permitted for installation,**” and updated information is sent to the user’s terminal, which is used to build the desktop. (Hayes, Abstract, [0012], [0004].) Hayes discloses that a user terminal “reads its preference information out of its ProfileManagementProperties object P, and begins to update the desktop accordingly.” (*Id.*, [0076]; Polish, ¶360.) Hayes discloses that a server (including servlet **214**) provides the owner control information. (Hayes, [0071], Fig. 8.) Thus, Hayes discloses that an “**update to the owner control information is provided by an external source**” (*e.g.*, a server). (Polish, ¶361; *see supra* claims 5 and 6.)

Hayes discloses that the update to the owner control information is “**for storage on the electronic device.**” (Polish, ¶362.) Hayes discloses that the

owner control information is “stored in the ProfileManagementProperties object P, and returned to the desktop object” on the user terminal. (Hayes, [0071], Abstract, [0012], Figs. 7-8.)

H. Claim 8

1. **“The method of claim 1, wherein a plurality of authorization records associated with a plurality of application identifiers are stored on the electronic device, the method further comprising:” (Claim 8[p])**

To the extent limiting, claim 8[p] is obvious over Hayes and Sprigg (Grounds 3-4). With respect to **“a plurality of authorization records associated with a plurality of application identifiers are stored on the electronic device,”** this limitation is rendered obvious by Hayes and Sprigg as discussed for claim 5. (Polish, ¶¶364-365, 340-351.) Hayes discloses providing a list of applications permitted for installation to user terminals (including **“a plurality of application identifiers”**), and it would have been obvious in light of Sprigg to provide these along with application permissions associated with each application (**“a plurality of authorization records”**). (Hayes, Figs. 13, 24, 17, [0085], [0071], [0013], claim 1, Abstract, [0012], Figs. 7-8; Sprigg, 5:41-43, 5:45-48, 5:63-66, 4:14-21, 5:25-28, Fig. 3, 5:32-35, 6:66-7:5, 6:10-12; Polish, ¶¶365-373, 341-351.)

2. **“receiving requests to access the local memory from a plurality of applications operating on the electronic device;” (Claim 8[a])**

As discussed for claim 1[d], this element is obvious over Hayes and Sprigg

(Grounds 3-4). Claim 8[a] additionally requires receiving requests from “**a plurality of applications**,” which Sprigg discloses. (Sprigg, 8:27-31 (“user privilege level” impacts “the access to resources across all applications the user executes”), 1:13-15, 5:41-45, 9:31-60; Polish, ¶¶375-376.)

3. “**determining whether access to the local memory is allowed to be performed by their respective applications based upon the stored authorization records and the application identifiers that are respectively associated with the plurality of applications; and:**” (Claim 8[b])

As explained for claims 1[e] and 6, Sprigg discloses determining local memory access based on policy rules, which (as discussed for claim 8[p]) constitute “**stored authorization records and the application identifiers that are respectively associated with the plurality of applications.**” Claim 8[b] is obvious based on Hayes in light of Sprigg (Grounds 3-4). (Polish, ¶¶378-379, 321-323.)

4. “**controlling access by the plurality of applications to local memory based upon whether access to the local memory is allowed.**” (Claim 8[c])

As discussed for claim 1[f], claim 8[c] is obvious over Hayes and Sprigg (Grounds 3-4). Claim 8[c] additionally requires controlling access by “**the plurality of applications.**” As discussed for claim 1[f], Sprigg discloses controlling various applications (“**plurality of applications**”). (Sprigg, 8:27-31 (“user privilege level” impacts “the access to resources across all applications the user executes”), 1:13-15, 5:41-45, 9:31-60; Polish, ¶¶381-382, 324-327.)

I. Claim 9: “The method of claim 8, wherein a group containing two or more of the applications is associated with one of the authorization records.”

Claim 9 is obvious over Hayes, Sprigg, and Lambert (EX1004) (Ground 5). As discussed above, the combination of Hayes and Sprigg involves a system in which an administrator can set application access restrictions per application. (Polish, ¶¶385, 293-295.) It would have been obvious based on Lambert to share access restrictions across applications, such that **“a group containing two or more of the applications is associated with one of the authorization records.”** (*Id.*, ¶¶385-396.) Like Hayes and Sprigg, Lambert discloses restricting application access to device resources. (Lambert, [0064]-[0065], [0068]-[0071], [0080], [0082]; Polish, ¶386.)

As discussed above for claim 9 for ground 1, Lambert discloses various examples of **“a group containing two or more of the applications”** having the same access restrictions, including filepath rules, zone rules, and reusable code authorization levels. (Lambert, [0058]-[0063], [0079], [0082], [0068], [0066], Fig. 7; Polish, ¶¶386-389, 197-199.)

It would have been obvious based on Lambert to implement the combined Hayes and Sprigg system such that **“a group containing two or more of the applications is associated with one of the authorization records”**—*i.e.*, multiple applications would be associated with the same access restriction level. (Polish,

Motivation to Combine: Hayes, Sprigg, and Lambert are analogous art to the '571 patent for the same reasons as described above. (Polish, ¶391.) Each relates to limiting corporate application use by users. (Hayes, [0001], [0059], [0002]; Sprigg, 1:13-15, 1:20-24, 3:36-41; Lambert, [0078], [0002], [0080]-[0081], [0003], [0007].) A POSITA would have been motivated to share access restrictions across applications to simplify policy input and adjustment. (Lambert, [0007]; Polish, ¶¶392, 394.) A POSITA would have been further motivated to facilitate more comprehensive access restriction policies that cover more applications. (*E.g.*, Sprigg, 1:31-36; Polish, ¶393.) A POSITA would have had more than a reasonable expectation of success and could have implemented the proposed combination using conventional techniques. (Sprigg, 1:61-2:2; Polish, ¶395.)

J. Claim 10

1. “A device, comprising:” (Claim 10[p])

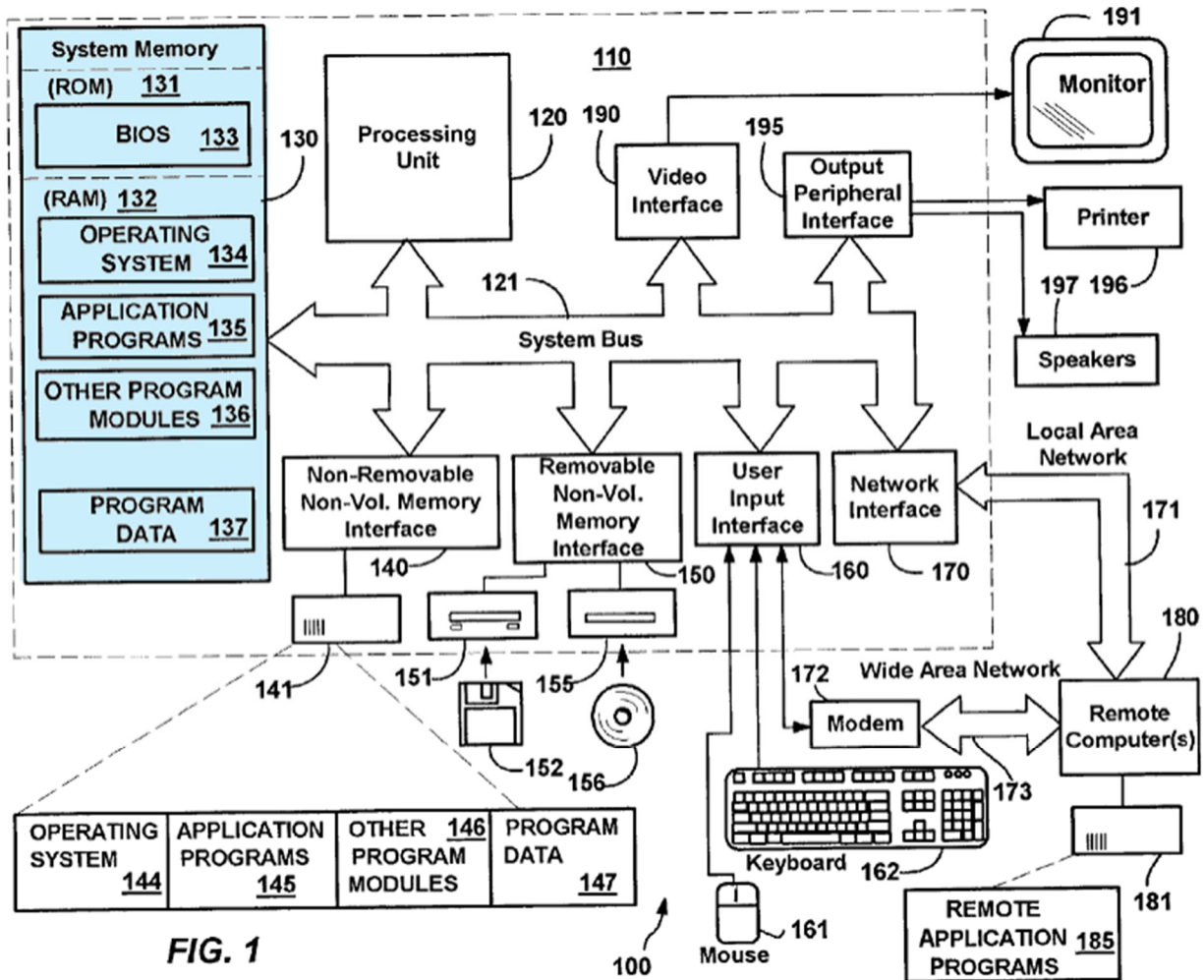
To the extent limiting, claim 10[p] is disclosed by Hayes (Grounds 3-5). Hayes teaches “user stations,” including “personal computers.” (Hayes, [0037], Fig. 1, [0011]; Polish, ¶398; *see supra* claim 1[p].) Hayes thus discloses “**a device.**”

2. “a memory; and” (Claim 10[a])

Claim 10[a] is obvious over Hayes (Grounds 3-4). The devices identified for claim 10[p] would have contained a “**memory.**” (Hayes, [0015], [0013], [0037], Fig. 1; Polish, ¶400; EX1016 (Ibrahim), p.008 (“memory”).)

Claim 10[a] is also obvious over Hayes and **Lambert** (EX1004; Ground 5).

Lambert discloses “system memory 131”:



(Lambert, Fig. 1 (highlighting added), [0031]; Polish, ¶¶402-403.)

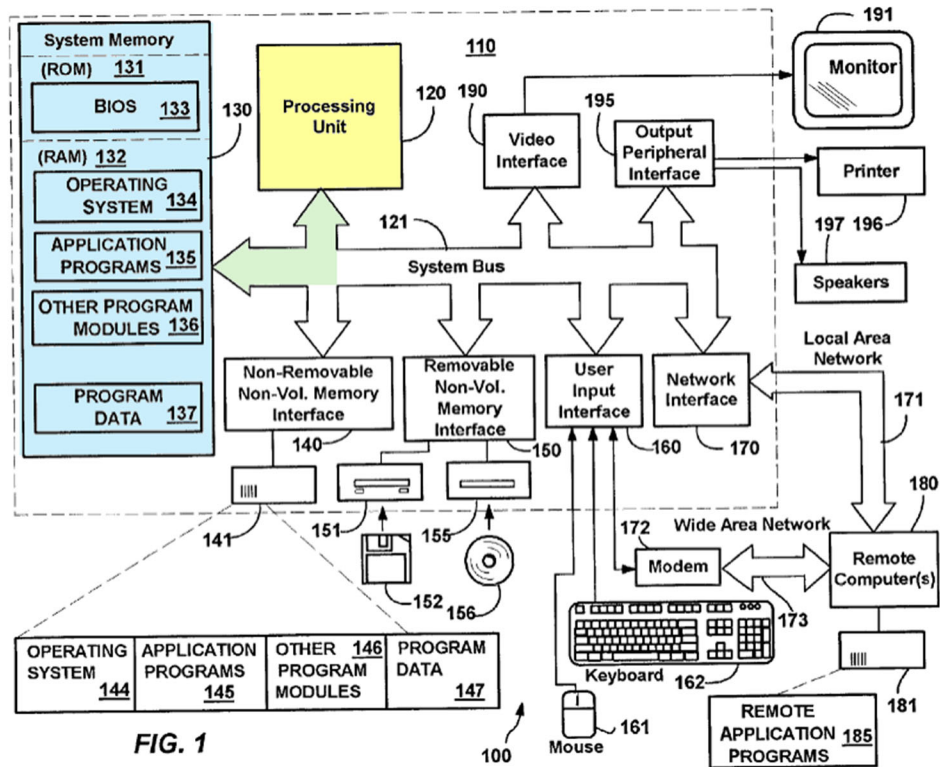
Motivation to Combine: Hayes and Lambert are analogous art to the '571 patent for the same reasons described above. (Polish, ¶¶404, 407, 302.) It would have been obvious for a POSITA to combine Hayes with Lambert. (*Id.*, ¶¶403, 390-396.) A POSITA would have looked to Lambert for hardware implementation details. (*Id.*, ¶¶403, 405; Lambert, [0029]-[0035], Fig. 1.) A

POSITA would have had more than a reasonable expectation of success and could have implemented the proposed combination using conventional techniques. (Polish, ¶406; EX1016, p.008, Fig. 12.1.)

3. “a processor coupled with the memory and configured by instructions stored in the memory to:” (Claim 10[b])

Claim 10[b] is obvious over Hayes (Grounds 3-4). The devices identified for claim 10[p] would have contained a “**processor coupled with the memory and configured by instructions stored in the memory.**” (Hayes, [0015], [0013], [0037], Fig. 1; Polish, ¶¶408-409; EX1016, p.008.)

Claim 10[b] is also obvious over Hayes and Lambert (EX1004; Ground 5). As explained above, Lambert teaches “**a processor coupled with the memory and configured by instructions stored in the memory.**”



(Lambert, Fig. 1 (highlighting added), [0031]; Polish, ¶¶409, 205.) As explained above, it would have been obvious to combine Hayes with Lambert. (Polish, ¶¶410, 390-396.)

4. “receive owner information from an owner of the device;” (Claim 10[b][i]) and “receive owner control information that comprises:” (Claim 10[b][ii]); “a list of applications permitted for installation on the device, and” (Claim 10[b][ii][A]); “permissions associated with each permitted application in the list of applications;” (Claim 10[b][ii][B]); “store the owner control information in a protected data store to prevent alteration or deletion of the owner control information in response to authenticating the owner control information using the owner information” (Claim 10[b][iii]); “receive, from an application operating on the electronic device, a request to access local memory on the electronic device;” (Claim 10[b][iv]); “determine whether the application is allowed to access the local memory in response to identifying a permission associated with the application in the owner control information; and” (Claim 10[b][v]); “control application access to the local memory based upon the determining.” (Claim 10[b][vi])

These elements are obvious for the same reasons explained for claims 1[a]-1[f]. (Polish, ¶¶411-417, 259-327; EX1020.)

- K. **Claim 11:** “The device of claim 10, wherein the request to access the local memory comprises a request to erase data from the local memory.”

This element is obvious for the same reasons explained for claim 2. (Polish, ¶¶420, 328-331; EX1020.)

- L. **Claim 12:** “The device of claim 10, wherein the request to access the local memory comprises a request to read data from the local memory.”

This element is obvious for the same reasons explained for claim 3. (Polish, ¶¶422, 332-335; EX1020.)

M. Claim 13: “The device of claim 10, wherein the processor is further configured to restrict modification of the owner control information after the owner control information has been stored on the electronic device.”

Claim 13 is substantially the same as claim 4 (Grounds 3-5). As discussed for claim 4, Sprigg discloses that its system checks whether permission information was modified before executing an application being checked against the permission information. (Sprigg, Fig. 4, 4:33-37, 5:25-38, 6:22-33, 6:51-7:23, 1:45-52.) If it is determined that the permission information was modified (*e.g.*, digital signature), then Sprigg discloses preventing the application from executing. (*Id.*) Thus, it would have been obvious that after a user terminal receives Sprigg’s application permission information, a user would be restricted from modifying that information. (Polish, ¶¶425-426.) In the combined system, “**the processor is further configured to restrict modification** [*i.e.*, prevent modification without consequences to the execution of applications] **of the owner control information after the owner control information has been stored on the electronic device.**” (*Id.*, ¶427.) Claim 13 would have been obvious over Hayes, Sprigg, and Lambert (per claim 10).

N. Claim 14: “The device of claim 10, wherein the owner control information includes an authorization record and an application identifier for each of the applications, and wherein the owner control information is provided by an external source through a wireless communication channel.”

This element is obvious for the same reasons explained for claim 5. (Polish, ¶¶429, 340-351; EX1020.)

- O. Claim 15: “The device of claim 14, wherein the authorization record related to determining whether access to the local memory is allowed is provided by an external computer that is used to enforce policies for operating electronic devices within an organization.”**

This element is obvious for the same reasons explained for claim 6.

(Polish, ¶¶431, 352-357; EX1020.)

- P. Claim 16: “The device of claim 10, wherein an update to the owner control information is provided by an external source for storage on the electronic device.”**

This element is obvious for the same reasons explained for claim 7.

(Polish, ¶¶433, 358-363; EX1020.)

- Q. Claim 17**

To the extent limiting, claim 17[p] is obvious over Hayes (Grounds 3-4). Hayes discloses a set of user stations connected to a network, such as desktop personal computers. (Hayes, [0015], [0013], [0037], Fig. 1.) These computers would have contained a **“non-transitory computer-readable medium with instructions stored thereon, wherein the instructions are executable by a processor to cause the processor to perform the [claimed] operations”**. (Polish, ¶435; EX1016, p.008.) Claim 17[p] is also obvious over Hayes and Lambert for the reasons explained above for claims 10[a]-10[b] (Ground 5). (Polish, ¶¶436, 410, 390-396, 402-403; Lambert, [0031].)

For claims 17[a]-17[f], these elements are obvious for the same reasons explained for claims 1[a]-1[f]. (Polish, ¶¶437-444, 359-327; EX1020.)

R. Claim 18

This element is obvious for the same reasons explained for claims 8[p], 1[b][ii], and 5. (Polish, ¶¶446-454, 270-289; EX1020.)

S. Claim 19

This element is obvious for the same reasons explained for claim 2. (Polish, ¶¶456, 328-331; EX1020.)

T. Claim 20

This element is obvious for the same reasons explained for claim 3. (Polish, ¶¶458, 332-335; EX1020.)

IX. NO SECONDARY CONSIDERATIONS OF NONOBVIOUSNESS

PO has not identified any alleged objective evidence of non-obviousness, nor is Petitioner aware of any.

X. CONCLUSION

Petitioner requests institution.

Dated: October 9, 2025

Respectfully submitted,

COOLEY LLP
ATTN: Patent Group
1299 Pennsylvania Ave. NW
Suite 700
Washington, DC 20004
Tel: (650) 843-5001
Fax: (650) 849-7400

By: /s/ Matthew J. Brigham
Matthew J. Brigham
Reg. No. 44,047
Counsel for Petitioner

CERTIFICATE OF COMPLIANCE WITH WORD COUNT

Pursuant to 37 C.F.R. § 42.24(d), I certify that this petition complies with the type-volume limits of 37 C.F.R. § 42.24(a)(1)(i) because it contains 13,756 words, according to the word-processing system used to prepare this petition, excluding the parts of this petition that are exempted by 37 C.F.R. § 42.24(a) (including the table of contents, a table of authorities, mandatory notices, a certificate of service or this certificate word count, appendix of exhibits, and claim listings).

DATED: October 9, 2025

COOLEY LLP

ATTN: Patent Docketing

1299 Pennsylvania Ave. NW Suite
700

Washington, D.C. 20004

Tel: (650) 843-5001

Fax: (650) 849-7400

/s/ Matthew J. Brigham

Matthew J. Brigham

Reg. No. 44,047

CERTIFICATE OF SERVICE

I hereby certify, pursuant to 37 C.F.R. Sections 42.6 and 42.105, that a complete copy of the attached **PETITION FOR INTER PARTES REVIEW OF U.S. PATENT NO. 9,542,571 B2**, including all exhibits (**Nos. 1001-1020**) and related documents, are being served via Federal Express on October 9, 2025, the same day as the filing of the above-identified document in the United States Patent and Trademark Office/Patent Trial and Appeal Board, upon Patent Owner by serving the correspondence address of record with the USPTO as follows:

82313 - CONLEY ROSE - BLACKBERRY FILES
ATTN: J. ROBERT BROWN
4965 PRESTON PARK BLVD, SUITE 195E
PLANO, TX 75093

And via Federal Express upon the following counsel for Patent Owner in *Malikie Innovations Ltd. v. Nintendo Co.* in the United States District Court, Western District of Washington, Case No. 2:24-cv-01490-JLR as follows:

Roger M. Townsend
TOWNSEND LEGAL PLLC
380 Winslow Way, Ste 200
Bainbridge Island, WA 98110

Khue Hoang
Patrick Colsher
REICHMAN JORGENSEN LEHMAN & FELBERG LLP
400 Madison Avenue, Suite 14D
New York, NY 10017

Matthew G. Berkowitz
Jeff Leung
Michael M. Polka
REICHMAN JORGENSEN LEHMAN & FELDBERG LLP (CA)
100 Marine Pkwy, Ste 300
Redwood Shores, CA 94065

Taylor N. Mauze
REICHMAN JORGENSEN LEHMAN & FELBERG LLP
515 Congress Avenue, Suite 1900
Austin, TX 78701

Dated: October 9, 2025

/s/ Matthew J. Brigham

Matthew J. Brigham