

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

AMAZON.COM, INC.,
Petitioner,

v.

KAIFI LLC,
Patent Owner.

IPR2025-00626
U.S. Patent No. 11,082,518

**PETITION FOR *INTER PARTES* REVIEW
UNDER 35 U.S.C. § 312 AND 37 C.F.R. § 42.104**

TABLE OF CONTENTS

PETITIONER’S EXHIBIT LIST	5
I. INTRODUCTION	8
II. GROUNDS FOR STANDING.....	9
III. NOTE.....	9
IV. THE ’518 PATENT.....	9
A. Technical Background.....	9
B. Overview	9
V. PROSECUTION HISTORY	11
VI. LEVEL OF ORDINARY SKILL IN THE ART.....	12
VII. CLAIM CONSTRUCTION	13
VIII. RELIEF REQUESTED AND THE REASONS FOR THE REQUESTED RELIEF	14
IX. IDENTIFICATION OF HOW THE CLAIMS ARE UNPATENTABLE....	14
A. Challenged Claims	14
B. Statutory Grounds for Challenges	14
C. Ground 1: Claims 1-20 are obvious under §103 over Ermis.	15
1. Ermis	15
2. Claim 1	19
3. Claim 2.....	46
4. Claim 3	50

5.	Claim 4.....	52
6.	Claim 5.....	56
7.	Claim 6.....	62
8.	Claim 7.....	64
9.	Claim 8.....	67
10.	Claim 9.....	69
11.	Claim 10.....	70
12.	Claim 11.....	73
13.	Claim 12.....	76
14.	Claim 13.....	77
15.	Claim 14.....	79
16.	Claim 15.....	80
17.	Claim 16.....	81
18.	Claim 17.....	84
19.	Claim 18.....	85
20.	Claim 19.....	88
21.	Claim 20.....	93
D.	Ground 2: Claims 1-20 are obvious under §103 over Ermis in view of Nakano.	95
1.	Nakano	95
2.	Reasons to Combine Ermis and Nakano.....	98

3.	Claim 1	103
4.	Claims 2-14	107
5.	Claim 15	107
6.	Claims 16-20	110
X.	DISCRETIONARY DENIAL WOULD BE INAPPROPRIATE.....	110
A.	No basis for <i>Fintiv</i> denial	110
B.	No basis for §325(d) denial	112
XI.	CONCLUSION.....	112
XII.	MANDATORY NOTICES	113
A.	Real Party-in-Interest	113
B.	Related Matters.....	113
C.	Lead and Back-up Counsel and Service Information	113
	CERTIFICATE OF WORD COUNT.....	115
	CERTIFICATE OF SERVICE	116

PETITIONER’S EXHIBIT LIST

Ex.1001	U.S. Patent No. 11,082,518
Ex.1002	Prosecution History of U.S. Patent No. 11,082,518
Ex.1003	Declaration of Henry Houh Ph.D. under 37 C.F.R. §1.68
Ex.1004	<i>Curriculum Vitae</i> of Henry Houh Ph.D.
Ex.1005	U.S. Patent Application Pub. 2012/0079091 to Ermis et al. (“Ermis”)
Ex.1006	U.S. Patent Application Pub. 2004/0131079 to Hedge et al. (“Hedge”)
Ex.1007	U.S. Patent Application Pub. 2003/0177283 to Hamilton et al. (“Hamilton”)
Ex.1008	U.S. Patent Application Pub. 2008/0256081 to Bui et al. (“Bui”)
Ex.1009	U.S. Patent Application Pub. 2004/0243580 to Markki et al. (“Markki”)
Ex.1010	U.S. Patent Application Pub. 2006/0070004 to Miller et al. (“Miller”)
Ex.1011	U.S. Patent Application Pub. 2007/0036461 to Atherton et al. (“Atherton”)
Ex.1012	Reserved
Ex.1013	U.S. Patent Application Pub. 2014/0359283 to Lee et al. (“Lee”)
Ex.1014	U.S. Patent No. 8,914,360 to Gailloux et al. (“Gailloux”)
Ex.1015	U.S. Patent Application Pub. 2014/0365964 to Forsblom et al. (“Forsblom”)
Ex.1016	U.S. Patent Application Pub. 2014/0325081 to Heinrich et al. (“Heinrich”)

Ex.1017	KAIFI's Complaint, <i>KAIFI LLC v. Amazon.com, Inc.</i> , No. 2:24-cv-00542 (EDTX)
Ex.1018	Docket Control Order, <i>KAIFI LLC v. Amazon.com, Inc.</i> , No. 2:24-cv-00542 (EDTX)
Ex.1019	United States District Court – National Judicial Caseload Profile (Dec 2024)
Ex.1020	Interim Procedure for Discretionary Denials in AIA Parallel District Court Litigation, June 21, 2022
Ex.1021	Kaifi's Infringement Contentions, including Appendix D (Claim Chart for U.S. Patent No. 11,082,518)
Ex.1022	Reserved
Ex.1023	U.S. Patent No. 9,720,391 to Nakano et al. ("Nakano")
Ex.1024	U.S. Patent No. 5,526,523 to Straub et al. ("Straub")
Ex.1025	U.S. Patent No. 5,951,639 to MacInnis ("MacInnis")
Ex.1026	U.S. Patent Application Pub. 2003/0177285 to Hunt et al. ("Hunt")
Ex.1027	U.S. Patent Application Pub. 2012/0291068 to Khushoo et al. ("Khushoo")
Ex.1028	Harper, Richard, "Inside the Smart Home," Springer, 2003 ("Harper")
Ex.1029	U.S. Patent Application Pub. 2016/0085222 to Yabe et al. ("Yabe")
Ex.1030	Schaefer, Robbie et al. "Profile Processing and Evolution for Smart Environments," 2006 ("Schaefer")
Ex.1031	U.S. Patent No. RE44,534 to Li et al. ("Li")
Ex.1032	U.S. Patent No. 8,868,424 to Moore et al. ("Moore")
Ex.1033	U.S. Patent Application Pub. 2014/0236630 to Murata ("Murata")

Ex.1034	U.S. Patent Application Pub. 2013/0041793 to Kashima et al. (“Kashima”)
Ex.1035	U.S. Patent Application Pub. 2014/0143419 to Vyatkin et al. (“Vyatkin”)
Ex.1036	U.S. Patent Application Pub. 2009/0319683 to Martin (“Martin”), Ex.1036
Ex.1037	U.S. Patent Application Pub. 2014/0040182 to Gilder et al. (“Gilder”)
Ex.1038	U.S. Patent Application Pub. 2005/0172280 to Ziegler et al. (“Ziegler”)
Ex.1039	U.S. Patent Application Pub. 2005/0066325 to Mori et al. (“Mori”)
Ex.1040	Docket for Judge Rodney Gilstrap, Eastern District of Texas, March 16, 2026.
Ex.1041	KAIFI’s Infringement Contentions, Cover Pleadings

I. INTRODUCTION

U.S. Patent No. 11,082,518 (the “’518 patent,” Ex.1001) describes multi-device home automation. “[U]sing a smart phone,” a user sends a command requesting that a task be performed. Ex.1001, 8:39-41, Fig. 2, 7:62-10:54 (providing movie recommendations, making a reservation, or operating a boiler). A “relation server” controls various machines to “execute [the] command.” Ex.1001, 1:23-29.

As of the claimed priority date of the ’518 patent (November 3, 2014), there was nothing novel about these features. On September 27, 2011, more than three years earlier, Ericsson filed a U.S. patent application (U.S. Patent Publication No. 2012/0079091 to Ermis et al. (“Ermis,” Ex.1005)) describing the same concepts of multi-device home automation. Ermis described a system 12 that received a task request from a user and controlled multiple “networked objects 10” to perform the requested task. *See* Ex.1005, Fig. 11. Ermis provided a host of examples of these tasks, like automatically recording a TV program, turning lights off, and starting a washing machine. *See* Ex.1005, Figs. 16-18, 23, [0045]-[0046], [0110]-[0113], [0125].

Pursuant to §§311, 314(a), and 37 C.F.R. §42.100, Amazon.com, Inc. (“Petitioner”) respectfully requests that the Board review and cancel claims 1-20 (the “Challenged Claims”) as unpatentable.

II. GROUNDS FOR STANDING

Petitioner certifies that the '518 patent is eligible for IPR and Petitioner is not barred or estopped from requesting IPR challenging the claims. 37 C.F.R. §42.104(a).

III. NOTE

Petitioner cites to exhibits' original page numbers. **Emphasis** in quoted material has been added. Claim terms are presented in *italics*.

IV. THE '518 PATENT

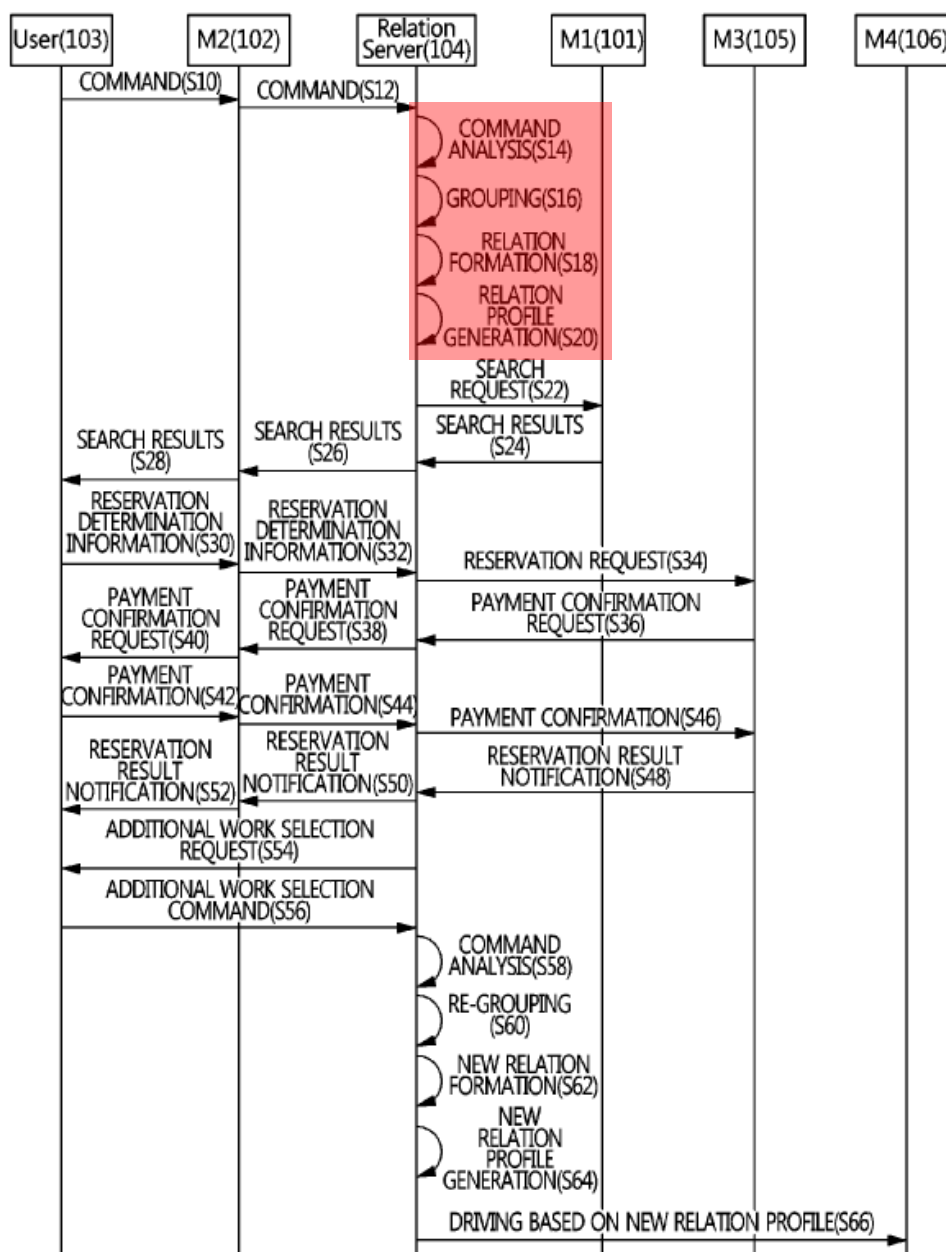
A. Technical Background

The '518 patent describes automating tasks performed by multiple devices using a “relation server.” Ex.1001, (54) (Title), Abstract. Dr. Houh's declaration describes the relevant background technology predating the '518 patent, cited as appropriate below. *See* Ex.1003, ¶¶22-30.

B. Overview

The relation server of the '518 patent “manag[es] relations between machines” to “execute a command.” Ex.1001, Abstract. The relation server receives a command from a user, determines which machines are necessary to execute the command, and controls those machines to execute the command. The '518 patent provides an example “scenario” with Figure 2, below Ex.1001, 8:62-9:3.

In response to a user command, the relation server “generate[s] a capability set required to execute the command,” and “group[s] the machines 101, 102, 105, and 106 which will execute the command.” Ex.1001, 9:13-18.



Ex.1001, Fig. 2 (annotated)

The relation server then transmits commands to the selected machines to perform the requested task. Ex.1001, Fig. 2 (steps S22-S66).

The user provides another command “to make a reservation (e.g. the selected movie, the selected movie theater, the movie start time, the number of tickets to be reserved, etc.).” Ex.1001, 9:63-10:3.

A “task processing schedule” parameter defines the processes performed by devices to execute the command. Ex.1001, 6:58-7:27 (Table 2).

The example “parameters” and “sub-parameters” in the ’518 patent describe information about users, machines, values or ID numbers. Ex.1001, 4:32-58 (Table 1), 6:58-7:27 (Table 2).

These concepts were well known before the priority date of the ’518 patent.

V. PROSECUTION HISTORY

The ’518 patent issued from U.S. Application 16/894,167 filed on June 5, 2020. Ex.1001, (21), (22). The ’518 patent is a continuation of U.S. Patent No. 10,701,169 which claims priority to multiple Korean applications with an earliest filing date of November 3, 2014. Ex.1001, (63), 1:7-17. The prior art relied upon pre-dates this earliest claimed priority date.

The ’518 patent was originally filed with a single claim. Ex.1002, 42. After one rejection, the applicant replaced the claim with new claims 2-21. Ex.1002, 98-101.

The Patent Office then allowed claims 2-21. Ex.1002, 115-122. The Notice of Allowance cited limitations [1.2.2], [1.2.3], [1.2.4], [1.3], and [1.4] as allowable subject matter:

[N]one of the references...suggest the combination of limitations specified in the independent claims: “storing a relation profile [1.2.2] including a task processing schedule parameter which defines a sequence of performing multiple processes [1.2.3]...requesting selected machines of the multiple machines to perform the multiple processes [1.2.4], wherein...generates a new relation profile based on an intervention by a user in the relation profile [1.3], wherein the task processing schedule parameter configures a start time of at least one process of the multiple processes [1.4].”

Ex.1002, 121.

As illustrated below, Ermis and Nakano render obvious each limitation. The Examiner erred in allowing the '518 patent.

VI. LEVEL OF ORDINARY SKILL IN THE ART

A Person of Ordinary Skill in the Art (“POSITA”) in November 2014 would have been someone knowledgeable and familiar with electronic device communication pertinent to the '518 patent. That person would have a bachelor's degree in electrical engineering, computer engineering, computer science, or equivalent training, and approximately two years of experience working in the field of electronic device communication. Lack of work experience can be

remedied by additional education, and vice versa. Ex.1003, ¶¶14-17.

VII. CLAIM CONSTRUCTION

In an IPR, claims are “construed using the same claim construction standard...in a civil action under 35 U.S.C. 282(b), including construing the claim in accordance with the ordinary and customary meaning of such claim as understood by one of ordinary skill in the art and the prosecution history pertaining to the patent.” 37 C.F.R. §42.100(b). The Board only construes claims to the extent necessary to resolve the underlying controversy. *Nidec Motor Corp. v. Zhongshan Broad Ocean Motor Co.*, 868 F.3d 1013, 1017 (Fed. Cir. 2017).¹ Petitioner submits that, for the purposes of this proceeding and the grounds presented herein, no claim terms require construction. Petitioner notes that the ’518 patent specification provides definitions for multiple claim terms, which are addressed below as needed (“*relation profile*” ([1.2.2]), “*capability set parameter*” ([4.0]), “*sub-parameter*” ([8.0]), “*input parameter sub-parameter*” ([11.1]), “*operating system parameter*” ([15.0]), and “*interface parameter*” ([16.0])). The following grounds demonstrate that these claim terms, as defined, are rendered obvious by the prior art.

¹ Petitioner does not concede that any term in the challenged claims meets the requirements of 35 U.S.C. §112, or that the challenged claims recite patentable subject matter under 35 U.S.C. §101.

VIII. RELIEF REQUESTED AND THE REASONS FOR THE REQUESTED RELIEF

Petitioner asks that the Board institute a trial for *inter partes* review and cancel the Challenged Claims.

IX. IDENTIFICATION OF HOW THE CLAIMS ARE UNPATENTABLE

A. Challenged Claims

Petitioner challenges claims 1-20, which correspond to the claims asserted in the plaintiff's Infringement Contentions in co-pending litigation. Ex.1021, 2.

B. Statutory Grounds for Challenges

Grounds	Claim(s)	Basis
#1	1-20	35 U.S.C. §103 over Ermis
#2	1-20	35 U.S.C. §103 over Ermis in view of Nakano

Ermis (Ex.1005) was published on March 29, 2012, over two years before the priority date of the '518 patent (November 3, 2014). Ex.1005, (43). Ermis is prior art under 35 U.S.C. §102(a)(1) because it was “a printed publication...available to the public before the effective filing date of the” '518 patent.

U.S. Patent No. 9,720,391 to Nakano et al. (“Nakano,” Ex.1023) was filed as Application No. 14/381,312 on August 1, 2013, before the earliest claimed priority date of the '518 patent. Nakano is prior art under 35 U.S.C. §102(a)(2) because it is a “patent or application” that “names another inventor and was effectively filed

before the effective filing date of the” ’518 patent.

Petitioner’s analysis cites additional prior art to demonstrate background knowledge of a POSITA and provide context regarding what a POSITA would have understood from the prior art. *See Yeda Research & Dev. Co. v. Mylan Pharm. Inc.*, 906 F.3d 1031, 1041-42 (Fed. Cir. 2018); 37 C.F.R. §42.104(b); *K/S HIMPP v. Hear-Wear Techs., LLC*, 751 F.3d 1362, 1365-66 (Fed. Cir. 2014).

C. Ground 1: Claims 1-20 are obvious under §103 over Ermis.

1. Ermis

Like the ’518 patent, Ermis describes a system, shown below, that receives commands from a user, identifies the machines necessary to execute the command, and executes commands.

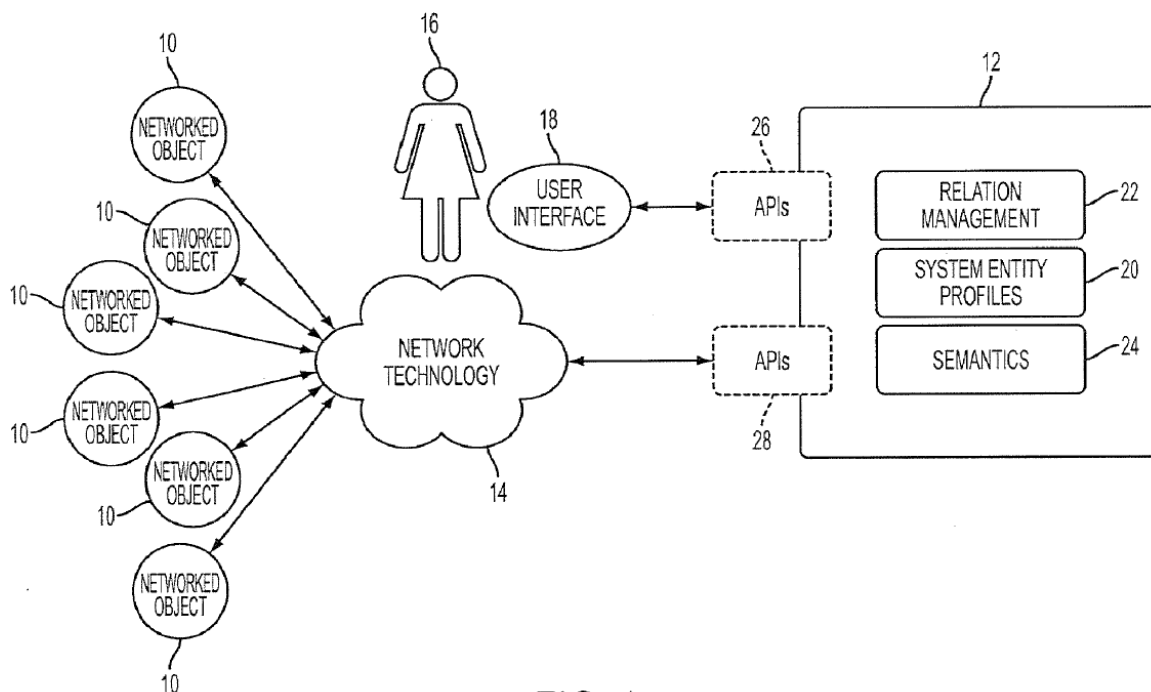


FIG. 1

Ex.1005, Fig. 1

Ermis's system includes a system 12 which "can...be implemented...as server-based software." Ex.1005, [0040]. It communicates with multiple "networked objects 10," including "consumer electronics, digitally tagged objects, computer devices, mobiles, sensors," etc. Ex.1005, [0039]-[0040].

The user 16 sends a task request to the system 12 via a "user interface[] 18," the user's "mobile device[]." Ex.1005, [0039].

After receiving the request, Ermis's system 12 determines which capabilities are needed to execute the command. Ex.1005, [0066] ("[a] service required capabilities database"). Then, the system 12 groups the machines with the required

capabilities and controls those networked objects 10 to execute the command.

Ex.1005, [0045]-[0046], [0101]-[0102].

Ermis provides multiple examples of task requests and explains how the system 12 generates a series of processes to be performed by the networked objects 10 (referred to as a “data flow”) to fulfil the task request. In one example, “the user 16...send[s] a task request...requesting the system 12 to record a certain TV program.” Ex.1005, [0045]. Under direction of the system 12, “[T]he TV will accept and acknowledge this instruction” and a “video recorder 10 will actually execute the job.” Ex.1005, [0045]-[0046].

In another example, the system 12 recommends that the lights be turned off when the user leaves.

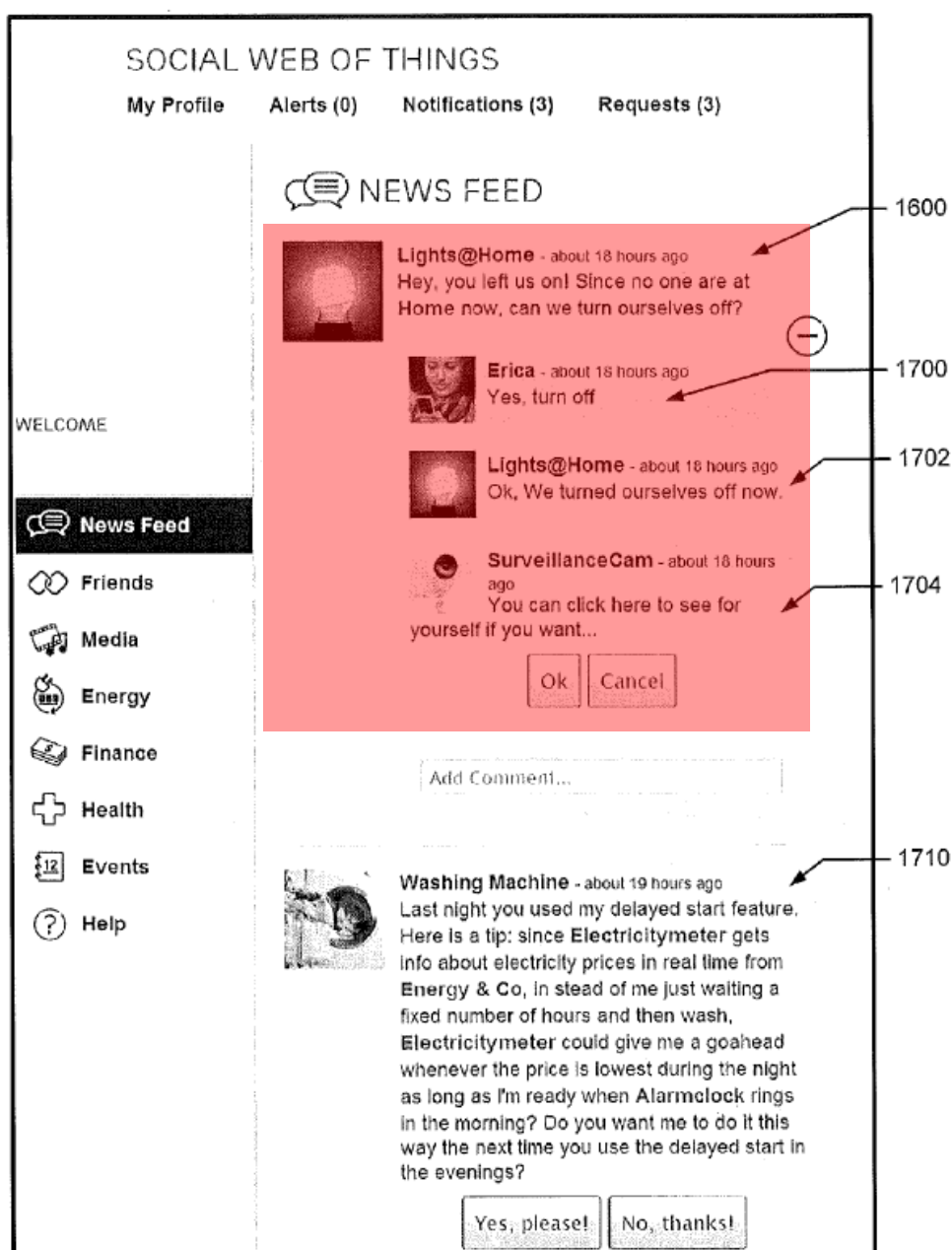


FIG. 17

Ex.1005, Fig. 17 (annotated)

If approved, the system 12 establishes “a data flow...between the light controller and a surveillance camera” allowing the user to “observe a room which

may include the controlled lights.” Ex.1005, [0110].

2. Claim 1

[1.0] *A system, comprising:*

To the extent the preamble is limiting, Ermis renders it obvious. Ex.1003, ¶51.

Ermis describes “systems...for management of, and interaction with, networked objects.” Ex.1005, [0038]. Ermis provides a “high level architectural view of the system...with respect to FIGS. 1 and 2,” shown below. Ex.1005, [0039]; Ex.1003, ¶52.

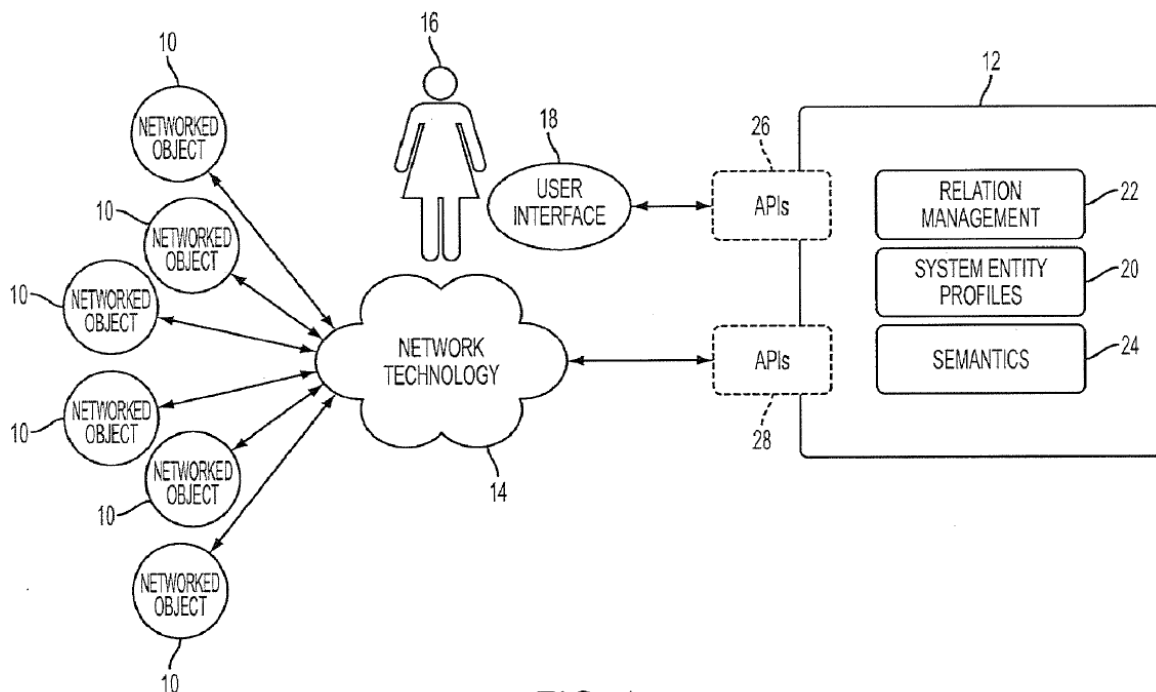
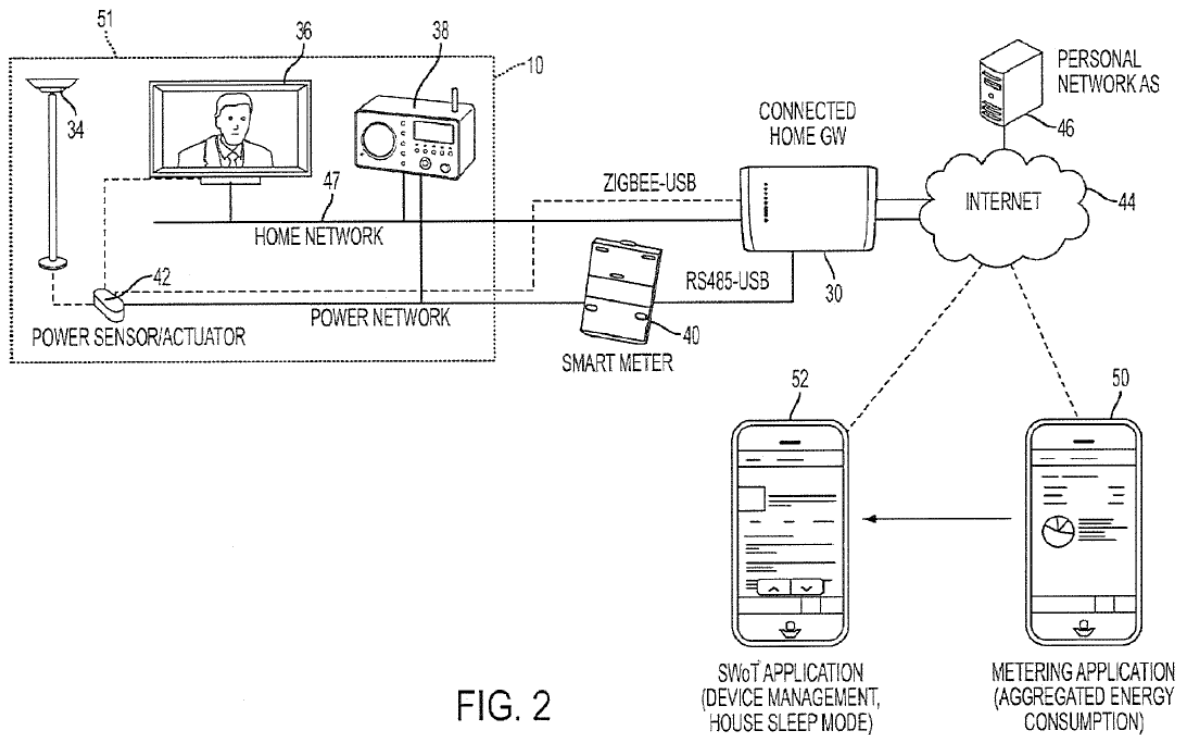


FIG. 1

Ex.1005, Fig. 1



Thus, Ermis renders obvious *a system* (“system[]”). Ex.1003, ¶¶51-53.

[1.1] *multiple machines; and*

Ermis’s system includes multiple networked objects 10 and a user interface 18, shown in Figure 1. Ex.1003, ¶54.

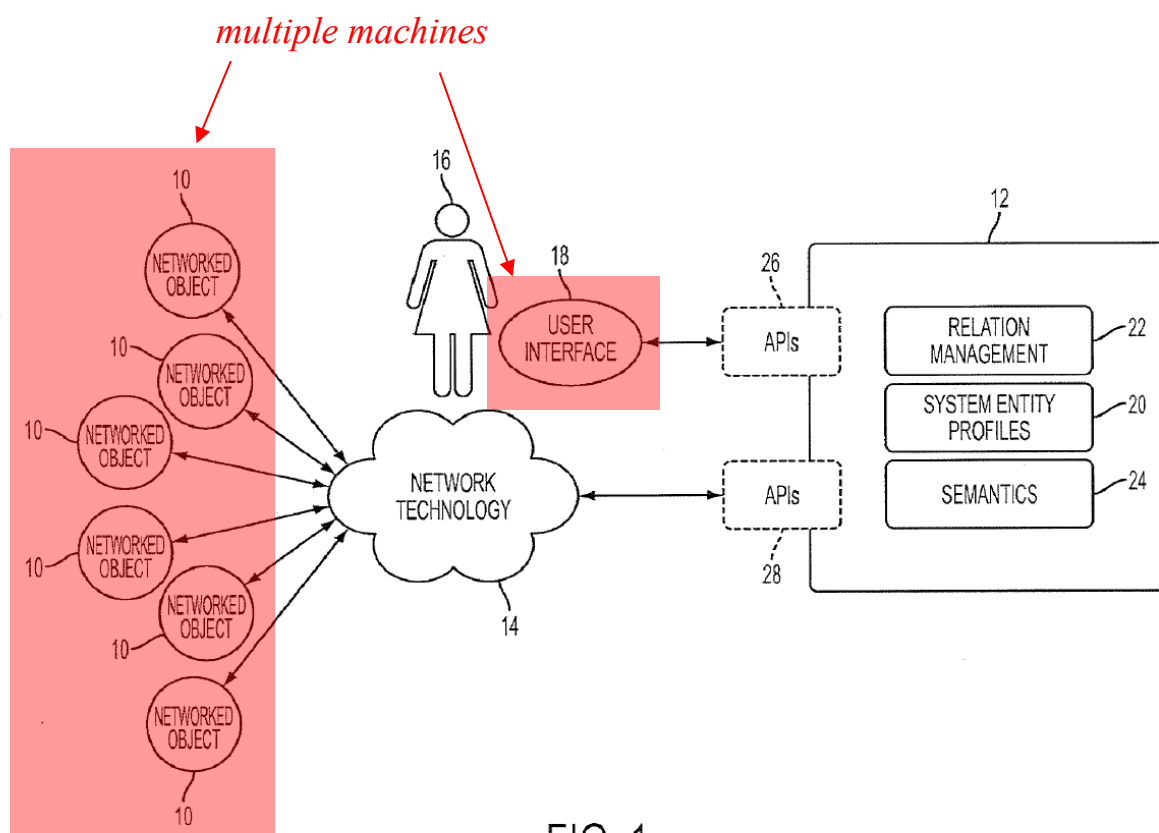


FIG. 1

Ex.1005, Fig. 1 (annotated); Ex.1003, ¶54

In Figure 1, “a plurality of networked objects 10 is connected to the system 12.” Ex.1005, [0039]. “Networked objects 10 include, but are not limited to: consumer electronics, digitally tagged objects, computer devices, mobiles, sensors, buildings, [or] vehicles.” Ex.1005, [0040]. The user interface 18 can be “a mobile phone or other end user terminal device.” Ex.1005, [0050]. These networked objects 10 and the user interface 18 are *machines* as claimed. In another example, Ermis describes “resource nodes 1330” and a “user equipment node 1320” which are also *machines*. Ex.1005, [0097]; Fig. 2; Ex.1003, ¶¶55-57.

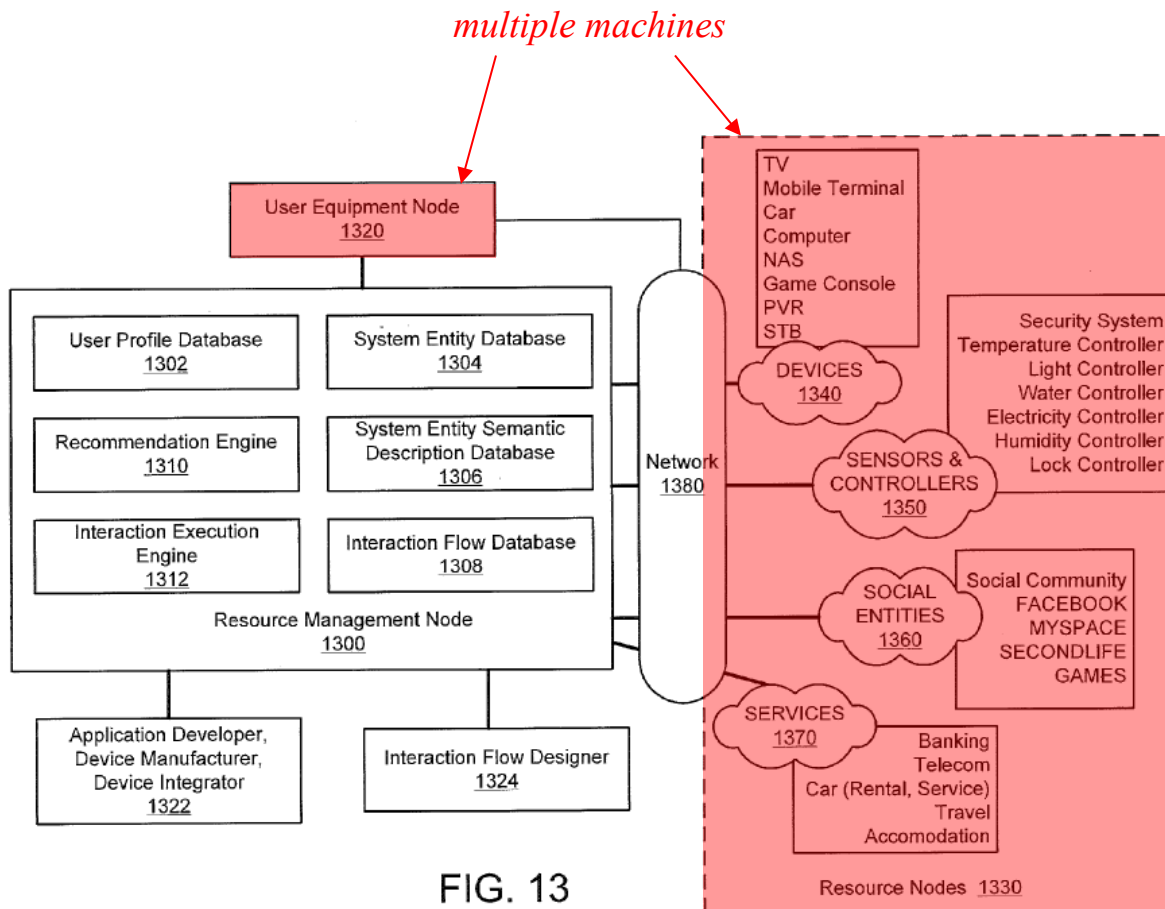


FIG. 13

Ex.1005, Fig. 13 (annotated); Ex.1003, ¶57

Therefore, Ermis renders obvious *multiple machines* (“networked objects 10,” “user interface 18,” “resource nodes 1330,” and/or “user equipment node 1320”). Ex.1003, ¶¶54-58.

[1.2.1] a relation server

In Figure 1, Ermis’s system includes a “system 12 which manages the networked objects 10” and “can...be implemented...as server-based software.” Ex.1005, [0040]. The system 12, alone or in combination with the APIs 26 and 28,

renders obvious a *relation server* because it “includes a **relation management function 22** which **coordinates the interactions between the networked objects** **10.**” Ex.1005, [0042]; Ex.1003, ¶¶59-60.

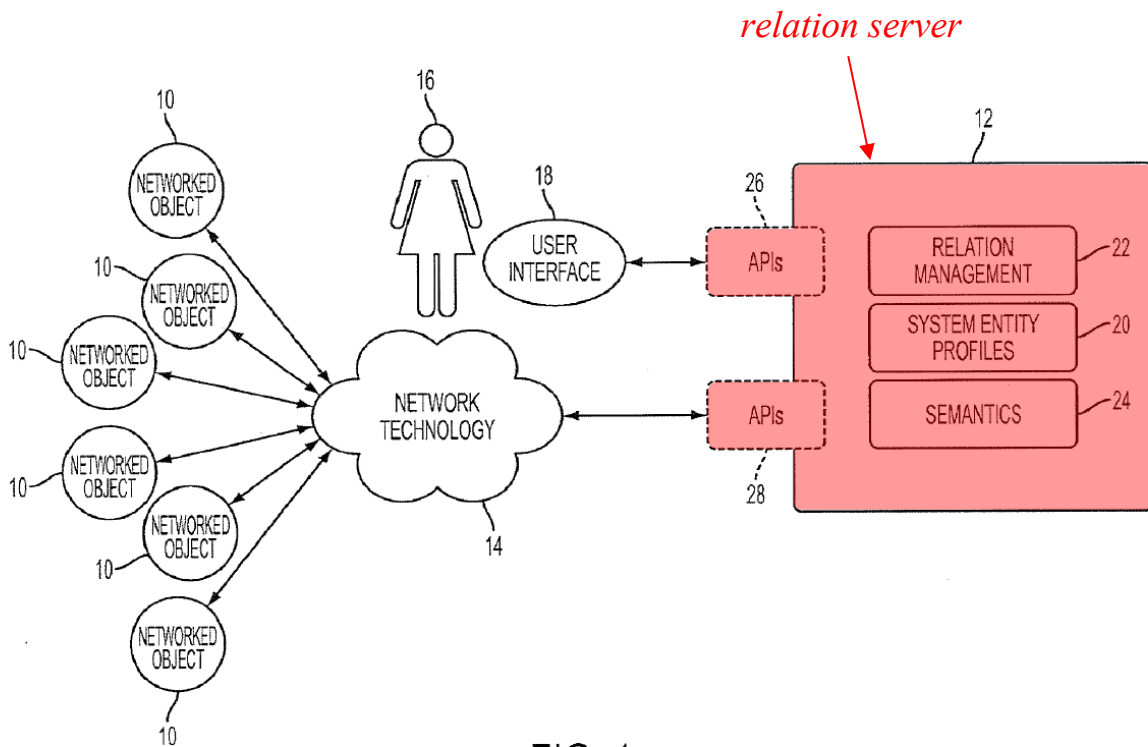
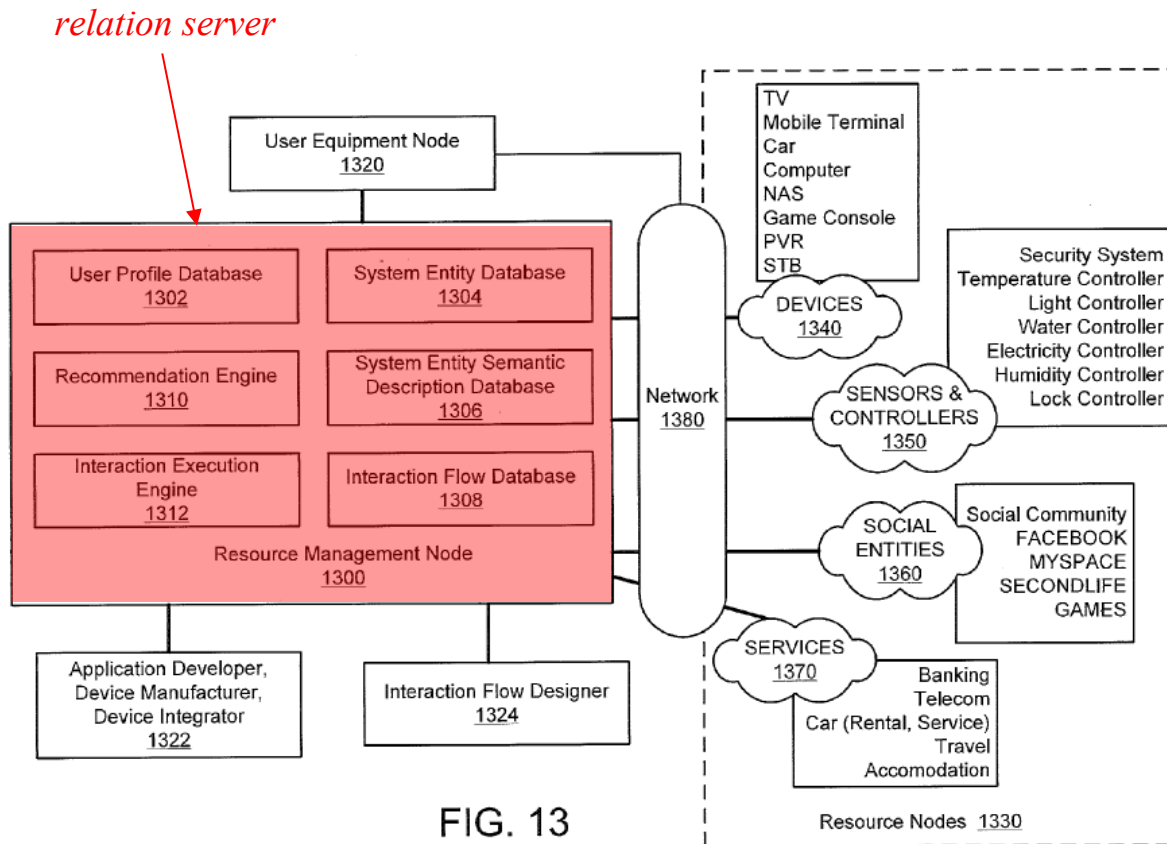


FIG. 1

Ex.1005, Fig. 1 (annotated); Ex.1003, ¶60

In Figure 13, Ermis “depicts another system architecture that includes a resource management node 1300 that manages a plurality of resource nodes 1330.” Ex.1005, [0096]. The resource management node 1300, alternatively or in combination with the system 12, also renders obvious a *relation server*. Ex.1003, ¶¶61-62.



Ex.1005, Fig. 13 (annotated); Ex.1003, ¶62

Ermis explains that its “example embodiments” together “provide a number of advantages and benefits relative to existing resource management software,” such as “streamline[d] interactions between users and the networked resource nodes.” Ex.1005, [0007]. In light of those advantages, Ermis expressly instructs a POSITA to combine the features of its example embodiments. Ex.1005, [0012] (“[I]t is intended that all embodiments disclosed herein can be implemented separately or combined in any way and/or combination.”), [0085] (“[E]ach feature or element can be used...in various combinations with or without other features

and elements disclosed herein.”). Accordingly, a POSITA would have been motivated to and expected success in combining elements from Ermis’s various examples to achieve its stated advantages. Ex.1003, ¶63; *Boston Scientific Scimed, Inc. v. Cordis Corp.*, 554 F.3d 982, 991 (“Combining two embodiments disclosed adjacent to each other in a prior art patent does not require a leap of inventiveness.”)

Thus, Ermis describes *a relation server* (system 12 alone or in combination with APIs and/or resource management node 1300). Ex.1003, ¶¶59-64.

[1.2.2] for storing a relation profile

The ’518 patent defines a *relation profile* as “data representing relations formed to perform a task.” Ex.1001, 6:54-56.

First, Ermis describes forming and storing relationships between networked objects/resource nodes and associated data flows, which renders obvious *a relation profile*. Ermis describes forming “relationship[s] between...at least two resource nodes.” Ex.1005, [0009], [0042] (“establish[ing] relationships between system entities 10”). These “relationships” are “**exercise[d]...in performance of various task requests.**” Ex.1005, [0044]. Thus, Ermis’s “relationships” render obvious a *relation profile* as described in the ’518 patent because they include “data representing relations formed to perform a task.” Ex.1001, 6:54-56; Ex.1003, ¶66.

Ermis provides multiple examples of these relationships. In one example shown in Figures 16-17, a relationship “between [a] light controller and a surveillance camera” is used to turn off the lights in a house when the user is away and show surveillance camera footage to confirm that the lights were turned off. Ex.1005, [0110]. The system “determines that one or more lights have been left on” and “communicat[es] a message to the user,” shown below. Ex.1005, [0109]. If “the user has selected the illustrated ‘yes, turn off’ button,” it “serves as a triggering condition that causes...the light controller to turn off the lights.” Ex.1005, [0110]. After the lights have been turned off, the system “communicates another message to the user...that queries whether the user wants to view a picture/video from the security camera to observe a room which may include the controlled lights.” Ex.1005, [0110]; Ex.1003, ¶67.

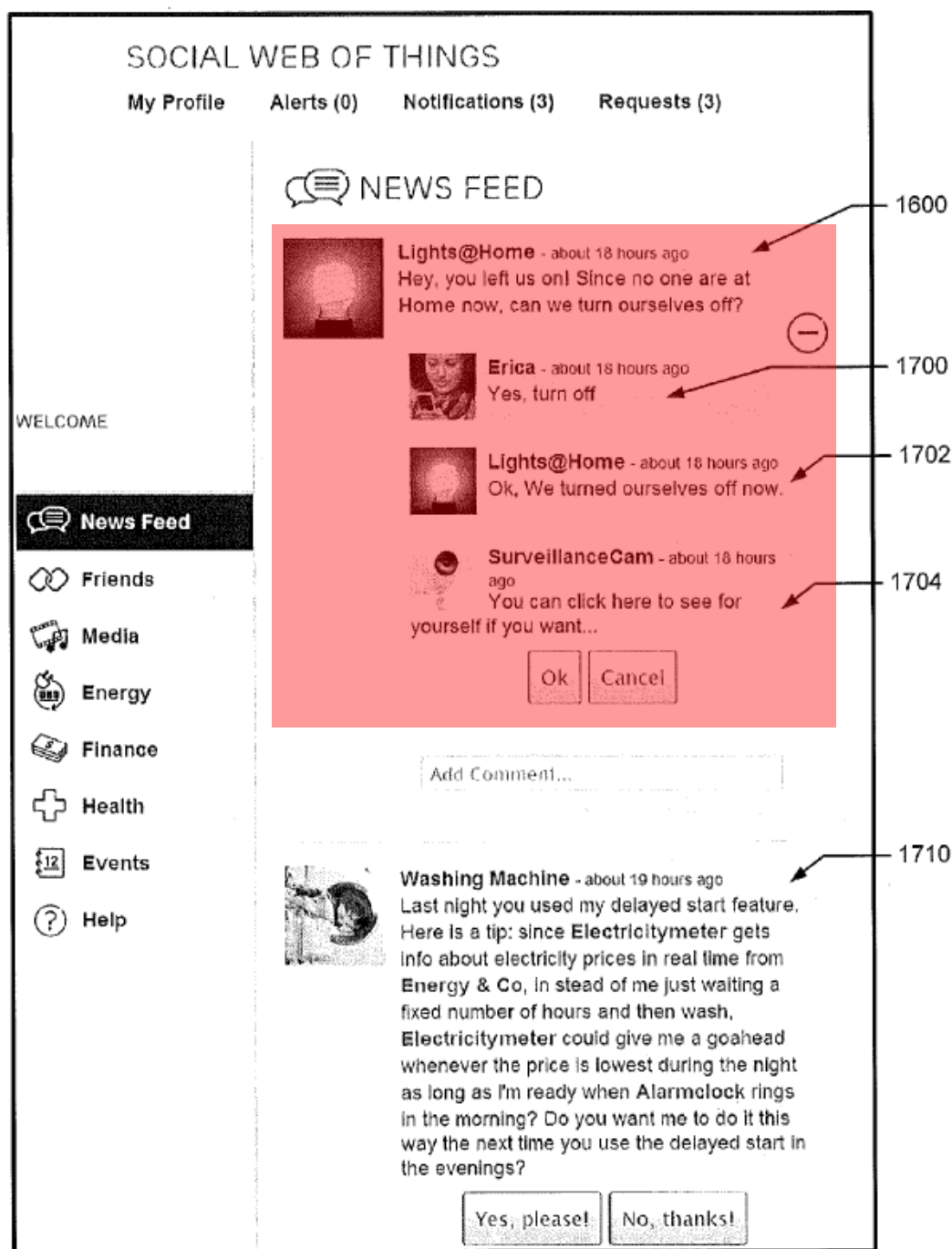


FIG. 17

Ex.1005, Fig. 17 (annotated); Ex.1003, ¶67

This example shows data representing relations formed between the light

controller and surveillance camera to perform the task of turning off lights and displaying surveillance camera footage, which renders obvious *a relation profile*. Ex.1003, ¶68.

In another example in Figures 17-18, a relationship between a “washing machine,” an “electricity controller,” and an “alarm clock” is used to schedule a washing machine while the user sleeps based on “the price of electricity and a planned wake-up alarm time.” Ex.1005, [0111]. “[W]hen the triggering criteria is satisfied (e.g., price of electricity has dropped below the define[d] threshold value and/or less than a defined length of time remains before the wake-up time),” the system “responds by turning on the washing machine.” Ex.1005, [0113]; Ex.1003, ¶69.

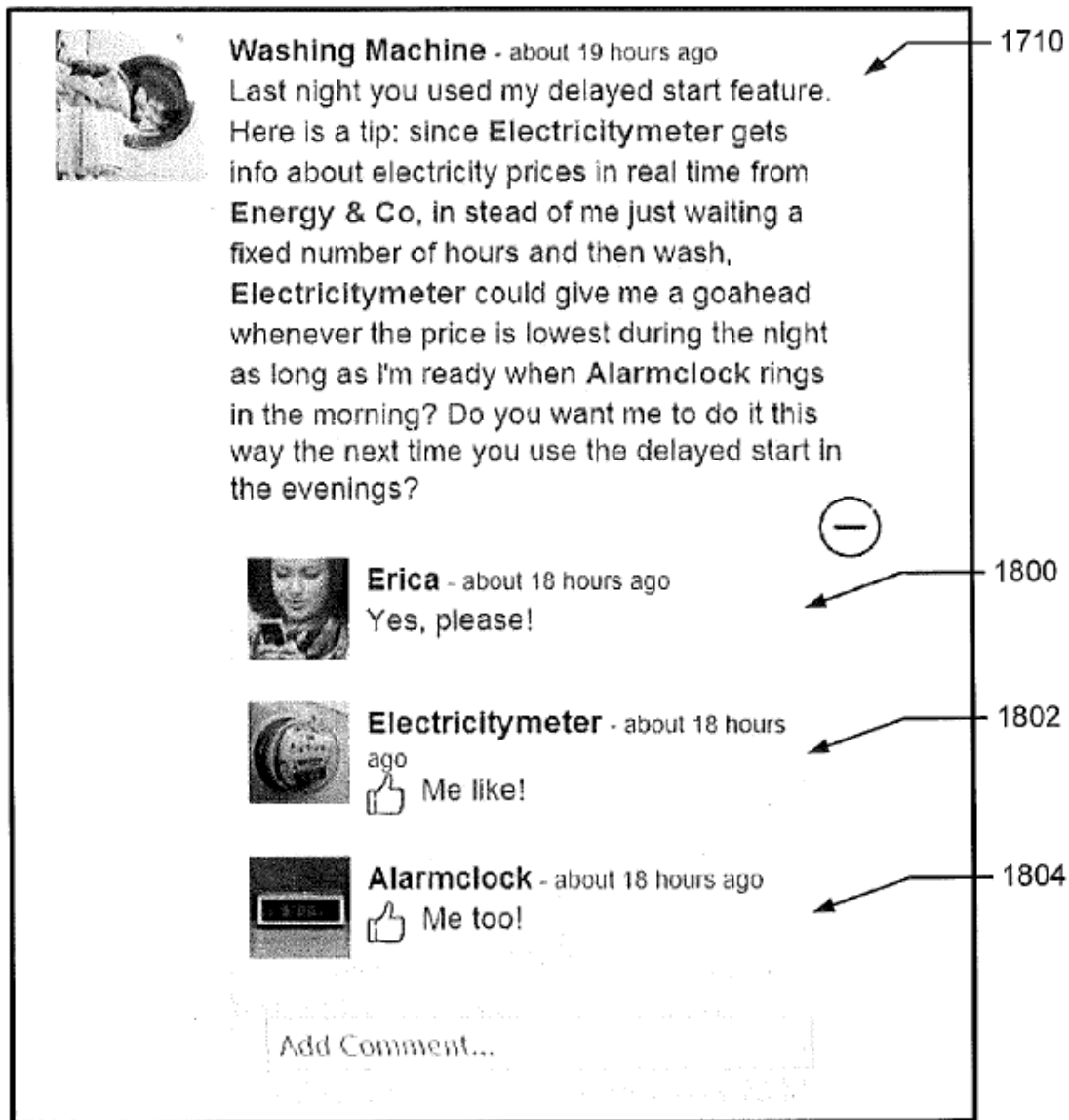


FIG. 18

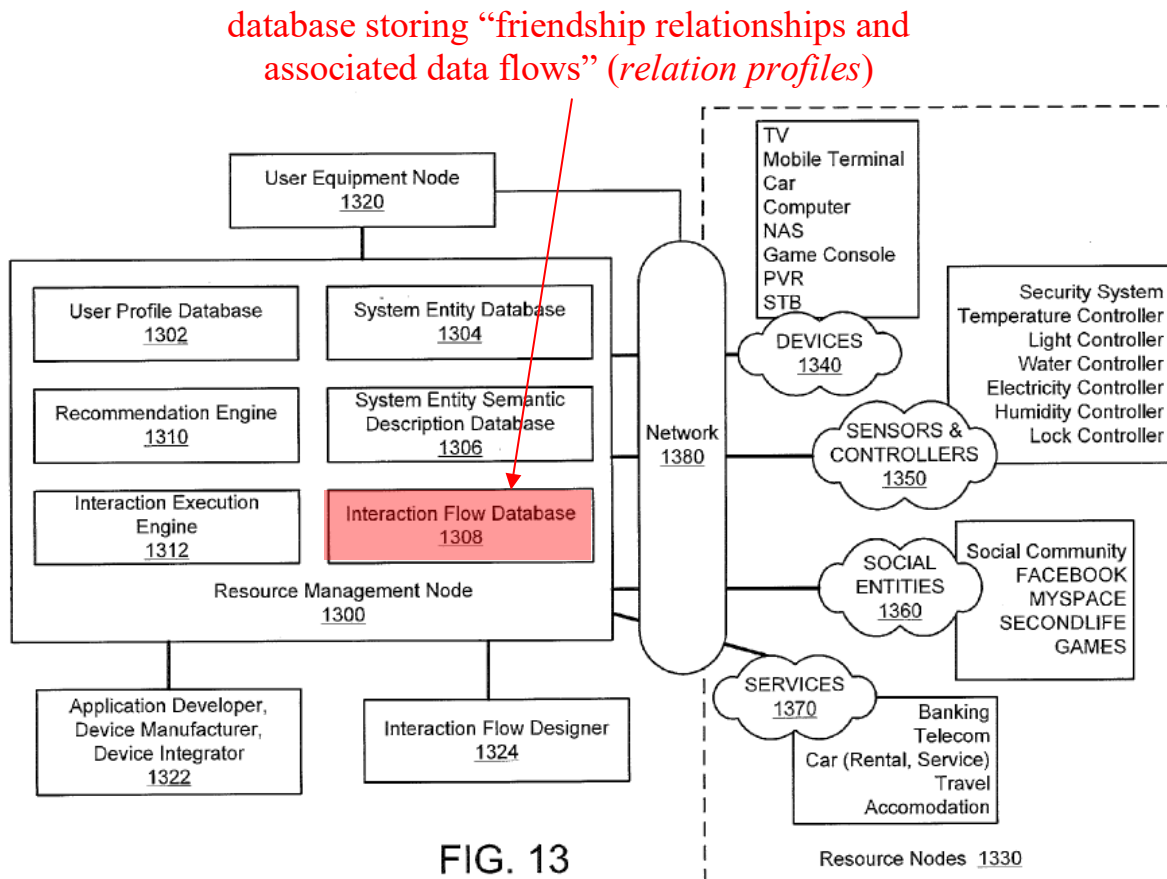
Ex.1005, Fig. 18

This example shows data representing relations formed among the washing machine, the electricity controller, and the alarm clock to perform the task of

starting the washing machine, and, therefore, also renders obvious a *relation profile*. Ex.1003, ¶70.

Ermis provides multiple other examples of relation profiles which will be explained throughout. *See* Ex.1005, [0126], Fig. 24 (new relationship formed between a rental car and a user’s phone to stream “movies and music” and provide navigation directions), [0045]-[0046], [0053]-[0054] (relation formed between a TV and a video recorder to record a TV program), [0047] (relationships between a network attached storage device with other machines to use the storage device “as media source”), [0048] (relations between “a light, 34, a television 36, and a radio 38” for monitoring “power consumption”), [0074] (relationship between a user and a friend’s storage device to “search for interesting media”), [0124] (relationship between a user’s car and smart phone to schedule a service appointment). Various aspects of these example relation profiles will be described as relevant to the claim limitations throughout. Ex.1003, ¶¶71-73.

Second, these relation profiles, along with associated data flows (the multiple processes performed by the devices within the relationship) are stored in the interaction flow database 1308 of the relation server. Ex.1005, [0101], [0105], [0128] (“The interaction execution engine 1312 **adds the selected friendship relationships and associated data flows to the interaction flow database 1308**”); Ex.1003, ¶74.



Ex.1005, Fig. 13 (annotated); Ex.1003, ¶74

In the embodiment of Figure 1, relationship data is stored in a database in the “relation management function 22,” which establishes and manages “**the relationship between networked objects.**” Ex.1005, [0038], [0042], [0052]. That the relationships are established and managed confirms that they are stored in the relation management function 22. *See* Ex.1005, [0042] (describing “temporal and permanent relationships”); Ex.1003, ¶75.

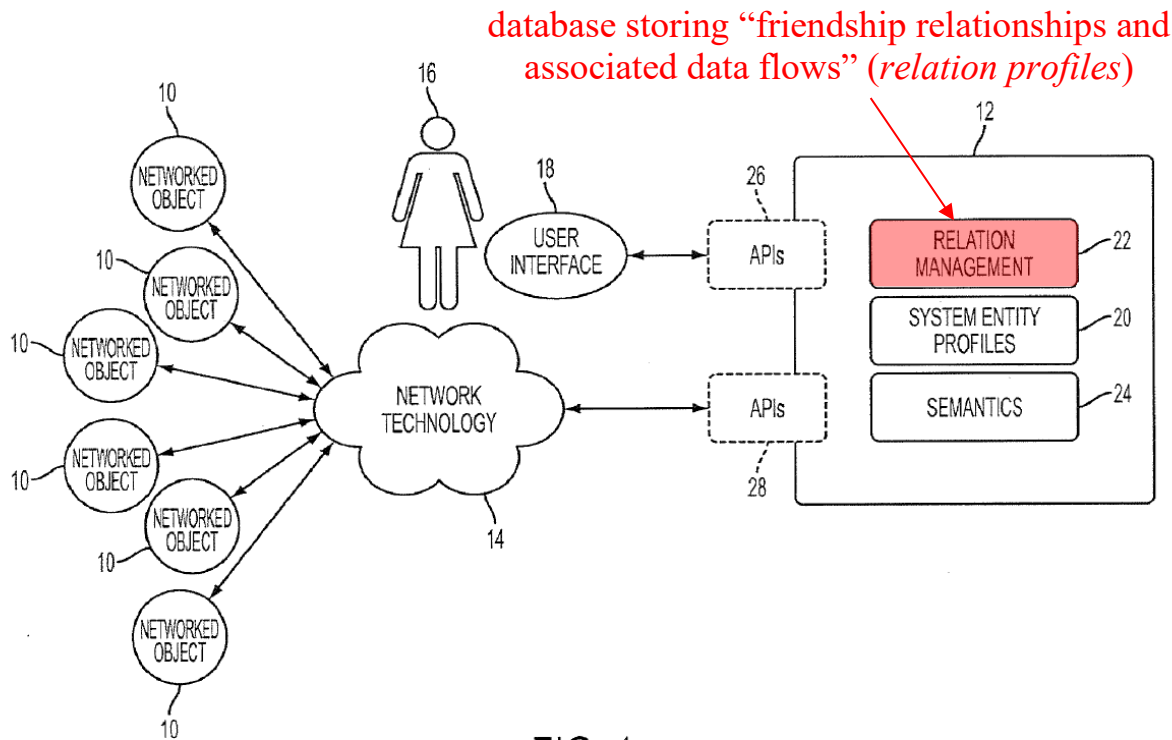


FIG. 1

Ex.1005, Fig. 1 (annotated); Ex.1003, ¶75

Thus, Ermis renders obvious *a relation server* (system 12 with APIs 26 and 28 and/or resource management node 1300) *for storing a relation profile* (“friendship relationships and associated data flows”). Ex.1003, ¶¶65-76.

[1.2.3] including a task processing schedule parameter which defines a sequence of performing multiple processes,

Ermis describes “data flows” which render obvious *task processing schedule parameter[s]* because they *define[] a sequence of performing multiple processes*.

Ex.1005, Title; Ex.1003, ¶77.

As explained at limitation [1.2.2], Ermis describes forming relationships

between the networked objects/resource nodes to perform a variety of tasks. *See* §IX.C.2.[1.2.2]. Ermis describes these data flows as “functional data flows,” Ex.1005, [0118], that “are to be **performed**,” Ex.1005, Abstract, [0008], [0011], [0101]-[0102]. “[D]ata flows...**define** how applications and/or connected resource devices will communicate with each other and **operate to provide defined functionality**.” Ex.1005, [0119]; Ex.1003, ¶78.

In Ermis, the data flows stored in interaction flow database 1308 contain information that “defines data flows that are permitted between identified ones of the resource nodes” and “further defines associated triggering criteria for **when identified ones of the data flows are to be performed**.” Ex.1005, [0101]. “Each data flow can identify two or more resource nodes 1330 that are communicatively connected so that the data output of one resource node is provided as a data input to another resource node, and may connect the respective output and input interfaces of a plurality of resource nodes to provide **a serial daisy chain** or other connection architecture between the identified resource nodes.” Ex.1005, [0101]; Ex.1003, ¶¶79-80.

In the light example described in [1.2.2], “a data flow has been established between the light controller and a surveillance camera” for turning off the lights and displaying surveillance footage. Ex.1005, [0110]. After the lights have been left on more than a threshold time, the user is prompted to select the “yes, turn off”

button, which serves as a triggering event to cause the light controller to turn off the lights. Ex.1005, [0110], Figs. 16-17. The user is then prompted “to view a picture/video from the security camera to observe a room which may include the controlled lights.” Ex.1005, [0110]. “If the user selects the illustrated ‘Ok’ button, a further triggering event causes...the security camera to output the picture/video.” Ex.1005, [0110], Fig. 17; Ex.1003, ¶81.

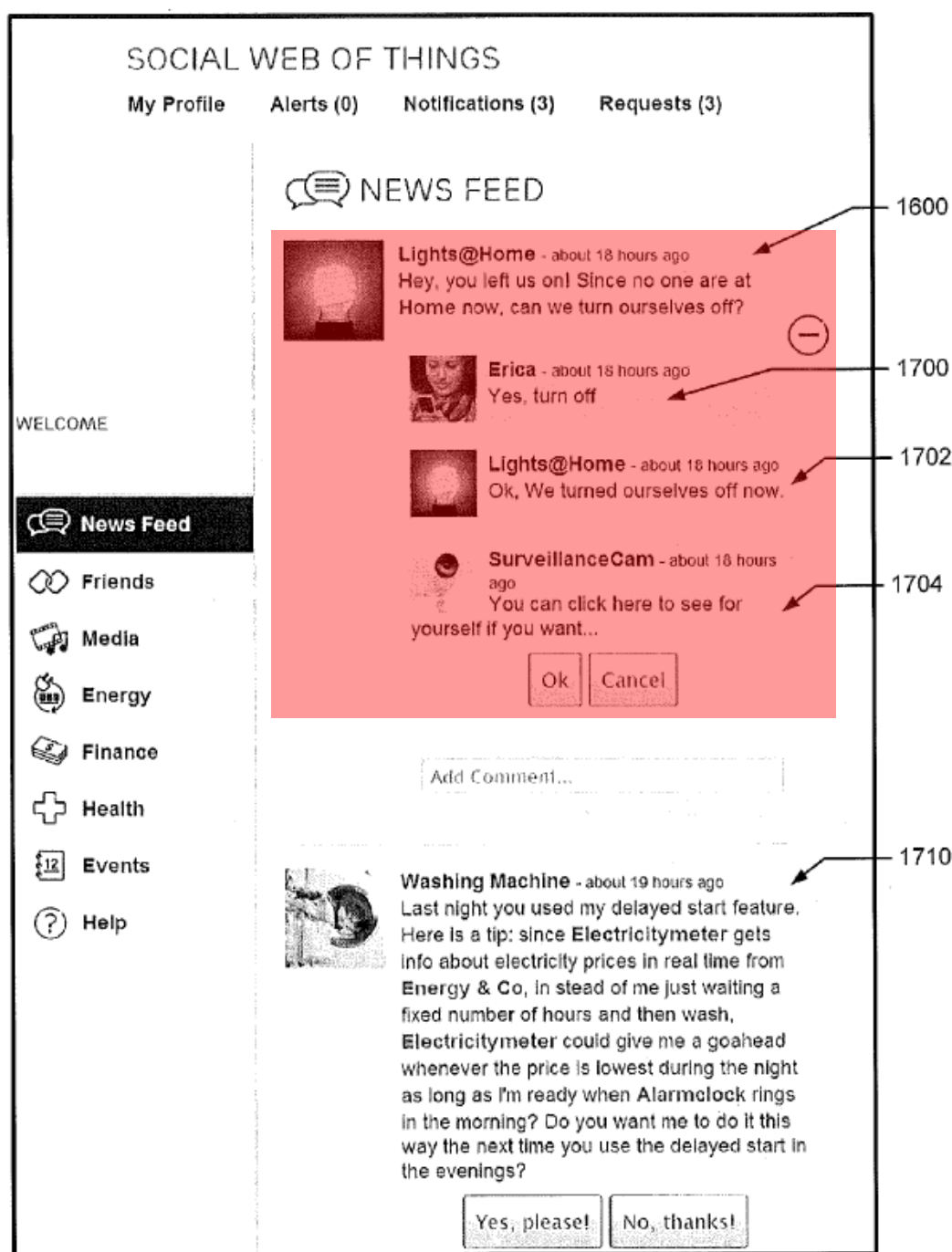


FIG. 17

Ex.1005, Fig. 17 (annotated); Ex.1003, ¶81

The data flow performed by the light controller and a surveillance camera,

including triggering criteria for turning off the lights by the light controller and displaying surveillance footage, renders obvious *a task processing schedule parameter which defines a sequence of performing multiple processes*. Ex.1003, ¶82.

In the “washing machine” example of Figs. 17-18, Ermis describes a “recommended data flow” from “the energy company controller interface to the electricity meter, from the alarm clock to the electricity meter, and from the electricity meter to the washing machine.” Ex.1005, [0112]. The information identifies the “triggering criteria...which may include when the price of electricity falls below a defined threshold value and/or a threshold time duration before a wake-up time defined by the alarm clock.” Ex.1005, [0112]. When “the triggering criteria is satisfied,” the washing machine is turned on. Ex.1005, [0113]. Ex.1003, ¶83.

This data flow including determining an electricity price, determining a threshold time duration before a wake-up time, and turning on the washing machine also renders obvious *a task processing schedule parameter*. Ex.1003, ¶84; *see also* Ex.1003, ¶¶85-87.

As explained at limitation [1.2.2], Ermis’s data flow is included within the *relation profile*. Ex.1005, [0101], [0105], [0128] (“friendship relationships and associated data flows”). Ermis confirms that data flows and triggering criteria are

part and parcel with the friendship relationships between devices because they are created and stored together. *See* Ex.1005, [0128] (“The interaction execution engine 1312 **adds the selected friendship relationships and associated data flows to the interaction flow database 1308.**”); Fig. 24 (illustrating a request to establish new relationships including associated data flows); [0101], [0105]. Ermis also explains that “friendship requests” are sent “**with information identifying the functionality** that will be created for the user by the one or more further recommended data flows.” Ex.1005, [0132]; Ex.1003, ¶¶88-89.

Therefore, Ermis renders obvious *a relation profile* (friendship relationship and associated data flow) *including a task processing schedule parameter* (data flow) *which defines a sequence of performing multiple processes* (data flows “define” how devices “operate to provide defined functionality” to perform a requested task). Ex.1003, ¶¶77-90.

[1.2.4] and for requesting selected machines of the multiple machines to perform the multiple processes,

After a data flow is established, the system 12 (*relation server*) requests the selected machines to perform the multiple processes. “The interaction execution engine 1312 can...**control operations performed by the resource nodes** on the data.” Ex.1005, [0101]-[0102]; Ex.1003, ¶¶91-92.

In the example of turning off lights when the user is away, “**the interaction**

execution engine 1312...signal[s] the light controller to turn off the lights” and later “signal[s] the **security camera to output the picture/video....**” Ex.1005, [0110]; Ex.1003, ¶93.

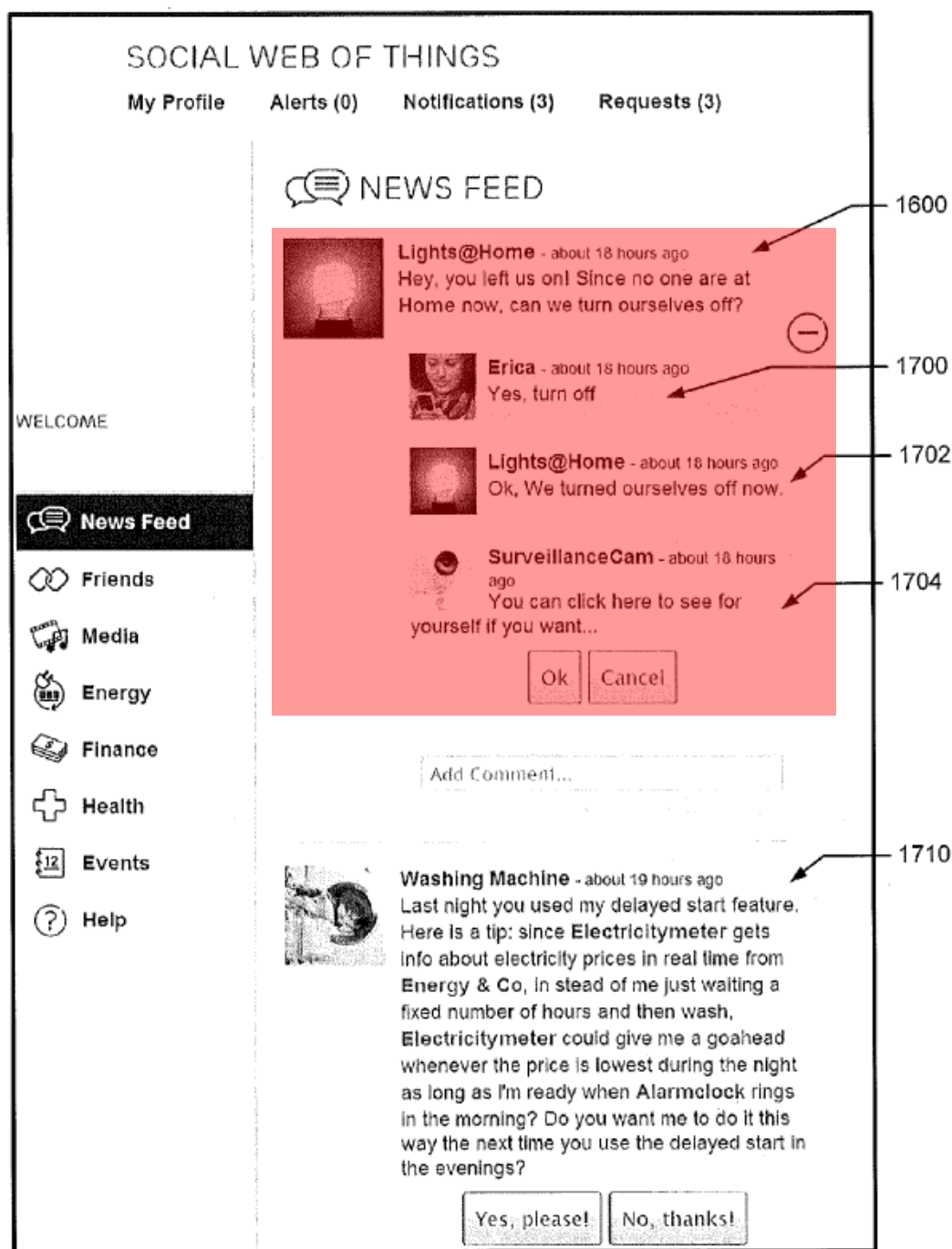


FIG. 17

Ex.1005, Fig. 17 (annotated); Ex.1003, ¶93

In the example of turning on the washing machine when the price of electricity is low and before a wake-up time of an alarm clock, the interaction execution engine 1312 first requests from the electricity meter to give it “a goahead whenever the price is lowest during the night as long as [the laundry is] ready when Alarmclock rings in the morning,” Ex.1005, Fig. 17, and “responds by **turning on the washing machine.**” Ex.1005, [0113], [0102] (“The interaction execution engine 1312 can...control the flow of data between resource nodes and further control operations performed by the resource nodes on the data.”); Ex.1003, ¶94; *see also* Ex.1003, ¶¶95-97.

In each of these examples, the user’s interface 18 (“mobile phone”) is included within the selected machines to perform the task because the relation server receives the task request from the user interface 18 and requests confirmation from the user interface 18 to implement or perform data flows. *See* Ex.1005, [0050], [0109]; Ex.1003, ¶98.

Thus, Ermis describes *requesting selected machines of the multiple machines* (networked objects 10, resource nodes 1330, user interface 18, user equipment node 1320) *of the multiple machines to perform the multiple processes* (system 12/resource management node 1300 instructs each identified machine to perform tasks to execute user request). Ex.1003, ¶¶91-99.

[1.3] wherein the relation server generates a new relation profile based on an intervention by a user in the relation profile,

As explained at limitation [1.2.3], Ermis describes creating new relationships (a *new relation profile*) including new data flows based on user actions. See Ex.1005, Fig. 24; Ex.1003, ¶100.

In Figure 24, a rental car requests that a new relationship be formed between the rental car, “a media server device and a phone terminal” as well as “a map application, and a navigator application.” Ex.1005, [0126]; Fig. 24; Ex.1003, ¶101.

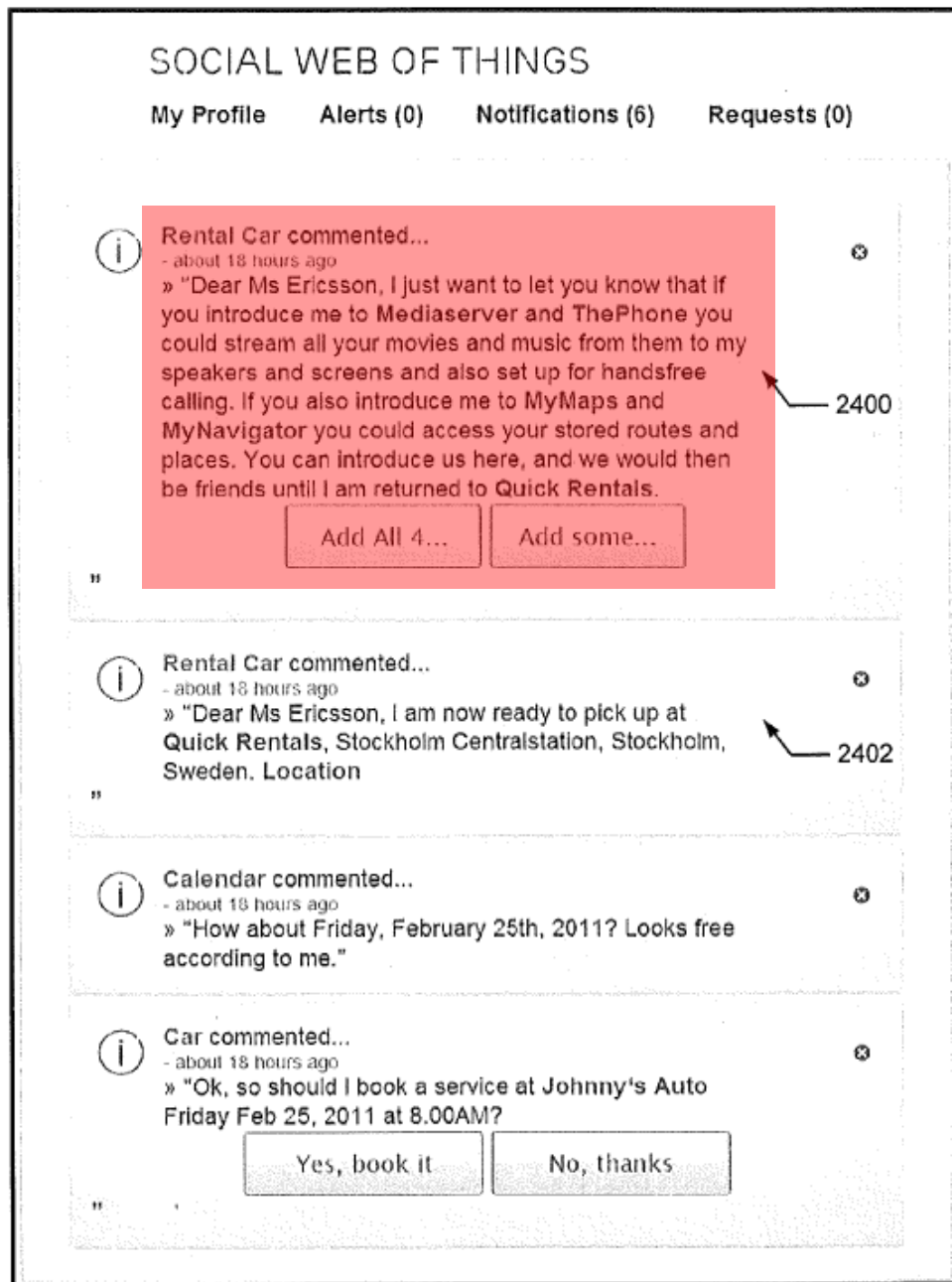


FIG. 24

Ex.1005, Fig. 24 (annotated); Ex.1003, ¶101

These new relationships are generated based on the user's selection of the "Add all 4..." button:

The user can establish a friendship status between the identified resource nodes **by selecting the “Add All 4...” button**, or can select which of the resource nodes are to have friendship satisfy selecting the “Add some...” button. The interaction execution engine 1312 **adds the selected friendship relationships and associated data flows to the interaction flow database 1308**.

Ex.1005, [0128]; Ex.1003, ¶102.

Thus, Ermis renders obvious *wherein the relation server* (interaction execution engine 1312 and/or interaction flow database 1308 within resource management node 1300) *generates a new relation profile* (“adds the selected friendship relationships”) *based on an intervention by a user in the relation profile* (user selecting “Add All 4...” button). Ex.1003, ¶103.

In another example, the user’s “VCR for some reason was unavailable or incapable of executing the requested task (e.g., recording a TV program).” Ex.1005, [0053]. The TV requests that a new relationship be formed with “other networked nodes.” Ex.1005, [0053]. “[T]he TV...ask[s] its owner if it is permitted to contact the owner’s friend to ask for the needed favor.” Ex.1005, [0054]. If approved, this new relationship is also *a new relation profile* created based on *an intervention by a user*. The new relationship information is added, along with the corresponding data flow, to the interaction flow database 1308. Ex.1005, [0128]; Ex.1003, ¶¶104-105.

Thus, Ermis renders obvious *wherein the relation server* (the relation management function 22 of the system 12) *generates a new relation profile* (establishing a new friendship relationship between TV and device of a user's friend) *based on an intervention by a user in the relation profile* (user 16 approves recommended friendship relationship and associated data flow). Ex.1003, ¶¶100-106.

[1.4] wherein the task processing schedule parameter configures a start time of at least one process of the multiple processes, and

Ermis provides multiple examples in which a data flow and associated triggering criteria (*task processing schedule parameter*) configures a start time. Ex.1003, ¶107.

In the example data flow for turning off lights, the *process* of turning off lights starts when “one or more lights have been left on **more than a threshold time**.” Ex.1005, [0109]. After this threshold time, a message is communicated “requesting the user's instructions for turning the lights off or leaving the lights on.” Ex.1005, [0109]. Starting the process of turning off the lights after a threshold time renders obvious *a start time of at least one process of the multiple processes*. Ex.1003, ¶108.

In the example data flow for starting a washing machine, the data flow configures a start time of the washing machine. “The recommendation engine 1310

has further determined that...the electricity controller can control **the start time** of the washing machine, and determined that the electricity controller can use the price of electricity and a planned wake-up alarm time defined in the alarm clock to make decisions when the washing machine should be started.” Ex.1005, [0111]-[0112]; Ex.1003, ¶¶109-111.

In the example of recording a TV program, the start time of the TV program is similarly determined and may be set as a “**future planned (timer set) event[]**.” Ex.1005, [0091]; Ex.1003, ¶112.

Thus, Ermis renders obvious *wherein the task processing schedule parameter (data flow) configures a start time of at least one process of the multiple processes (determining a start time of a washing machine or a start time for recording a TV program)*. Ex.1003, ¶¶107-113.

[1.5] wherein the multiple machines includes at least one of a home appliance, a smart phone or a search engine.

Ermis’s machines include a TV, a video recorder, a “washing machine,” an “alarm clock,” and lights (*home appliance[s]*). Ex.1005, [0046], [0109]-[0111]; Ex.1003, ¶114.

Ermis also describes the “user’s application interface 18” as a “mobile phone,” Ex.1005, [0047], which runs various applications, [0049]. Accordingly, this mobile phone is *a smart phone*. Ex.1003, 115.

Thus, Ermis renders obvious *a home appliance* (TV, video recorder, washing machine, alarm clock, lights), or *a smart phone* (mobile phone). Ex.1003, ¶¶114-116.

3. Claim 2

[2.0] *The system of claim 1, wherein: a first machine of the multiple machines forwards a command received from the user to the relation server,*

In Ermis, a user command is sent to the system 12 via the user interface 18, (the user's mobile phone). "[T]he user 16 can for example send **a task request**, for example via **user interface 18** and API 26 **to the TV's system entity (operative within system 12)** requesting the system 12 to record a certain TV program." Ex.1005, [0045]; Ex.1003, ¶¶117-118.

The path of this task request (*a command*) is illustrated in Figure 1 from the user 16's mobile phone (*first machine*) to the system 12 (*relation server*), below. Ex.1003, ¶119.

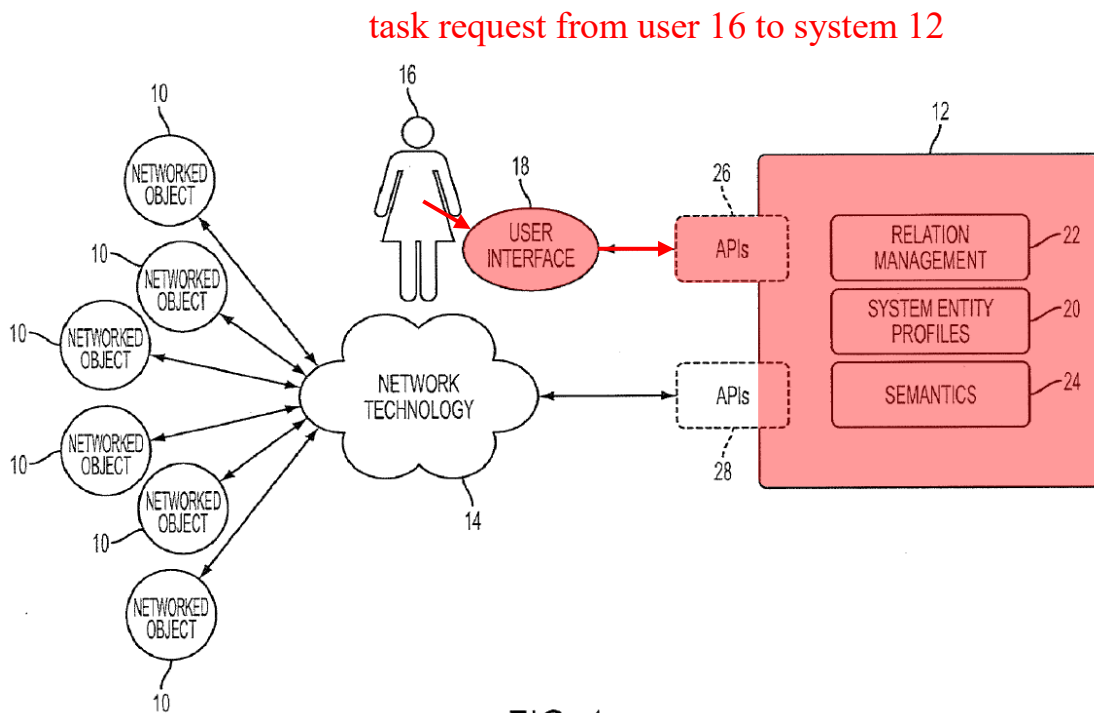


FIG. 1

Ex.1005, Fig. 1 (annotated); Ex.1003, ¶119

The user interface 18 is the user's mobile phone or other electronic device. Ex.1005, [0039] (“user interfaces 18” may be implemented as “**mobile devices, computers, laptops, televisions or other devices**”). Therefore, the user interface 18 renders obvious *a first machine*. Ex.1003, ¶¶120-121.

Therefore, Ermis renders obvious *a first machine of the multiple machines* (user interface 18) *forwards a command* (“send[s] a task request”) *received from the user* (from user 16) *to the relation server* (to the system 12). Ex.1003, ¶¶117-122.

[2.1] the relation server controls the selected machines for performing the multiple processes according to the command, and

As explained at limitation [1.2.4], Ermis's system 12 controls the devices necessary for performing a user-requested task to perform the task. Ex.1005, [0101]-[0102]. In the TV program recording example, the system 12 (via system entities for the TV and recorder) issues commands to the TV and video recorder to record the TV program. Ex.1005, [0045]-[0046]; Ex.1003, ¶123.

Ermis's description of a server controlling machines in response to a user command is consistent with the '518 patent's description. Ex.1001, Fig. 2, 9:4-6 (commands S10 and S12 including "a 'movie recommendation' command," steps S30 and S32 selecting a movie, steps S42 and S44 confirming payment, etc.); Ex.1003, ¶124.

Ermis explains how its server (system 12/resource management node 1300) controls machines as follows:

The interaction execution engine 1312 may, for example, provide a daisy chain data flow through three resource nodes, by **controlling** the first resource node to output data having certain characteristics (e.g. data format, data content, etc.) to an input interface of the second resource node, **control** the second resource node to operate on the data and to output data having certain characteristics (e.g. data format, data content, etc.) to an input interface of the third resource node. The interaction execution engine 1312 can similarly **initiate and control** the flow of data between resources nodes and further **control** operations performed by the resource nodes on the data.

Ex.1005, [0102]; Ex.1003, ¶125.

Figure 13 shows that the interaction execution engine 1312 is part of the resource management node 1300. Ex.1003, ¶126.

Thus, Ermis describes *the relation server* (system 12/resource management node 1300) *controls the selected machines* (“control[s] operations performed by the resource nodes”) *for performing the multiple processes according to the command* (to execute processes to perform a user’s task request). Ex.1003, ¶¶123-127.

[2.2] *the selected machine² includes the first machine.*

As explained at limitation [1.2.2], Ermis’s system 12 identifies all the machines necessary to perform a task. As explained at limitation [1.2.4], the selected machines include the user’s interface 18 or mobile phone. In the examples of turning off lights and starting a washing machine, the data flows are implemented and performed after approval by the user. Ex.1005, Figs. 16-18. In the example of recording a TV program, the system 12 receives the command from the user interface 18 and sends this command to the relevant machines (the TV and

² There is no antecedent basis for the term “*selected machine*” (singular). Prior claim language referred to “*selected machines*” (plural). Ermis renders this limitation obvious regardless.

video recorder, Ex.1005, [0045]-[0046]). “[T]he user 16...receive[s] a confirmation from the TV’s system entity (again via API 26 and **user interface 18**) that the requested task will be performed, and later on that it has been successfully completed.” Ex.1005, [0046]; Ex.1003, ¶¶128-129.

Thus, Ermis renders obvious *the selected machine[s]* (networked objects 10 and user interface 18 necessary to complete a user request) *include[] the first machine* (user interface 18). Ex.1003, ¶¶128-130.

4. Claim 3

[3.0] *The system of claim 2, wherein: the command includes a reservation or a payment.*

Ermis describes another example in which the user requests that a car (one of the networked objects 10) schedule a reservation for a maintenance appointment with a mechanic (*a reservation*). Ex.1005, Fig. 23; Ex.1003, ¶131.

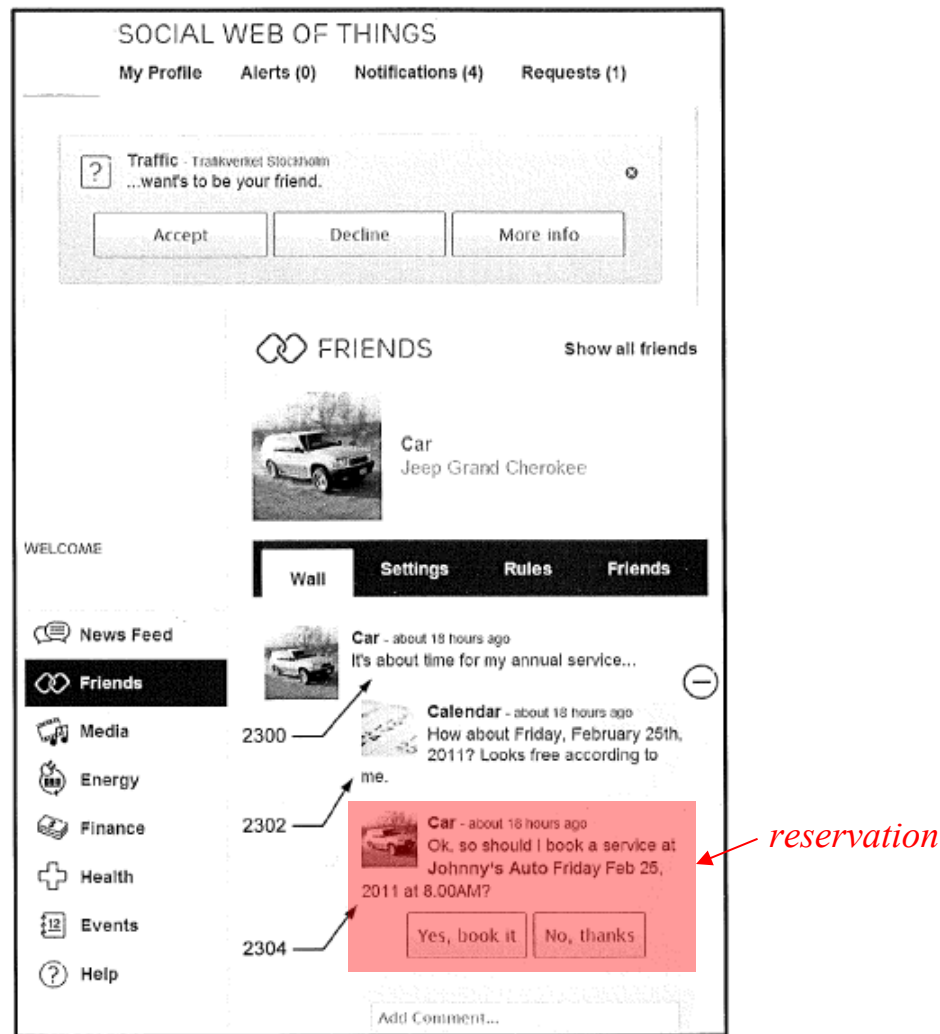


FIG. 23

Ex.1005, Fig. 23 (annotated); Ex.1003, ¶131

“The interaction execution engine 1312 may...display an action message 2304 indicating that the car can **book the service** at the identified auto repair establishment at the indicated time of day and requesting the user’s authorization to proceed with the **booking**.” Ex.1005, [0125]; Ex.1003, ¶132.

When a data flow is performed, “the user 16 will receive a confirmation”

both “that the requested task will be performed, and later on that it has been successfully completed.” Ex.1005, [0046]. Therefore, if the user selects the “Yes, book it” button, the system sends a confirmation that the maintenance service will be booked and later that it has been successfully booked. Ex.1003, ¶133 (citing Ex.1005, [0046]).

Thus, Ermis renders obvious *the command includes a reservation* (user requests that a car of the networked objects 10 schedule a maintenance appointment). Ex.1003, ¶¶131-134.

5. Claim 4

[4.0] *The system of claim 2, wherein: the relation profile includes a capability set parameter, and the capability set parameter corresponds to information about capabilities required for performing the command.*

Consistent with the claim language, the ’518 patent specification describes *a capability set parameter* as “includ[ing] information about capability parameters, which are required for relevant task [sic].” Ex.1001, 7:33-36.

Ermis’s system determines the required capabilities for a given data flow, which renders obvious *a capability set parameter* including *information about capabilities required for performing the command*. “[T]he network management system assigns the requested task to one of the plurality of networked objects which **has a capability to perform the requested task**.” Ex.1005, [0085].

Therefore, the system first determines the capabilities required to perform a

requested task and then assigns the task according to those determined capabilities.

Ex.1003, ¶136.

Ermis's system also includes “[a] **service required capabilities database** which describes **what a service would require** to be meaningfully consumed.” Ex.1005, [0066]. The system first identifies which capabilities of machines are required to perform any command/service. This information may be stored as a parameter in a “service required capabilities database.” Ex.1005, [0066], [0071]. This service required capabilities database is stored in the system 12. *See* Ex.1005, [0048] (system 12 is “implemented as a home gateway (GW) 30” in the embodiment in Figure 2), [0064]-[0066] (“service required capabilities database” is stored in “context management system 110, 112” within “the gateway 30”). These required capabilities for a data flow would be stored with information identifying the relationships performing the data flow. *See* Ex.1005, [0012], [0085]; Ex.1003, ¶137.

Ermis also explains that after the capabilities for a service are identified, the system matches these capabilities with devices in the network that provide these capabilities. A “Service Matchmaker 118 compares the **requirement of the services** in the Service Registry 114 with device capability.” Ex.1005, [0072]; Ex.1003, ¶138.

Again, Ermis provides multiple examples. In Ermis's example of starting the

washing machine, the system first “**identifies capabilities** of the washing machine, the electricity controller, the energy company controller interface, and the alarm clock.” Ex.1005, [0111]. For example, “[t]he recommendation engine 1310 has further determined[:]

[1.] that the price of electricity **can** be obtained in real-time from the energy company controller interface,

[2.] that the electricity controller **can** control the start time of the washing machine,

[3.] ...that the electricity controller **can** use the price of electricity and a planned wake-up alarm time defined in the alarm clock to make decisions for when the washing machine should be started.”

Ex.1005, [0111]. Each of these functions that “can” be performed by these machines are capabilities of the devices needed to perform the requested task (*capability set parameter*). Ex.1003, ¶¶139-140 (citing Ex.1031, Abstract, Fig. 2A). Further, this data identifying the capabilities needed to a perform a task would be generated based on metadata within each profile of the respective machine. *See infra*, claim 11; *e.g.*, Ex.1005, Abstract, [0008], [0011] (“metadata identifying capabilities of the resource nodes”), [0041] (“The system entity profiles 20 may include, but are not limited to, information about...capability.”); Ex.1003, ¶140.

A POSITA would also have understood that this capability data corresponds

to a *capability set parameter* based on the use of the term “parameter” in the ’518 patent and in the art at the time. Data is another term for parameter. *See* Ex.1003, ¶141 (citing Ex.1008, Abstract; Ex.1009, Abstract; Ex.1010, [0007]; Ex.1011, Abstract; Ex.1025, [0072]-[0073]).

In Ermis, the data describing capabilities required to perform a data flow (*capability set parameter*) are included within the relationships between devices and associated data flows (*relation profile*). As explained at limitation [1.2.3], relationships between networked objects in Ermis include the data flows they enable. *See* Ex.1005, [0044], [0128] (“**add[ing] the selected friendship relationships and associated data flows to the interaction flow database 1308**”), Fig. 24. Relationships between devices are created based on the functionality of the devices to perform tasks. *See* Ex.1005, [0087] (System entities “establish a level of friendship relation...**based on the...functionality**” of the devices.). Therefore, relationships between devices in Ermis include the capabilities of the devices required to perform a particular command and are stored in the interaction flow database 1308 and/or system entity database 1304. Ex.1005, [0044], [0128], [0098]; Ex.1003, ¶¶142-143.

Therefore, Ermis renders obvious *the relation profile* (friendship relationship and associated data flow) *includes a capability set parameter* (data identifying capabilities of networked objects), *and the capability set parameter corresponds to*

information about capabilities required for performing the command (capability data identifying capabilities of networked objects of a particular friendship relationship and associated data flow to perform a task). Ex.1003, ¶¶135-144.

6. Claim 5

[5.0] *The system of claim 2, wherein: the relation profile includes a group parameter, and the group parameter corresponds to identities of machines required for performing the command.*

Ermis explains that multiple devices may be grouped together to perform a task and the information identifying the group is included as part the corresponding friendship relationship and associated data flow (*relation profile*). A relationship is formed “between **at least two** of the resource nodes.” Ex.1005, [0009]. The devices are grouped by sending a “message” such as “a friendship request” to the relevant devices. Ex.1005, [0009]. “[T]he user’s acceptance of the friendship request...add[s] **information** to the interaction flow database **that identifies a friend relationship between the at least two resource nodes.**” Ex.1005, [0009]; Ex.1003, ¶145.

The information identifying the at least two resource nodes that form the relationship renders obvious *a group parameter*. As explained by Ermis, these machines were selected to perform “a recommended data flow,” which shows that they are *identities of machines required for performing the command*. Ex.1005, [0009]; Ex.1003, ¶146.

Ermis provides an example in Figure 14 in which the lights are grouped and labeled with the string “Lights@Home.” Ex.1005, Fig. 14; Ex.1003, ¶147.

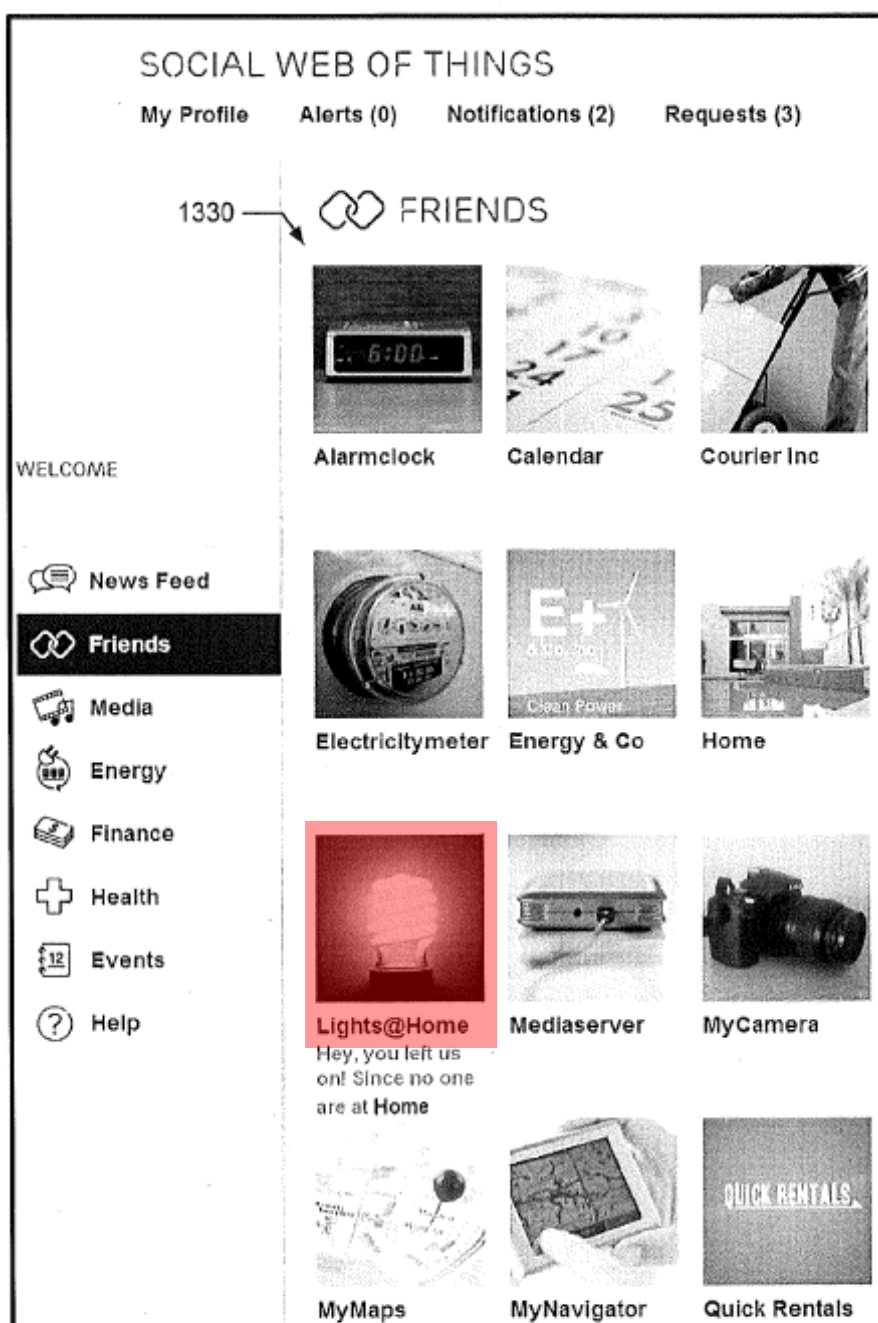


FIG. 14

Ex.1005, Fig. 14 (annotated); Ex.1003, ¶147

When the user leaves the house, they receive a message, as from the

“Lights@Home” group saying “Hey, you left us on! Since no one are [sic] at Home now, can we turn ourselves off?” Ex.1005, Fig. 16; Ex.1003, ¶148.

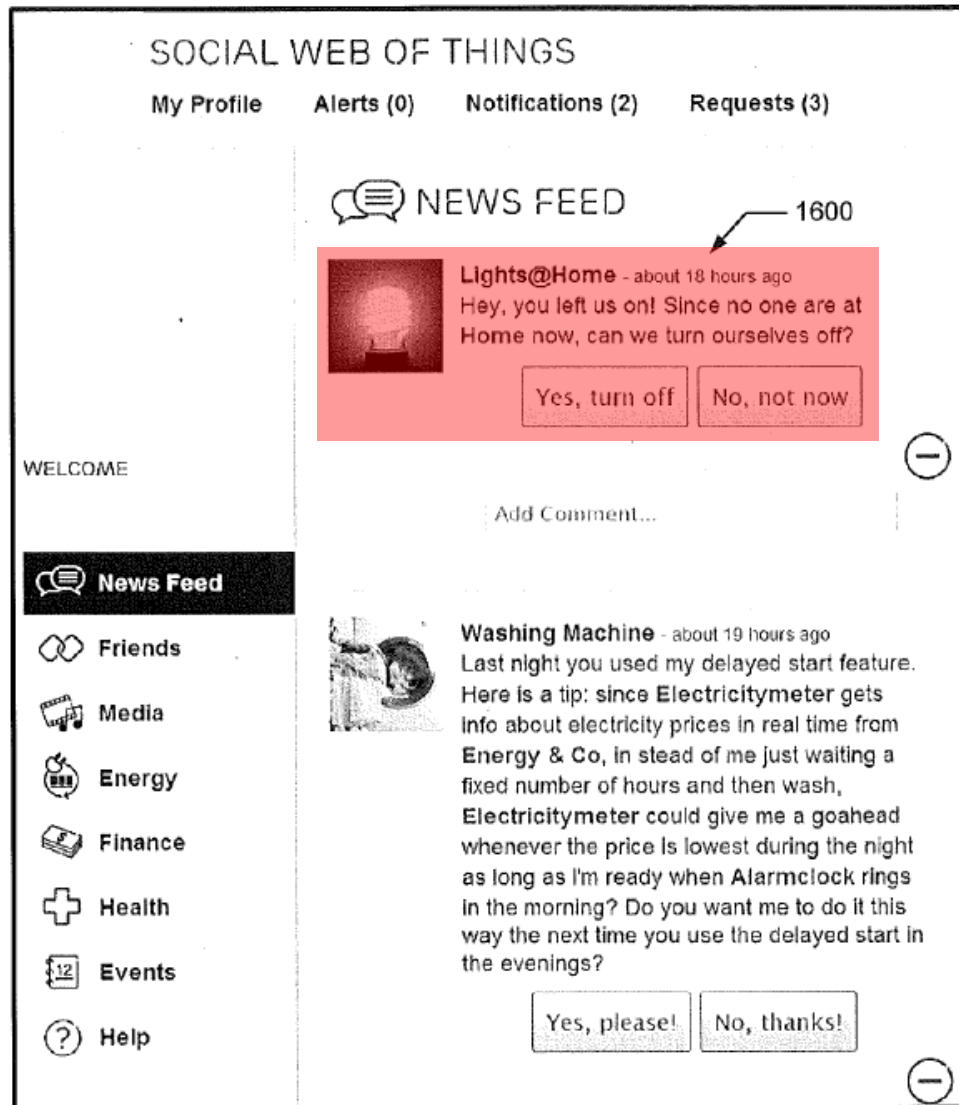


FIG. 16

Ex.1005, Fig. 16 (annotated); Ex.1003, ¶148

Because the “Lights@Home” group includes light devices that will be

turned off by the data flow, they are *machines required for performing the command*. Information identifying this friend relationship between these lights therefore renders obvious *a group parameter which corresponds to identities of machines required for performing the command*. Ex.1003, ¶149.

This group information is included in a relationship and associated data flow (*relation profile*). See Ex.1005, [0009] (relationship data includes “information...that identifies [the] friend relationship between the at least two resource nodes.”). Similarly, the method in Figure 25, which explains how data flows are created, Ex.1005, [0034], [0130], includes first “**identifying a friend relationship** between the at least two resource nodes” at step 2404. Ex.1005, Fig. 25; Ex.1003, ¶¶150-151.

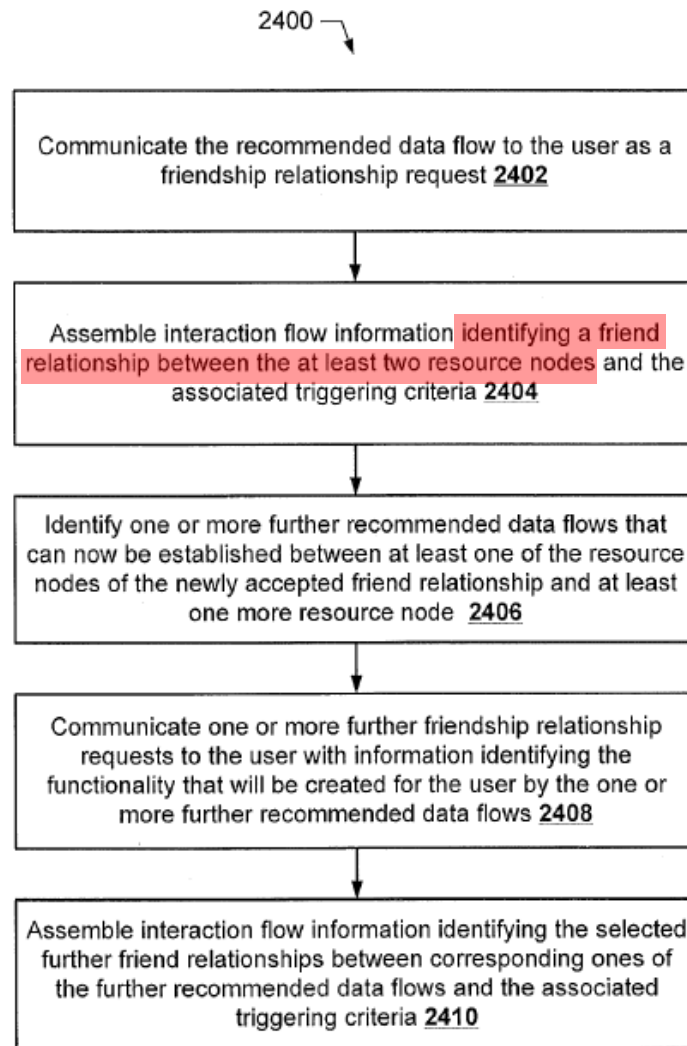


FIG. 25

Ex.1005, Fig. 25 (annotated); Ex.1003, ¶150

Thus, Ermis renders obvious *the relation profile* (friendship relationship and associated data flow) *includes a group parameter* (information identifying a relationship between at least two machines), *and the group parameter corresponds to identities of machines required for performing the command* (machines of the relationship include those needed to perform a user-requested task). Ex.1003,

¶¶145-152.

7. Claim 6

[6.0] *The system of claim 2, wherein: the relation profile includes a group identity parameter, and the group identity parameter corresponds to an identity of a group consisting of machines required for performing the command.*

As explained for claim 5, Ermis describes groups of devices required for a command. The group of lights is labeled “Lights@Home,” which is a label identifying the group (*group identity parameter*). Ex.1005, Fig. 16, [0041] (describing an **identity** of each machine), [0060] (“the **identity** of a device”); Ex.1003, ¶¶153-155.

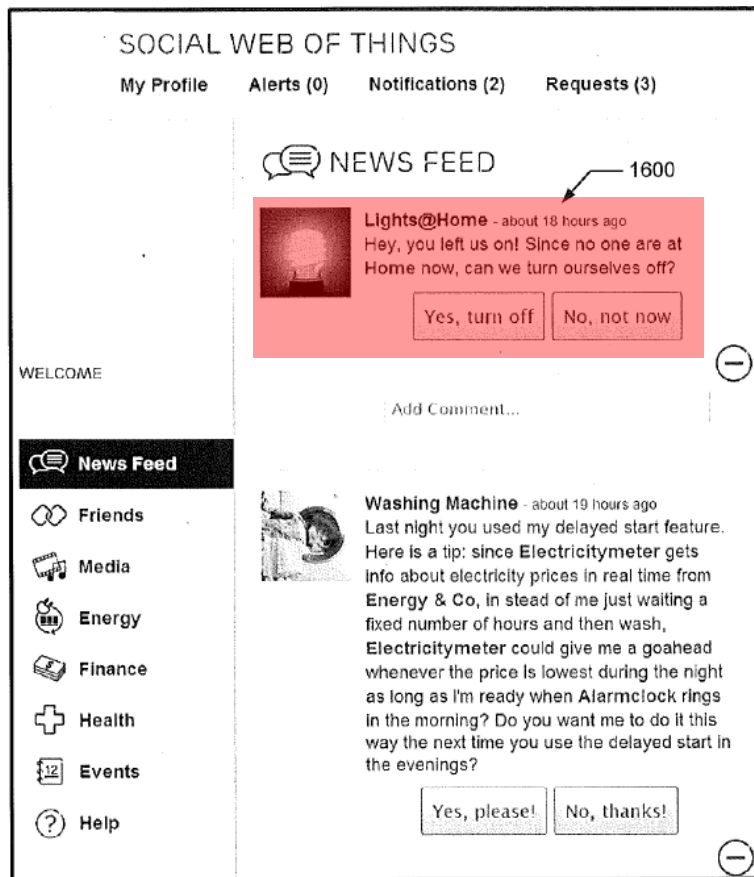


FIG. 16

Ex.1005, Fig. 16 (annotated); Ex.1003, ¶155

Just as the information identifying the machines within a group (*group parameter*) is included within the friendship relationship and associated data flow (*relation profile*), the identify of that group (*group identity parameter*) would also be included within the relationship data. Ex.1003, ¶156.

Thus, Ermis renders obvious *the relation profile* (friendship relationship and associated data flow) *includes a group identity parameter* (title identifying a

relationship between at least two machines, such as “Lights@Home” title), *and the group identity parameter corresponds to an identity of a group consisting of machines required for performing the command* (title identifying machines of the friend relationship needed to perform a user-requested task). Ex.1003, ¶¶153-157.

8. Claim 7

[7.0] *The system of claim 2, wherein: the relation profile includes a task description parameter, the task description parameter corresponds to information related to the command,*

As explained at claim 1, Ermis describes multiple examples of task requests and series of processes (data flows) to fulfil those requests. *See* §IX.C.2.[1.2.3]. These data flows are stored in the “interaction flow database 1308.” Ex.1005, [0101]. The information describing these data flows renders obvious *a task description parameter* because it *corresponds to information related to the command*. Ex.1003, ¶¶158-159.

Information related to data flows includes identities of the machines performing processes of the data flow, which are identified by “a MAC address” or “explicit names.” Ex.1005, [0060]. The names of these machines are displayed in “**a message** to a user that **identifies the recommended data flow**,” as shown in Figure 16. Ex.1005, [0009]; Figs. 17-18, 23-24; Ex.1003, ¶¶160-161.

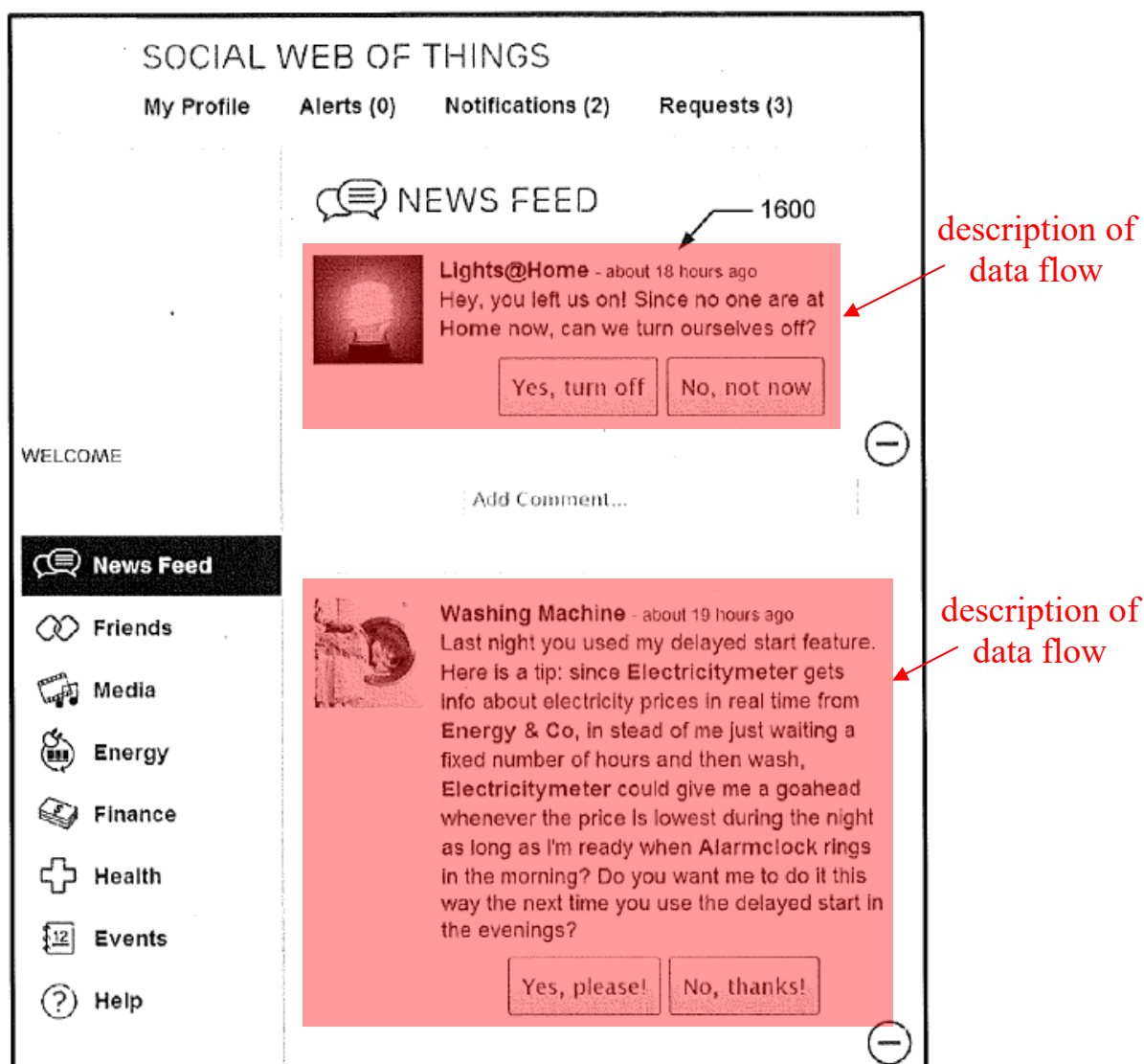


FIG. 16

Ex.1005, Fig. 16 (annotated); Ex.1003, ¶161

“[A] recommended data flow” is identified with “a natural language message 1710” that provides “an explanation of the functional capabilities that will be provided.” Ex.1005, [0111], [0126]; Ex.1003, ¶¶162-163.

Thus, Ermis renders obvious *the relation profile* (friendship relationship and associated data flow) *includes a task description parameter, the task description parameter corresponds to information related to the command* (information defining a data flow illustrated by messages to the user describing the data flow). Ex.1003, ¶¶158-164.

[7.1] and the information is saved in a form of text.

As explained at limitation [7.0], Ermis's data flows, including "information that defines data flows" (a *task description parameter*), are stored in an "interaction flow **database** 1308." Ex.1005, [0101], [0065]-[0069]. To access the information in Ermis's databases, Ermis uses "quer[ies]." Ex.1005, [0070]; Ex.1003, ¶165.

These queries may "take various forms, for instance a SPARQL or XQUERY query." Ex.1005, [0070]. An XQUERY is used to query **text-based databases**. "XQuery" is "a query language **for XML data sources**." Ex.1015, [0030]; Ex.1013, [0045] ("XQuery (a query language that is designed to process data **from XML files**...)"). Because Ermis's databases are queried with XQUERYs, the databases described by Ermis are XML databases. Ex.1003, ¶166. XML databases are text-based. See Ex.1014, 5:1-3 ("**Text-based data**, such as **XML-structured data**..."); Ex.1015, [0029] ("**text-based eXtensible Markup Language (XML)**"); Ex.1016, [0032] ("**text-based XML** documents"). Therefore,

information defining data flows (*a task description parameter*) stored in Ermis's XML databases is also text-based. Ex.1003, ¶¶166-167.

In addition, *information related to [a] command* includes the identities of the machines, such as “a MAC address,” or “explicit names,” which are text-based. Ex.1005, [0060]. It would have been obvious that text-based data would be stored in the same format (as text), not converted into another non-text format for storage. Ex.1003, ¶168.

Therefore, Ermis renders obvious task description *information is saved in a form of text* (information in a text-based XML database, “MAC address,” or “explicit name[]” of machines). Ex.1003, ¶¶165-169.

9. Claim 8

[8.0] *The system of claim 1, wherein: the task processing schedule parameter includes a sub-parameter configuring a start condition of at least one process of the multiple processes, or a sub-parameter configuring an allocated machine identity of at least one process of the multiple processes.*

The '518 patent explains that “the term ‘sub-parameter’ may denote a parameter including detailed items for the corresponding parameter.” Ex.1001, 4:29-31.

As described at limitation [1.2.2], Ermis describes a data flow in which the washing machine starts when the price of electricity is lowest but before the user's alarm in the morning. This data flow (*task processing schedule parameter*)

includes a *start condition* of the washing machine—when the price of electricity is at a minimum with enough time to finish before the morning alarm. See Ex.1005, [0111]-[0112]. This is described by Ermis as “the triggering criteria” which renders obvious a *start condition*. Ex.1005, [0112]. Because this triggering criteria is part of the data flow for starting the washing machine, it includes detailed items for the corresponding parameter (data flow) and is a *sub-parameter*. Ex.1003, ¶¶171-173.

Therefore, Ermis renders obvious *the task processing schedule parameter* (washing machine data flow) *includes a sub-parameter configuring a start condition of at least one process of the multiple processes* (triggering criteria configuring start time). Ex.1003, ¶174.

In addition, a MAC address is associated with each machine in Ermis’s system. A MAC address is *an allocated machine identity* and, as explained at claim 7, the MAC addresses identifying the machines required to perform a task would be included in a *task processing schedule parameter*. See §IX.C.8.[7.0]. Therefore, Ermis also renders obvious *the task processing schedule parameter* (data flow) *includes a sub-parameter configuring an allocated machine identity of at least one process of the multiple processes* (MAC address of machine configured to perform a process of a data flow). Ex.1003, ¶¶170-175.

10. Claim 9

[9.0] The system of claim 1, wherein: the relation server stores multiple machine profiles of the multiple machines.

Ermis's system 12 (*relation server*) includes system entity profiles 20.

Ex.1003, ¶176.

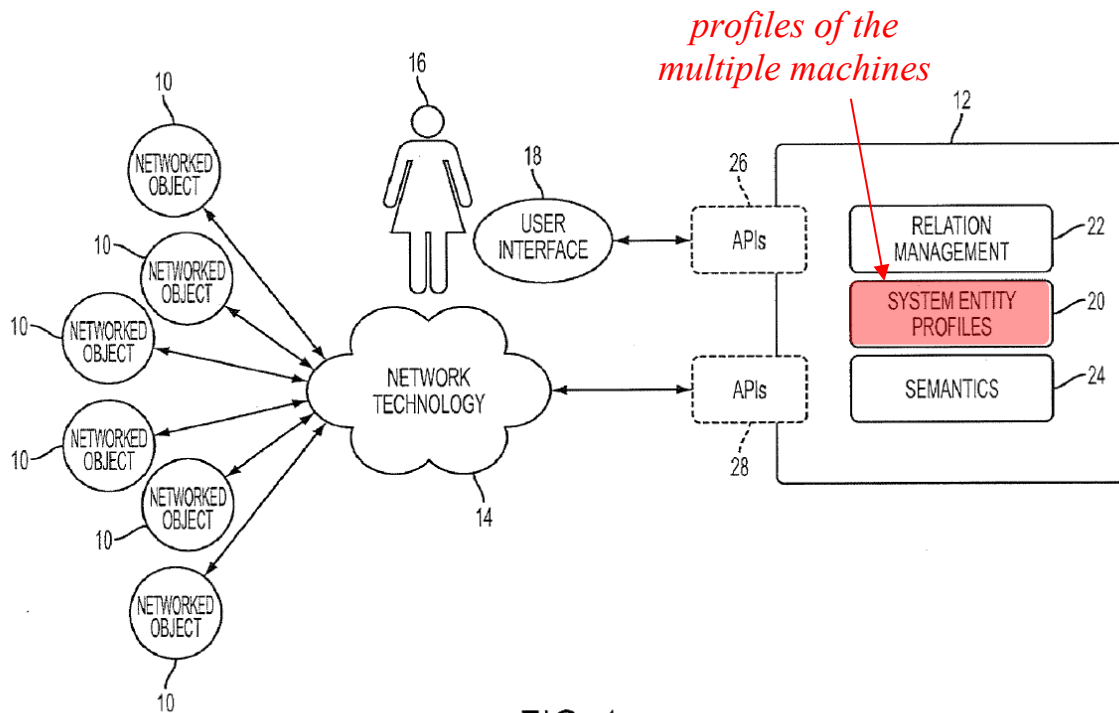


FIG. 1

Ex.1005, Fig. 1 (annotated); Ex.1003, ¶176

“[E]ach networked object is represented by a unique and identifiable profile that contains information about each object’s system characteristics.”

Ex.1005, [0038]. “The networked objects 10 can also be represented and identified in the system 12 by unique profiles.” Ex.1005, [0040]. In Figure 13, profiles are stored in the “system entity database 1304.” Ex.1005, [0099]; Ex.1003, ¶177.

Thus, Ermis renders obvious *the relation server* (system 12, resource management node 1300) *stores multiple machine profiles of the multiple machines* (stores a profile for each networked object 10, resource nodes 1330). Ex.1003, ¶¶176-178.

11. Claim 10

[10.0] *The system of claim 9, wherein: at least one machine profile of the multiple machine profiles includes a status parameter, and*

The system entity profiles 20 include *a status parameter*. “Each system entity can be made aware of **its connected system entities profiles**, including but not limited to data about their functionality, dependencies, **current status** as well as previous and future planned (timer set) events, capabilities, mandate and responsibilities.” Ex.1005, [0091]; Ex.1003, ¶179.

The status of machines is illustrated in Figure 7b. Ex.1005, [0050], [0074]; Ex.1003, ¶180.

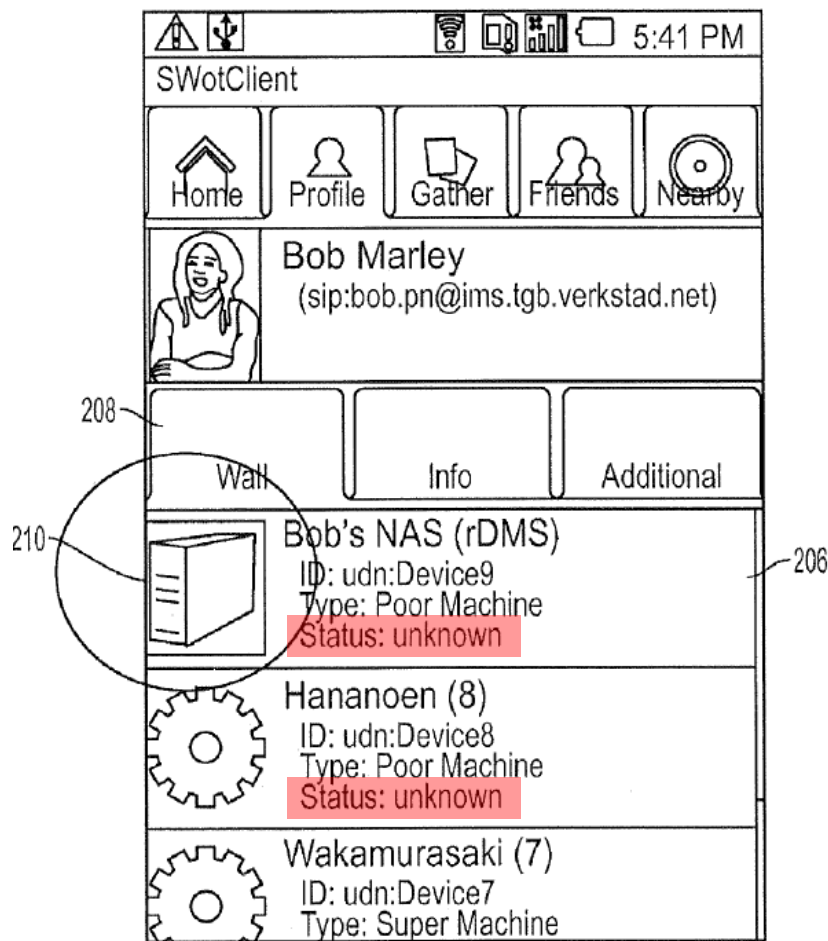


FIG. 7B

Ex.1005, Fig. 7B (annotated); Ex.1003, ¶180

The system manages a “database, list or other data structure indicating **the current status of a particular device** or personal network.” Ex.1005, [0069]. The status may include “[c]urrent bandwidth and other situational parameters, such as ambient temperature, light level, physical orientation, etc.” Ex.1005, [0069]; Ex.1003, ¶181.

Thus, Ermis renders obvious *at least one machine profile of the multiple*

machine profiles (profile for a networked object 10) *includes a status parameter* (data “indicating the current status of a particular device”). Ex.1003, ¶¶179-182.

[10.1] *the status parameter includes at least one sub-parameter of a current operation status sub-parameter, a current availability sub-parameter, a current process sub-parameter, a scheduled termination time of current process sub-parameter, a current execution function sub-parameter, or a scheduled termination time of currently executed function sub-parameter.*

Ermis describes data relating to the current operation status of the networked objects 10. For example, “[e]ach system entity can be made aware of its connected system entities profiles, including but not limited to data about their functionality, dependencies, **current status** as well as previous and future planned (timer set) events, capabilities, mandate and responsibilities.” Ex.1005, [0091]. “[D]ata about” a networked object’s “current status” renders obvious a *current operation status*. This information regarding the networked object’s “current status” would be included within the status parameter of the system entity profile 20, described above for limitation [10.0]. Ex.1003, ¶183.

Ermis also renders obvious *the status parameter includes at least one sub-parameter of...current availability sub-parameter*. Ermis also describes a “database or list **indicating which devices and user-provided services are available** in particular user situations.” Ex.1005, [0067]. Ermis provides an example in which “the VCR for some reason was unavailable.” Ex.1005, [0053]. This availability data would also be included in the status parameter of the system

entity profile and renders obvious a *current availability* for a networked object.

Ex.1003, ¶¶184-185.

Thus, Ermis renders obvious *the status parameter* (data “indicating the current status of a particular device” including a “situational parameter”) *includes at least one sub-parameter of a current operation status sub-parameter* (data corresponding to the “current status” of a “system entity”), *a current availability sub-parameter* (data corresponding to “which devices and user-provided services are available”). Ex.1003, ¶¶183-186.

12. Claim 11

[11.0] *The system of claim 9, wherein: at least one machine profile of the multiple machine profiles includes a capability parameter, and*

As explained at claim 9, “each networked object is represented by **a unique and identifiable profile**.” Ex.1005, [0038]. The profile is a “system entity profile[] 20” and “contains information about each object’s system characteristics.” Ex.1005, [0038], [0041]. “System entity information is assembled that identifies communication addresses for the resource nodes and **associated metadata identifying capabilities of the resource nodes**.” Ex.1005, [0011]; Ex.1003, ¶187.

“The system entity database 1304 contains information that identifies communication (e.g., network) addresses for each of the resource nodes 1330 of FIG. 14 and **metadata that identifies their respective capabilities** (e.g.,

functional capabilities, data input/output format(s), data content characteristics, performance characteristics, etc.).” Ex.1005, [0105]; *see also* Ex.1005, [0041] (system entity profiles 20 may include...information about...**capability**, location, history and other **metadata**), [0111] (“the system entity database 1302...identifies **capabilities** of the” machines). A networked object’s capability, included in “entity profile[] 20” is “metadata,” which renders obvious *a capability parameter* claimed. As explained at claim 4, metadata is another term for parameter. *See* §IX.C.5; Ex.1003, ¶¶188-190.

Thus, Ermis renders obvious *at least one machine profile of the multiple machine profiles* (“system entity profile[] 20”) *includes a capability parameter* (includes “metadata” “about...capability”). Ex.1003, ¶¶187-191.

[11.1] the capability parameter includes at least one sub-parameter of a title of process sub-parameter, an input parameter sub-parameter, an output parameter sub-parameter, a process processing time sub-parameter, a process processing condition sub-parameter, or a function sub-parameter.

The ’518 patent explains that “[t]he input parameter may refer to information that is input in order to operate the corresponding machine.” Ex.1001, 5:13-17.

Ermis’s capability metadata (*capability parameter*) includes input and output format parameters: “The system entity database 1304 contains...metadata that identifies their respective capabilities (e.g., functional capabilities, **data**

input/output format(s), data content characteristics, performance characteristics, etc.).” Ex.1005, [0105], [0041]. These data inputs are used by each device to perform tasks or processes of a data flow, and are therefore used “to operate the corresponding machine.” Ex.1005, [0101]-[0102]; Ex.1001, 5;13-14; Ex.1003, ¶193.

Therefore, Ermis renders obvious *the capability parameter includes at least one sub-parameter of...an input parameter sub-parameter* (an input technical specification of a resource node), *an output parameter sub-parameter* (an output technical specification of a resource node). Ex.1003, ¶194.

Ermis describes a “triggering criteria” which renders obvious a *process processing condition sub-parameter*. Ex.1005, [0100]-[0101]; Ex.1003, ¶¶195-197.

Ermis renders obvious *the capability parameter includes...a function sub-parameter*. “Any system entity associated with system 12 can be aware of **functionality** provided by networked objects 10..., e.g., the **functionality** of networked objects 10.” Ex.1005, [0053]. This functionality data would be included in an object’s system entity profile 20. *See* Ex.1005, [0052] (describing identifying friend networked objects and the “functionality...they provide”); Ex.1003, ¶¶198-199.

Therefore, Ermis also renders obvious *the capability parameter* (metadata” including “information about...capability” of a networked object 10) *includes at*

least one sub-parameter of...a function sub-parameter (“functionality” of a networked object). Ex.1003, ¶¶192-200.

13. Claim 12

[12.0] *The system of claim 9, wherein: at least one machine profile of the multiple machine profiles includes a machine identity parameter.*

Each networked object 10 includes an identity value. “Regarding the identities of the various entities described above with respect to FIGS. 4 and 5, the identity of a device 96, i.e., a networked object 10, can be for example provided by the device manufacturer, e.g., a MAC address. However, device identities could also be provided in other ways, e.g., on an IP level by a home DNS service or using explicit names, e.g., entered into the system by user configuration.”

Ex.1005, [0060]. A MAC address or name entered by the user are *machine identity parameter*[s]. Ex.1003, ¶201.

These machine identities are part of the networked object’s profile. *See* Ex.1005, [0041] (“The system entity profiles 20 may include, but are not limited to, information about name.”); Ex.1003, ¶¶202-203.

Thus, Ermis renders obvious *at least one machine profile of the multiple machine profiles* (“system entity profile[] 20” of a networked object 10) *includes a machine identity parameter* (networked object identity, such as MAC address or user-assigned name). Ex.1003, ¶¶201-204.

14. Claim 13

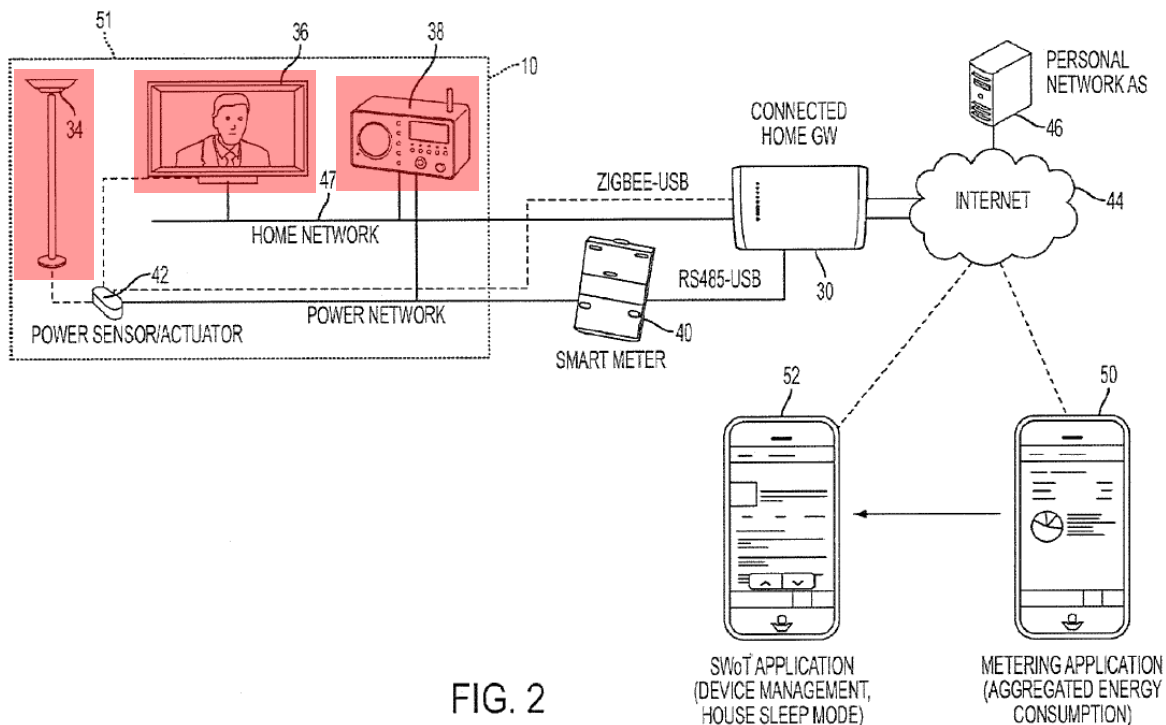
[13.0] *The system of claim 9, wherein: at least one machine profile of the multiple machine profiles includes a user identity parameter, and the user identity parameter corresponds to an identity of a user being capable of using a machine corresponding to the at least one machine profile.*

Each of Ermis's networked objects 10 includes "a unique and identifiable profile." Ex.1005, [0038]. Similarly, "[p]ersons that are using the system 12, i.e., the users 16, are represented and identified as entities in the system 12 by **unique user profiles in the system.**" Ex.1005, [0040]. "The users connect their own system entity to system entities of networked objects." Ex.1005, [0087]. In this way, a user can "connect[] his/her user profile to the device." Ex.1005, [0087]; Ex.1003, ¶¶205-206.

When a user's profile is connected to an object's profile, the "connected users and objects' system entities...**have reciprocal presence in each other's profiles.**" Ermis, [0090]. Thus, Ermis's machine profiles include an indication of a user profile of a user that is capable of using the machine (*user identity parameter*). Ex.1003, ¶¶207-209.

In another example, multiple users may be associated with a networked object. For example, "[g]roups of users could be created among users living within the same household, or other attributes that the user has decided to share." Ex.1005, [0086], [0122]. Regarding Figure 2 below, the networked objects can be

connected to “**a particular user** (or **group of users**, e.g., **a family**).” Ex.1005, [0048]. In this example, machines (“a light, 34, a television 36, and a radio 38”) may be associated with members of a group of users or family living in the home. Ex.1005, [0048]; Ex.1003, ¶210.



Ex.1005, Fig. 2 (annotated); Ex.1003, ¶210

In this example, the “metadata” of these devices, stored in their respective machine profiles, would indicate the association with multiple users. Ex.1005, [0041]. Therefore, Ermis explains that networked objects can be connected to “**a particular user**.” Ex.1005, [0048]; Ex.1003, ¶¶211-212.

Therefore, Ermis renders obvious *at least one machine profile of the*

multiple machine profiles (entity profiles of networked objects 10) *includes a user identity parameter* (include metadata corresponding to one or more user profiles of users that may use the machines). Ex.1003, ¶¶205-213.

15. Claim 14

[14.0] *The system of claim 9, wherein: at least one machine profile of the multiple machine profiles includes a group identity parameter, and the group identity parameter corresponds to an identity of a user group being capable of using a machine corresponding to the at least one machine profile.*

As explained at claim 13, Ermis describes associating a user profile with the profile of a networked object with “metadata.” *See* Ex.1005, [0048], [0087], [0041]; Ex.1003, ¶214.

As relevant to claim 14, “**[g]roups of users could be created** among users living within the same household, or other attributes that the user has decided to share.” Ex.1005, [0086], [0122]. In further detail, Ermis describes these networked objects as “a set of networked objects with which a particular user (or **group of users**, e.g., **a family**) **has a relationship**.” Ex.1005, [0048]; *see* §IX.C.14; Ex.1003, ¶215.

Thus, Ermis renders obvious *at least one machine profile of the multiple machine profiles* (entity profiles of the light 34, TV 36, or radio 38) *includes a group identity parameter* (metadata of these devices indicating association with a group of users), *and the group identity parameter corresponds to an identity of a*

user group being capable of using a machine corresponding to the at least one machine profile (each user of the “group of users, e.g., a family” “has a relationship” with the light 34, TV 36, or radio 38). Ex.1003, ¶¶214-216.

16. Claim 15

[15.0] *The system of claim 8, wherein: at least one machine profile of the multiple machine profiles includes an operating system parameter, and the operating system parameter corresponds to a type of an operating system which is used by a machine corresponding to the at least one machine profile.*

The '518 patent explains that “[t]he machine operating system parameter may include sub-parameters related to a type and a version of an operating system.” Ex.1001, 2:10-12.

In Ermis, the user 16’s mobile device “may include **an operating system.**” Ex.1005, [0076]. Also, each device has a “system entity profile[] 20,” containing “technical specifications.” Ex.1005, [0041]. An indication of the operating system is a “technical specification[]” of the user 16’s mobile device, and would be indicated as “metadata” in the profile of the device. Ex.1005, [0041]; Ex.1003, ¶218.

Ermis recognizes that networked objects 10 include different operating systems and ensures that the inputs and outputs of the various devices are interoperable. *See* Ex.1005, [0043] (“transcod[ing] data that is output by one of the resource nodes into a format that is compatible with the input of another one of the

resource nodes **to provide compatible communications.**”), [0115] (ensuring “brand-dependent semantic interaction language[s]” are interoperable). These operating systems would similarly be identified by “metadata” indicating “technical specifications” of these devices, such as the type and version number of the operating system. Ex.1005, [0041]; Ex.1003, ¶219.

Thus, Ermis renders obvious *at least one machine profile of the multiple machine profiles* (system entity profiles 20 of networked objects 10) *includes an operating system parameter, and the operating system parameter corresponds to a type of an operating system which is used by a machine corresponding to the at least one machine profile* (metadata indicating the operating system of a networked object 10 and whether its output is compatible with an input of another networked object 10). Ex.1003, ¶¶217-220.

17. Claim 16

[16.0] *The system of claim 8, wherein: at least one machine profile of the multiple machine profiles includes an interface parameter, and the interface parameter corresponds to an interface protocol between the relation server and a machine corresponding to the at least one machine profile.*

The ’518 patent explains that “[t]he machine interface parameter may include sub-parameters related to interfaces and interface protocols between the machines.” Ex.1001, 2:13-16.

Ermis explains that the networked objects 10 interface with the system 12

via APIs 26 and 28. *See* Ex.1005, [0043]. To do so, each networked object 10 includes interface data stored in its system entity profile 20.

The system entity database 1304 contains information that identifies communication (e.g., network) addresses for each of the resource nodes 1330 of FIG. 14 and **metadata that identifies their respective capabilities** (e.g., functional capabilities, **data input/output format(s)**, data content characteristics, performance characteristics, etc.).

Ex.1005, [0105]; Ex.1003, ¶222.

A device's "**metadata**" identifying "**data input/output format(s)**" is *an interface parameter*. Further this data is a "technical specification[]" included within the entity profiles for each device. Ex.1005, [0041]. Ex.1003, ¶223.

Ermis's disclosure of using "**Application Programming Interfaces (APIs) 26 and 28**" to communicate with the networked objects confirms that the profiles of the networked objects include *an interface parameter*. Ex.1005, [0043]; Ex.1003, ¶224.

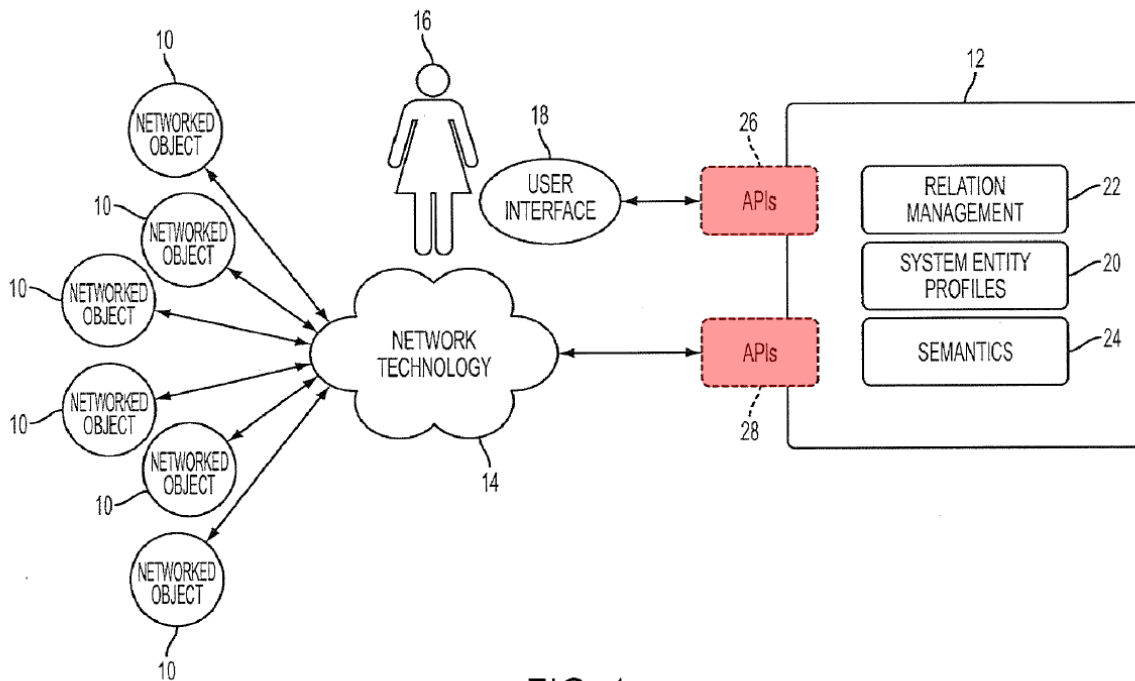


FIG. 1

Ex.1005, Fig. 1 (annotated); Ex.1003, ¶224

“The system 12 may interface with its objects 10 and users 16 through a set of Application Programming Interfaces (APIs) 26 and 28.” Ex.1005, [0044]. An API defines a set of rules and protocols that allow different devices to communicate. *See* Ex.1006, [0041] (An “API provides protocol configuration information.”); Ex.1007, [0005]-[0018] (listing multiple “API[s]” that “provide a protocol” for a number of communication functions). Because the APIs 26 and 28 facilitate communication between the system 12 and the objects 10/interface 18, they *correspond[] to an interface protocol between the relation server (system 12) and a machine corresponding to the at least one machine profile (networked*

objects 10 or application interface 18). Ex.1003, ¶¶225-226.

Thus, Ermis renders obvious *at least one machine profile of the multiple machine profiles includes an interface parameter* (system entity profile for network device includes technical specification metadata including interface data), *and the interface parameter corresponds to an interface protocol* (APIs 26 and 28 establish interface protocols) *between the relation server* (system 12) *and a machine corresponding to the at least one machine profile* (networked objects 10/interface 18 and system). Ex.1003, ¶¶221-227.

18. Claim 17

[17.0] *The system of claim 1, wherein: the relation server saves information related to grouping the multiple machines.*

As explained at claim 5, in Ermis, “it is also possible to **aggregate multiple networked objects under a common profile.**” Ex.1005, [0041]. In one example, a “group of location sensors may be identified in the system as one home location sensor.” Ex.1005, [0041], [0047] (“group of system entities”). The interaction flow database includes “information...that identifies a friend relationship between the at least two resource nodes.” Ex.1005, [0009]. In the examples above, this information indicates the machines required to perform commands (lights and surveillance camera in lights data flow example; electricitymeter, alarm clock, and washing machine in washing machine example, etc.). Ex.1005, Figs. 16-18;

Ex.1003, ¶228.

In Figure 1, this information is saved by the system 12 (*relation server*) in the “relation management function 22” which manages “the relationship between networked objects.” Ex.1005, [0038]. The system 12 also maintains “system entity profiles 20 for each networked object and user 16.” Ex.1005, [0041]. This “relation management function 22...coordinates the interactions between the networked objects 10 by applying social management principles.” Ex.1005, [0042]. In Figure 13, this information is saved in the interaction flow database 1308. Ex.1005, [0009]; Ex.1003, ¶229.

Thus, Ermis renders obvious *the relation server* (system 12 and/or resource management node 1300) *saves information related to grouping the multiple machines* (stores information regarding the group of machines required to perform a task in the relation management function 22 and/or interaction flow database 1308). Ex.1003, ¶¶228-230.

19. Claim 18

[18.0] *The system of claim 1, wherein: the relation server generates a capability set required to execute a user command, and*

After receiving a request, Ermis’s system “determine[s] device capabilities” of the networked objects 10 “in response to [the] service request.” Ex.1005, [0063]. “Determin[ing] device capabilities” needed “in response to a service

request” renders obvious *generat[ing] a capability set required to execute a user command*. See §IX.C.5; Ex.1005, [0085], [0066], [0071]-[0072], [0111]; Ex.1003, ¶231.

This process is performed by a component of the system 12. Figure 6 provides “a more detailed architectural schematic for implementing management of networked objects,” such as the system 12 and includes a “context manager 110” which performs the process of “determin[ing] device capabilities.” Ex.1005, [0019], [0085], [0063]; Ex.1003, ¶232.

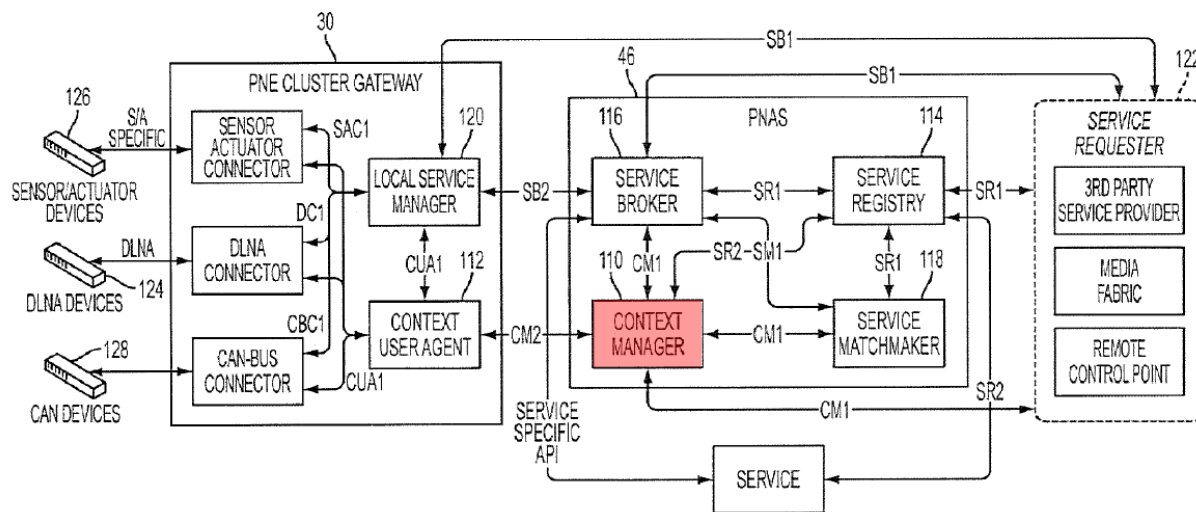


FIG. 6

Ex.1005, Fig. 6 (annotated); Ex.1003, ¶232

Thus, Ermis renders obvious *the relation server* (context manager 110 of system 12) *generates a capability set required to execute a user command*

(“determine[s] device capabilities...in response to a service request”). Ex.1003, ¶¶231-233.

[18.1] generates the task processing schedule parameter according to the capability set, a status parameter of at least one machine of the multiple machines, and a capability parameter of the at least one machine.

Ermis explains that a data flow is generated according to the capabilities of the devices. In the example of recording a TV program, the system 12 first determines that “the TV is in fact not capable of recording anything,” but “another networked object 10,” the “video recorder,” “can perform the requested service.” Ex.1005, [0045]-[0046]. Then, the system 12 generates a data flow in which the TV entity within the system 12 “relay[s] the command to the system entity of the video recorder.” Ex.1005, [0046]; Ex.1003, ¶234.

As explained at limitation [1.2.2], data flows (*task processing schedule parameter[s]*) are generated by “[t]he interaction flow database 1308” which “**(assembles) information that defines data flows.**” Ex.1005, [0100]-[0101]. The interaction flow database 1308 is a part of the relation management node 1300 (*relation server*). As explained at claim 4, the capabilities of these devices are included in “**metadata identifying capabilities of the resource nodes,**” which renders obvious a *capability set parameter*. Ex.1005, [0008]; Ex.1003, ¶235.

As explained at claim 10, the system entity 20 for a device includes a *status parameter*. See Ex.1005, [0091], [0096]-[0097]. A data flow is generated in

response to a request and based on the current status of the devices with capabilities to perform the request. In the example of recording a TV program, Ermis explains that if the status of the video recorder is “unavailable,” the data flow may be modified to use another device to perform the recording process. Ex.1005, [0053]; Ex.1003, ¶¶236-237.

In addition, as explained at claim 4, the devices of a particular data flow are selected based on their respective capabilities. Specifically, the relation server “assigns the requested task to one of the plurality of networked objects which **has a capability to perform the requested task**.” Ex.1005, [0085], [0008]; Ex.1003, ¶238.

Thus, Ermis renders obvious *the relation server (system 12)...generates the task processing schedule parameter (generates a data flow) according to the capability set (after determining the capabilities of the networked objects 10 based on the capability metadata), a status parameter of at least one machine of the multiple machines (selecting devices of a data flow based on their availability), and a capability parameter of the at least one machine (selecting devices of a data flow based on their capabilities)*. Ex.1003, ¶¶234-239.

20. Claim 19

[19.0] The system of claim 1, wherein: the intervention includes selecting some processes of the multiple processes.

As explained at limitation [1.3] above, Ermis describes *an intervention by a user in the relation profile*. See §IX.C.2.[1.3]. Specifically, the rental car in Figure 24 requests that the user allow new relationships to be generated between the rental car and “a media server device and a phone terminal” as well as “a map application, and a navigator application.” Ex.1005, [0126], Fig. 24; Ex.1003, ¶240.

The user may select the “Add All 4...” button or may select some but not all of the recommended data flows. Ex.1005, Fig. 24, [0128] (User can “select[] the ‘Add some...’ button.”); Ex.1003, ¶241.

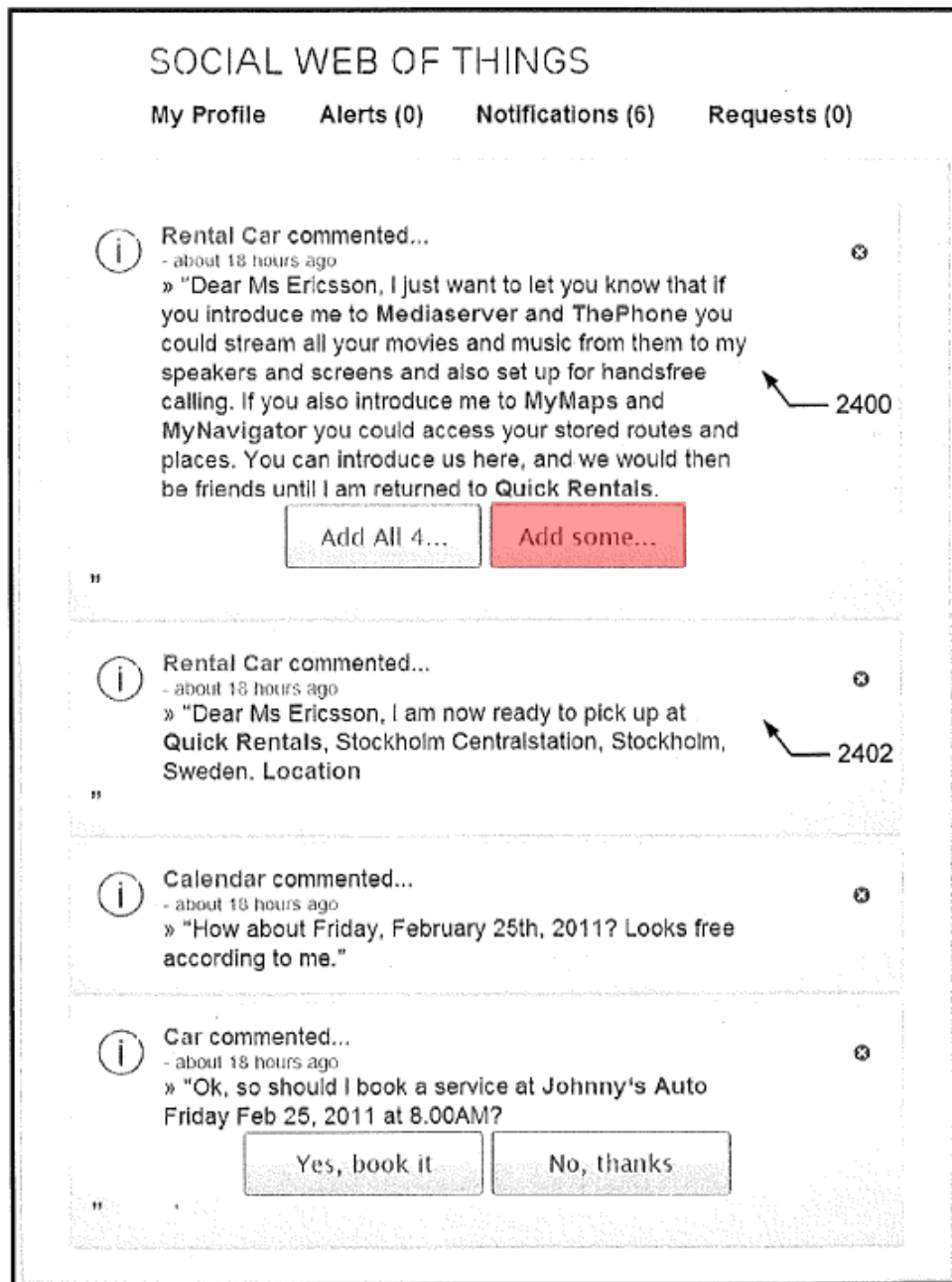


FIG. 24

Ex.1005, Fig. 24 (annotated); Ex.1003, ¶241

If the user selects the “Add some...” button, “[t]he interaction execution engine 1312 adds the selected friendship relationships and associated data flows to

the interaction flow database 1308.” Ex.1005, [0128]; Ex.1003, ¶¶242-243.

In the lighting example of Figure 17, two processes of a data flow are recommended: (1) turning off the lights while the user is away and (2) viewing the surveillance camera footage. Ex.1003, ¶244.

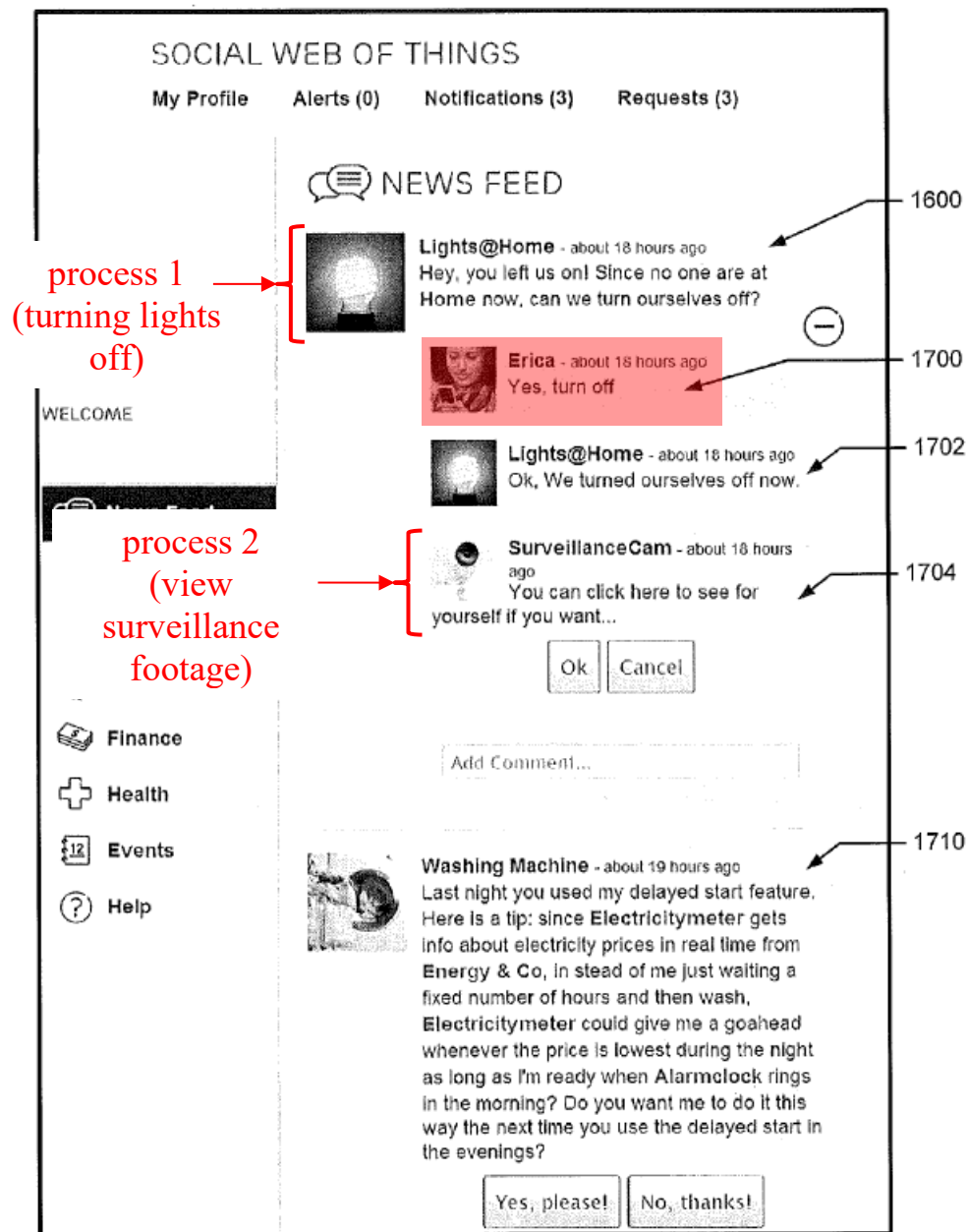


FIG. 17

Ex.1005, Fig. 17 (annotated); Ex.1003, ¶244

The user can approve or not approve either process in this data flow. Above, the user approved process 1 of the recommended data flow (turning the lights off)

at step 1700, but not process 2 (viewing surveillance footage). These examples render obvious a user *intervention* that *includes selecting some processes of the multiple processes*. Ex.1003, ¶¶240-245.

21. Claim 20

[20.0] *The system of claim 1, wherein: the intervention includes approving the multiple processes.*

As described at limitation [1.3], Figure 24 illustrates creating four new relationships. Ex.1005, [0126]; Fig. 24; Ex.1003, ¶¶246-247.

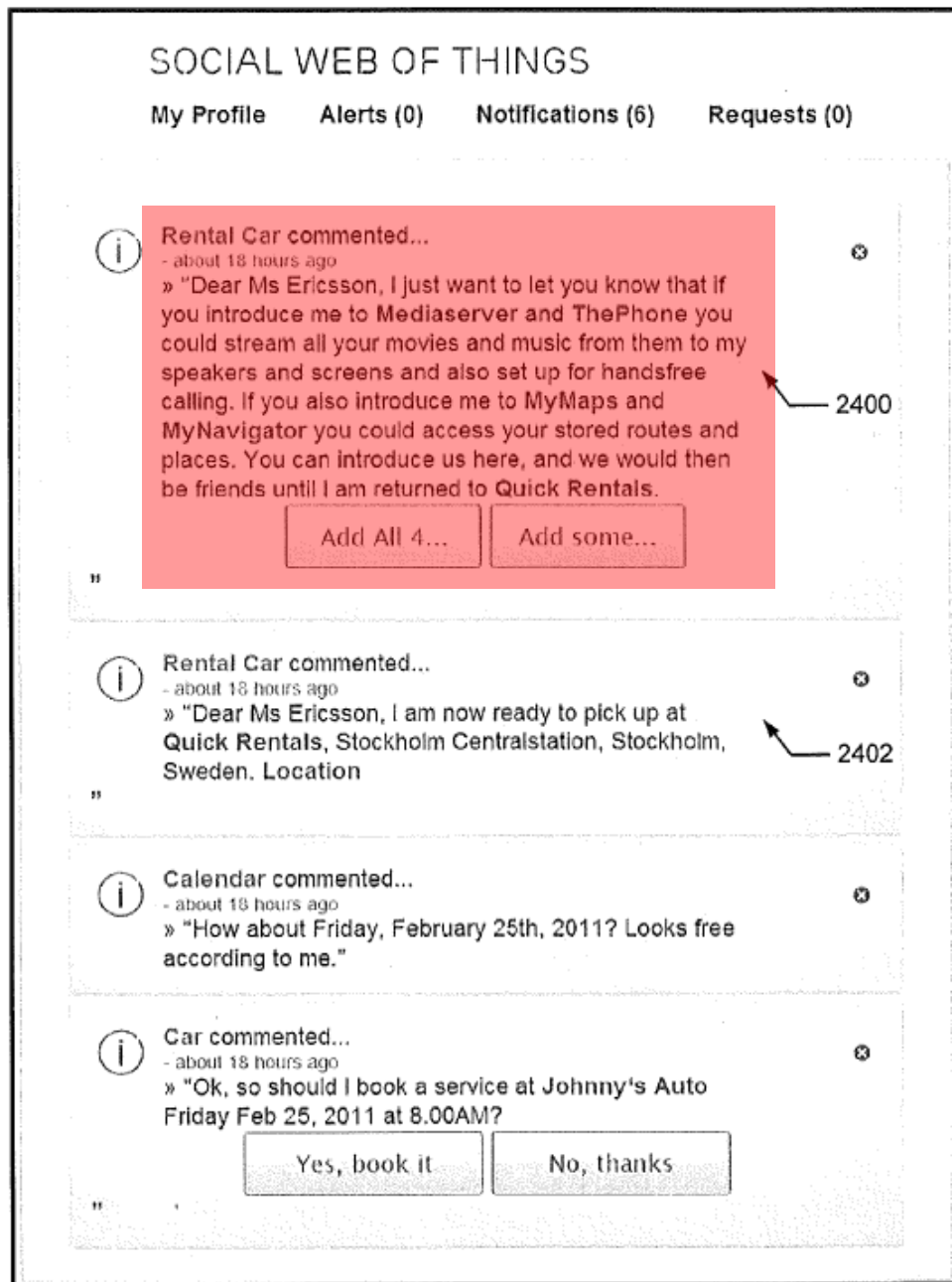


FIG. 24

Ex.1005, Fig. 24 (annotated); Ex.1003, ¶247

These new relationships and associated data flows (“stream[ing]...movies and music” and “access[ing] stored routes and places”) are added based on the

user's selection of the "Add all 4..." button, which approves the processes of the data flows. Ex.1005, [0128], Fig. 24; Ex.1003, ¶248.

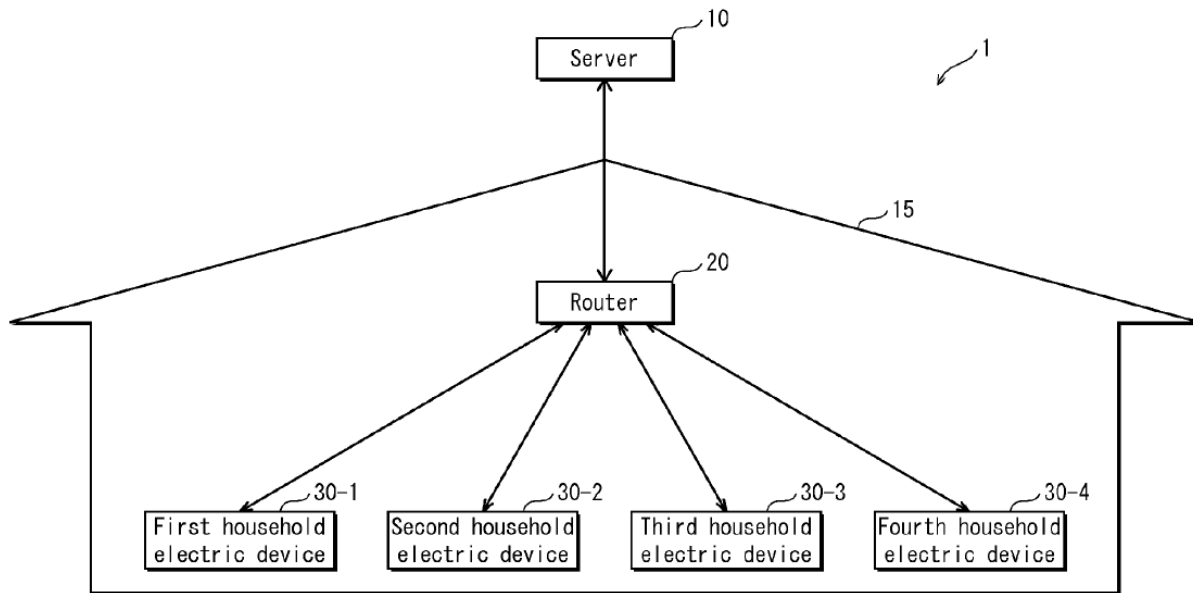
Thus, Ermis renders obvious *the intervention includes approving the multiple processes* (user selecting the "Add All 4..." button to implement recommended data flows). Ex.1003, ¶¶246-249.

D. Ground 2: Claims 1-20 are obvious under §103 over Ermis in view of Nakano.

1. Nakano

Like the '518 patent and Ermis, Nakano describes principles of home automation. Ex.1023, (54), Abstract. Nakano describes multiple machines, referred to as "household electric devices," in communication with "a server 10," shown in Figure 1. Ex.1023, 3:63-67.

FIG. 1



Ex.1023, Fig. 1

Figure 3, below, illustrates profile information stored for each of the household electric devices. For example, each device has a “device ID,” a “product code,” and the operating software running on the device indicated by “a version number.” Ex.1023, 6:37-43. This information is stored in a “Device Information Table,” shown below.

FIG. 3

	Device ID (Device name)	Product code (Product type)	Software version	Operation ID (Operation name)	Control command	Regist- ration state
311 →	First household electric device	P-06D (Smartphone)	1. 12	A1 (Power ON)	D201xxxx	1 (Registered)
				A2 (Alarm)	D205xxxx	
				⋮	⋮	
312 →	Second household electric device	CS-X252C (Air conditioner)	1. 01	B1 (Power ON)	9203xxxx	1 (Registered)
				B2 (High-power mode operation)	9252xxxx	
				B3 (Power-saving mode operation)	9280xxxx	
				⋮	⋮	
313 →	Third household electric device	SR-SX102 (Rice cooker)	1. 03	C1 (Power ON)	800Fxxxx	1 (Registered)
				C2 (Rice cooking start)	8020xxxx	
				⋮	⋮	
314 →	Fourth household electric device	NA-VX8200L (Washing machine)	1. 10	D1 (Power ON)	9101xxxx	1 (Registered)
				D2 (Washing start)	9104xxxx	
				⋮	⋮	
315 →	Fifth household electric device	DMC-TZ40 (Digital camera)	1. 00	E1 (Power ON)	A209xxxx	0 (Not registered)
				E2 (Image capture)	A285xxxx	
				⋮	⋮	
316 →	Sixth household electric device	HH-LC712A (Ceiling light)	1. 01	F1 (Light up)	E942xxxx	0 (Not registered)
				F2 (Cinema mode)	EA94xxxx	
				F3 (Power-saving mode)	E548xxxx	
				F4 (ON timer)	E968xxxx	
317 →	Seventh household electric device	BG-849 (Air conditioner)	2. 10	G1 (Power ON)	3924xxxx	0 (Not registered)
				G2 (ON timer)	9643xxxx	
				⋮	⋮	
318 →	Eighth household electric device	COF-63B (Coffee maker)	1. 09	H1 (Power ON)	5829xxxx	0 (Not registered)
				H2 (Brewing start)	6487xxxx	
				H3 (Timer brewing)	935Exxxx	
				⋮	⋮	
319 →	Ninth household electric device	TH-P65VT60 (Television)	1. 01	J1 (Power ON)	6385xxxx	0 (Not registered)
				⋮	⋮	
	⋮	⋮	⋮	⋮	⋮	⋮

Ex.1023, Fig. 3

Nakano's system also uses information identifying devices to form "cooperative processes" included in a "cooperative process composition table," in Figure 4, below. Ex.1023, 8:4-11.

	First cooperative process		Second cooperative process		Third cooperative process		Fourth cooperative process		Fifth cooperative process		Sixth cooperative process		Seventh cooperative process		Eighth cooperative process		...
	Device ID	Operation ID	Device ID	Operation ID	Device ID	Operation ID	Device ID	Operation ID	Device ID	Operation ID	Device ID	Operation ID	Device ID	Operation ID	Device ID	Operation ID	...
First operation	1	A2	1	A2	1	A2	1	A2	1	A2	2	B1	6	F1	6	F1	...
Second operation	2	B3	2	B2	2	B3	7	G2	7	G2	6	F1	2	B3	2	B1	...
Third operation	3	C2	3	C2	3	C2	3	C2	3	C2					9	J1	...
Fourth operation			8	H2	4	D2	8	H2	4	D2							...

411
412

Ex.1023, Fig. 4

"[A] cooperative process refers to a group of operations linked together that are each executed by a different one of a plurality of household electric devices." Ex.1023, 3:46-49. Each cooperative process lists the "device ID of a household electric device executing the operation and an operation ID identifying the operation." Ex.1023, 8:17-20.

2. Reasons to Combine Ermis and Nakano

A POSITA would have been motivated to combine the teachings of Ermis with Nakano. Before the priority date of the '518 patent, it would have been obvious, beneficial, and predictable to implement Nakano's teachings of using data tables to store information related to a task or device. A POSITA would have recognized these techniques as well-known techniques to ensure proper operation and coordination between devices to perform various task requests. Ex.1003, ¶255.

a) Analogous art

A POSITA considering the teachings of Ermis would have naturally considered the teachings of Nakano, because Nakano is analogous prior art to both the '518 patent and Ermis, namely, “managing **relations** between machines...**to execute a command**.” Ex.1001, Abstract; Ex.1005, [0044] (“establish[ing] **relationships**...and...exercis[ing] those relationships in **performance of various task requests**.”); Ex.1023, 3:50-55 (managing a “**cooperative process...to execute an operation**”). Both Ermis and Nakano describe managing relations between multiple devices and controlling the multiple devices to execute processes of a task. *See* Ex.1005, [0101]-[0102] (“The interaction execution engine 1312 can...**control operations performed by the resource nodes**.”); Ex.1023, 3:50-55 (“**controlling each of one or more household electric devices to execute an operation**.”); Ex.1003, ¶¶256-257.

b) Motivation to combine Ermis with Nakano

A POSITA would have been motivated to implement Nakano’s teachings regarding storing data for cooperative processes in cooperative process composition tables within Ermis’s system. A POSITA would have recognized that data related to operations performed by multiple devices to execute a task would advantageously be stored as part of a cooperative process composition table, as described by Nakano. Ex.1003, ¶258.

Storing this data as a structured data table would advantageously allow for efficient querying using common database languages, such as XML referenced by Ermis. *See* Ex.1005, [0070]. This format ensures that queries easily retrieve all operations for a device, all devices involved in a task, or sequences of operations to perform a task. Ex.1003, ¶259 (citing Ex.1032, 1:39-42; Ex.1033, [0018]). This format would additionally enforce constraints on the data stored, ensuring that only valid operations are assigned to devices. Ex.1003, ¶259 (citing Ex.1034, [0127]). Implementing Nakano's cooperative process composition table increases scalability and flexibility which would be particularly useful in Ermis's system which describes sharing popular data flows with different users. Ex.1003, ¶259 (citing Ex.1035, [0223]). Nakano's cooperative process composition table also optimizes performance of indexing and caching by making lookups more efficient. Ex.1003, ¶259 (citing Ex.1036, [0047]). The consistent structure of Nakano's cooperative process composition table would also increase interoperability and integration. Ex.1003, ¶259 (citing Ex.1035, [0023], Ex.1037, [0060]).

Accordingly, the Ermis-Nakano combination would have been the combination of prior art elements (identifying multiple processes for a requested task) according to known methods (storing the data for the multiple processes in a table including device IDs and operation IDs) to yield the predictable result of efficiently coordinating multiple networked devices to execute a task. *KSR Int'l*

Co. v. Teleflex Inc., 550 U.S. 398, 416 (2007); Ex.1003, ¶260.

A POSITA would also have been motivated to incorporate Nakano’s teachings regarding identifying an operating system using a product code and software version because it was a well-known method of indicating operating system characteristics and identifying whether the operating system is updated and capable of performing as expected. Ex.1003, ¶261.

Because a device’s operating system is assigned a new “number greater than that given to the older version,” when it is updated, a user, manufacturer, or service provider may easily determine whether the operating system of the device is updated and capable to install applications or software or perform tasks unique to the operating system version. Ex.1023, 6:7-10; Ex.1003, ¶262 (Ex.1038, Abstract, [0006]).

Implementing Nakano’s teachings of indicating an operating system by a product code and version number ensures simply identification of interoperability of a device. Ex.1003, ¶263 (citing Ex.1039, Abstract).

Ermis’s devices include operating systems. *See* Ex.1005, [0076] (describing “an operating system” in the “storage/memory 704” of a device). Ermis also explains that each device includes a “system entity profile[] 20,” which contains a “technical specification[]” of the device, including “metadata” of any suitable type. Ex.1005, [0041]. A POSITA would have recognized that because each device

includes an operating system and metadata describing characteristics of the device, this metadata would include an operating system parameter. *See* §IX.C.16.

However, to the extent Ermis does not explicitly describe a specific parameter indicative of the device operating system, a POSITA would have been motivated to implement Nakano's techniques indicating a device's operating system with a product code and version number that align with Ermis's system. Ex.1003, ¶264.

Both Ermis and Nakano list interoperability of devices as an objective. *See* Ex.1005, [0043] ("using transcoding information" for outputs of devices "**to provide compatible communications**" with other devices), [0115]; Ex.1023, 3:58-62, 15:1-5. A POSITA would have been motivated to incorporate Nakano's use of product codes and software version numbers to improve Ermis "in the same way." *KSR*, 550 U.S. at 417; Ex.1003, ¶¶265-267.

c) Reasonable expectation of success

A POSITA also would have reasonably expected success in making the combination due to the compatibility and similarity of their teachings. Ex.1003, ¶268. Both Ermis and Nakano describe controlling interconnected machines to perform commands. Ex.1005, [0044] ("exercis[ing]...relationships in performance of various task requests."); Ex.1023, 3:50-55 ("controlling...household electric devices to execute an operation."); Ex.1003, ¶268.

Both Ermis and Nakano use standard, commercially available devices. *See*

Ex.1005, [0045]-[0046] (a TV and video recorder), [0111] (a washing machine, electricity controller, and alarm clock); Ex.1023, 3:63-4:9 (a smartphone, air conditioner, rice cooker, and a washing machine). Both use a server to control these devices. *See* Ex.1005, [0040]; Ex.1023, 4:10-14. Based on these teachings, a POSITA would have been familiar with the aspects of Ermis's system and recognized its compatibility with Nakano's. Ex.1003, ¶269.

Accordingly, a POSITA would have been motivated and found it obvious to combine Ermis with Nakano in the manner proposed. Ex.1003, ¶270.

3. Claim 1

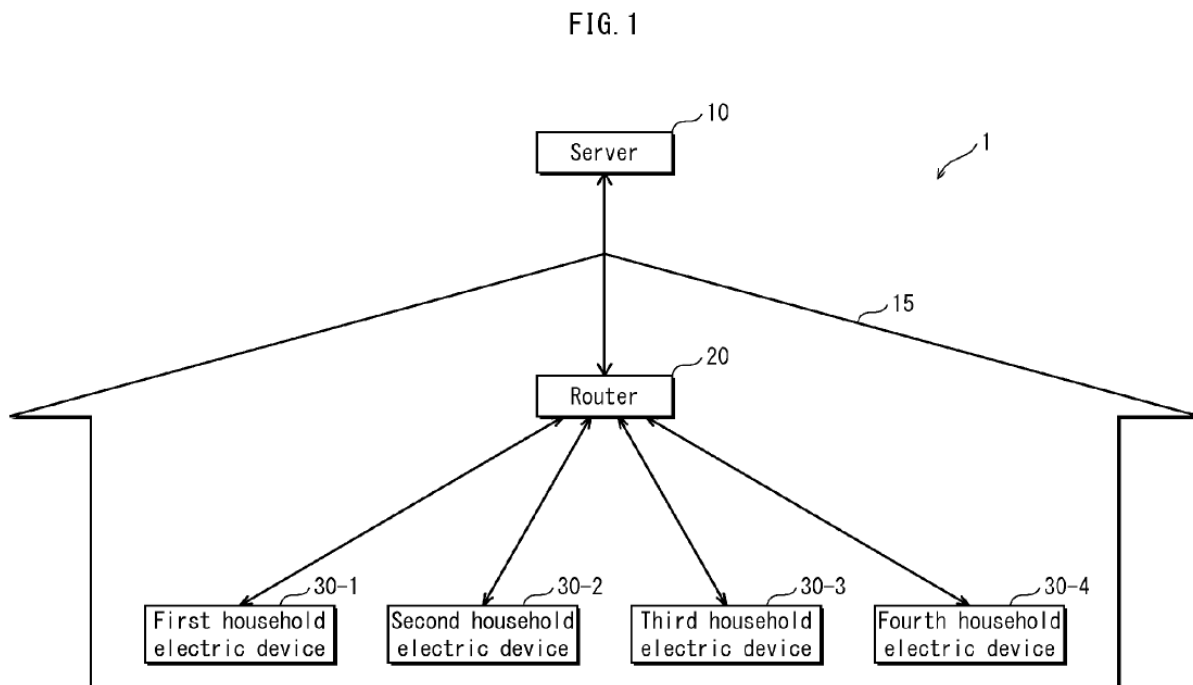
Limitations [1.0]-[1.2.2] and [1.2.4]-[1.5] are rendered obvious for the reasons described in Ground 1 above. *See* §IX.C.2; Ex.1003, ¶271.

[1.2.3] including a task processing schedule parameter which defines a sequence of performing multiple processes,

As explained above, *see* §IX.C.2.[1.2.3], Ermis describes data flows which define a series of steps to be performed by multiple devices to execute a requested task. For example, a washing machine, electricity controller, and alarm clock perform various tasks to determine a start time for a washing machine. *See* Ex.1005, [0111]. Ermis provides several additional examples of data flows including a sequence of processes performed by multiple devices. *See* [0045]-[0046], [0053]-[0054], [0048], [0124]; Ex.1003, ¶¶272-273.

Nakano further confirms that it was well known for a parameter to define multiple processes to be performed by multiple devices, e.g., *a task processing schedule parameter*. Ex.1003, ¶274.

Nakano describes methods for coordinating multiple “household electric devices that are registered to a server to execute **cooperative processes**.” Ex.1023, Abstract. These electric devices 30 are shown in Figure 1 in communication with a server 10. Ex.1003, ¶275.



Ex.1023, Fig. 1

“[A] cooperative process refers to **a group of operations** linked together that are each **executed by a different one of a plurality of household electric devices**.” Ex.1023, 3:46-49. Cooperative processes are illustrated in Figure 4,

below. Ex.1003, ¶276.

FIG. 4

task processing schedule parameter[s]

	First cooperative process		Second cooperative process		Third cooperative process		Fourth cooperative process		Fifth cooperative process		Sixth cooperative process		Seventh cooperative process		Eighth cooperative process		...
	Device ID	Operation ID	Device ID	Operation ID	Device ID	Operation ID	Device ID	Operation ID	Device ID	Operation ID	Device ID	Operation ID	Device ID	Operation ID	Device ID	Operation ID	...
First operation	1	A2	1	A2	1	A2	1	A2	1	A2	2	B1	6	F1	6	F1	...
Second operation	2	B3	2	B2	2	B3	7	G2	7	G2	6	F1	2	B3	2	B1	...
Third operation	3	C2	3	C2	3	C2	3	C2	3	C2					9	J1	...
Fourth operation			8	H2	4	D2	8	H2	4	D2							...

411 412

Ex.1023, Fig. 4; Ex.1003, ¶276

As illustrated above, each cooperative process lists the “device ID of a household electric device executing the operation and operation ID identifying the operation.” Ex.1023, 8:17-20; Ex.1003, ¶277.

In Nakano, cooperative processes define a *sequence of multiple processes*. With reference to Figure 4, the “first operation” in each process “triggers the cooperative process” and the rest of the operations are referred to as “remaining processing.” Ex.1023, 8:4-16. With reference to Figure 9, below, each operation is assigned to a “[s]pecification of time.” Ex.1005, Fig. 9; Ex.1003, ¶278.

sequence of performing multiple processes

Execution control number	Specification of time (Parameter)	Device ID	Operation ID
1	7:00	2	B2
	7:00	3	C2
2	6:55	2	B2
	7:05	3	C2
	7:10	4	D2
.	.	.	.
.	.	.	.
.	.	.	.

Ex.1023, Fig. 9; Ex.1003, ¶278

Because the time parameters of each operation are scheduled to occur one after another (e.g., 6:55, 7:05, and 7:10), a cooperative process in Nakano defines a *sequence of multiple processes*. Ex.1003, ¶279.

It would have been obvious to structure and store a “data flow,” described by Ermis, in a manner similar to Nakano’s “cooperative process.” As explained above, storing information regarding tasks to be performed by multiple devices as processes in a table offers several advantages, including more efficient querying, data consistency, scalability, flexibility, and interoperability. *See* §IX.D.2 (Reasons to Combine Ermis and Nakano); Ex.1003, ¶280.

Thus, the combination of Ermis and Nakano renders obvious *a task*

processing schedule parameter (cooperative process) which defines a sequence of performing multiple processes (a sequence of operations performed by various devices). Ex.1003, ¶¶272-281.

4. Claims 2-14

Claims 2-14 are rendered obvious for the reasons described in Ground 1 above. *See* §§IX.C.3-15; Ex.1003, ¶282.

5. Claim 15

[15.0] *The system of claim 8, wherein: at least one machine profile of the multiple machine profiles includes an operating system parameter, and the operating system parameter corresponds to a type of an operating system which is used by a machine corresponding to the at least one machine profile.*

As explained in Ground 1, Ermis renders obvious claim 15. Alternatively, the combination of Ermis and Nakano also renders obvious claim 15 as shown below. Ex.1003, ¶283.

Nakano describes data associated with various “household electric device[s].” Nakano, Fig. 3. This data includes a “[p]roduct code” and a “[s]oftware version,” highlighted below. Ex.1003, ¶284.

FIG. 3

	Device ID (Device name)	Product code (Product type)	Software version	Operation ID (Operation name)	Control command	Regist- ration state
311 →	First household electric device	P-06D (Smartphone)	1. 12	A1 (Power ON) A2 (Alarm) ⋮	D201xxxx D205xxxx ⋮	1 (Registered)
312 →	Second household electric device	CS-X252C (Air conditioner)	1. 01	B1 (Power ON) B2 (High-power mode operation) B3 (Power-saving mode operation) ⋮	9203xxxx 9252xxxx 9280xxxx ⋮	1 (Registered)
313 →	Third household electric device	SR-SX102 (Rice cooker)	1. 03	C1 (Power ON) C2 (Rice cooking start) ⋮	800Fxxxx 8020xxxx ⋮	1 (Registered)
314 →	Fourth household electric device	NA-VX8200L (Washing machine)	1. 10	D1 (Power ON) D2 (Washing start) ⋮	9101xxxx 9104xxxx ⋮	1 (Registered)
315 →	Fifth household electric device	DMC-TZ40 (Digital camera)	1. 00	E1 (Power ON) E2 (Image capture) ⋮	A209xxxx A285xxxx ⋮	0 (Not registered)
316 →	Sixth household electric device	HH-LC712A (Ceiling light)	1. 01	F1 (Light up) F2 (Cinema mode) F3 (Power-saving mode) F4 (ON timer)	E942xxxx EA94xxxx E548xxxx E968xxxx	0 (Not registered)
317 →	Seventh household electric device	BG-849 (Air conditioner)	2. 10	G1 (Power ON) G2 (ON timer) ⋮	3924xxxx 9643xxxx ⋮	0 (Not registered)
318 →	Eighth household electric device	COF-63B (Coffee maker)	1. 09	H1 (Power ON) H2 (Brewing start) H3 (Timer brewing) ⋮	5829xxxx 6487xxxx 935Exxxx ⋮	0 (Not registered)
319 →	Ninth household electric device	TH-P65VT60 (Television)	1. 01	J1 (Power ON) ⋮	6385xxxx ⋮	0 (Not registered)
	⋮	⋮	⋮	⋮	⋮	⋮

Ex.1023, Fig. 3 (annotated); Ex.1003, ¶284

“[T]he device information table” of Figure 3 includes “(ii) a product code, [and] (iii) a software version” for each device. Ex.1023, 5:44-58. “The product code is an identifier given to the household electric device by the manufacturer.” Ex.1023, 5:63-65. “The software version is a number indicating a version of software (can also be firmware, middleware, or the like) for controlling the household electric device, and is given to the software by a software developer or the like.” Ex.1023, 6:4-10; Ex.1003, ¶285.

Nakano explains that this data is included in the product’s “device operation table.” Ex.1023, 9:29-31. That the product code and software version listed in the operation table refers to an operating system parameter is confirmed by the types of devices described by Nakano, such as “a smartphone” which commonly includes an operating system. Ex.1023, 10:3-8; Ex.1003, ¶286.

While Ermis explains that its networked objects execute an “operating system,” to the extent Ermis does not explicitly disclose a parameter indicative of these various operating systems of devices, it would have been obvious in view of the teachings of Nakano. Ex.1005, [0076]. Like the product code and software version parameters described by Nakano, a type and version of an operating system of an electronic device was conventionally indicated by a product code and software version number. Ex.1003, ¶287 (citing Ex.1024, 3:11-17; Ex.1025, 4:42-47, Fig. 2; Ex.1026, [0009]). In the combination, this product code and software

version number data would be stored in the system entity profile of the device. *See* Ex.1005, [0041]; Ex.1003, ¶287.

Thus, the combination of Ermis and Nakano renders obvious *at least one machine profile of the multiple machine profiles* (system entity profile 20) *includes an operating system parameter, and the operating system parameter corresponds to a type of an operating system which is used by a machine corresponding to the at least one machine profile* (indication of product code and software version number). Ex.1003, ¶¶283-288.

6. Claims 16-20

Claims 16-20 are rendered obvious for the reasons described in Ground 1 above. *See* §§IX.C.17-21; Ex.1003, ¶289.

X. DISCRETIONARY DENIAL WOULD BE INAPPROPRIATE

A. No basis for *Fintiv* denial

The *Fintiv* factors favor institution. *Apple Inc. v. Fintiv, Inc.*, IPR2020-00019, Paper 11 (Mar. 20, 2020) (precedential).

Factor 1: No motion to stay has been filed, and thus this factor is neutral.

Factor 2: The Board will likely issue a Final Written Decision in July/August of 2026, and the district court trial date is currently set for March 16, 2026. Ex.1018, 1. The estimated trial date based on time-to-trial statistics is June 2026. Ex.1019, 5 (23 months). This factor is neutral because the trial date is

speculative and subject to change. *See* Ex.1020, 8 (Memo) (“scheduled trial dates are unreliable and often change”). For example, Judge Gilstrap has *eleven* trials set for March 16, 2026. Ex.1040. Further, a Motion to Dismiss for Lack of Venue is currently pending before the Court.

Factor 3: The co-pending litigation is in its early stages, and investment has been minimal. The claim construction hearing is not until September 16, 2025, fact discovery is not set to close until October 17, 2025, and expert discovery is not set to close until December 4, 2025. Ex.1018, 3-4. Moreover, Petitioner only learned which claims were being asserted on November 14, 2024, and has worked expeditiously to file this petition within three months. Ex.1041 (infringement contentions cover pleading). This factor favors institution.

Factor 4: Final invalidity theories will not be submitted until March 11, 2026, so the extent of overlap is currently speculative. If Patent Owner raises a *Fintiv* challenge in its preliminary response, Petitioner will evaluate whether a stipulation is appropriate based on the state of the district court proceeding at that time. This factor is neutral.

Factor 5: Petitioner is a defendant in the co-pending litigation, and thus this factor is neutral.

Factor 6: This Petition presents compelling evidence of unpatentability, and thus denial is inappropriate. Ex.1020 (Memo). The Petition—supported by Dr.

Houh's expert testimony—plainly shows that the '518 patent claims no more than what was already known or obvious.

In sum, every *Fintiv* factor is either neutral or weighs against denial.

B. No basis for §325(d) denial

Denial under §325(d) is not warranted because the challenges presented in this petition are neither cumulative nor redundant to the prosecution of the '518 patent. The Examiner did not consider any of the references applied in the above grounds.

XI. CONCLUSION

Petitioner has established a reasonable likelihood that the Challenged Claims are unpatentable.

Respectfully submitted,

Dated: February 26, 2025
HAYNES AND BOONE, LLP
2801 N. Harwood St., Suite 2300
Dallas, Texas 75201
Customer No. 27683

/Scott T. Jarratt/
Scott T. Jarratt
Lead Counsel for Petitioner
Registration No. 70,297

XII. MANDATORY NOTICES

A. Real Party-in-Interest

Pursuant to 37 C.F.R. § 42.8(b)(1), Petitioner certifies that the real parties-in-interest are Amazon.com, Inc., Amazon.com Services LLC, and Amazon Web Services, Inc.

B. Related Matters

Pursuant to 37 C.F.R. § 42.8(b)(2), to the best knowledge of the Petitioner, the '518 patent is or was involved in the following case:

Case Heading	Number	Venue	Filed
<i>KAIFI LLC v. Amazon.com, Inc. et al</i>	2:24-cv-00542	EDTX	Jul. 17, 2024

C. Lead and Back-up Counsel and Service Information

Lead Counsel

Scott T. Jarratt
HAYNES AND BOONE, LLP
2801 N. Harwood St., Suite 2300
Dallas, TX 75201

Phone: (972) 739-8663
Fax: (214) 200-0853
scott.jarratt.ipr@haynesboone.com
USPTO Reg. No. 70,297

Back-up Counsel

Brian Kwok
HAYNES AND BOONE, LLP
2801 N. Harwood St., Suite 2300
Dallas, TX 75201

Phone: (415) 293-8916
Fax: (214) 200-0853
brian.kwok.ipr@haynesboone.com
USPTO Reg. No. 58,828

Adam L. Erickson
HAYNES AND BOONE, LLP
2801 N. Harwood St., Suite 2300
Dallas, TX 75201

Phone: (202) 654-4531
Fax: (214) 200-0853
adam.erickson.ipr@haynesboone.com
USPTO Reg. No. 77,434

Please address all correspondence in this proceeding to lead and back-up counsel. Petitioner consents to service in this proceeding by email at the addresses above.

CERTIFICATE OF WORD COUNT

Pursuant to 37 C.F.R. § 42.24(d), Petitioner hereby certifies, in accordance with and reliance on the word count provided by the word-processing system used to prepare this Petition, that the number of words in this paper is 13,954. Pursuant to 37 C.F.R. § 42.24(d), this word count excludes the table of contents, table of authorities, mandatory notices under § 42.8, certificate of service, certificate of word count, appendix of exhibits, and any claim listing.

Dated: February 26, 2025

/Scott T. Jarratt/
Scott T. Jarratt
Lead Counsel for Petitioner
Registration No. 70,297

CERTIFICATE OF SERVICE

The undersigned certifies that, in accordance with 37 C.F.R. § 42.6(e) and 37 C.F.R. § 42.105, service was made on Patent Owner as detailed below.

Date of service February 26, 2025

Manner of service FEDERAL EXPRESS

Documents served Petition for *Inter Partes* Review Under 35 U.S.C. § 312 and 37 C.F.R. § 42.104 of U.S. 11,082,518; Petitioner's Power of Attorney; Petitioner's Exhibit List; Exhibits 1001-1011, 1013-1021, 1023-1041.

Persons served Broadview IP Law, PC
3200 El Camino Real, Suite 260
Irvine, CA 92602

/Scott T. Jarratt/
Scott T. Jarratt
Lead Counsel for Petitioner
Registration No. 70,297