

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

STODGE, INC. D/B/A POSTSCRIPT

Petitioner

v.

ATTENTIVE, INC.

(record) Patent Owner

Patent No. 11,553,074

DECLARATION OF DR. HENRY HOUH

TABLE OF CONTENTS

TABLE OF CONTENTS.....	2
I. ENGAGEMENT AND COMPENSATION	4
II. SUMMARY OF OPINIONS.....	6
III. QUALIFICATIONS.....	6
IV. MY UNDERSTANDING OF THE RELEVANT LAW	8
A. Claim Construction.....	8
B. Anticipation.....	9
C. Obviousness.....	9
V. TECHNICAL INTRODUCTION	13
Two-tap example: U.S. Pat. App. Pub. 2015/0178784 to Oliver (Ex. 1011).....	19
Two-tap example: U.S. Pat. No. 8,532,283 to Haggerty (Ex. 1007).....	20
Two-tap example: U.S. Pat. App. Pub. 2015/0142568 To Hsu (Ex. 1013).....	23
Two-tap example: U.S. Pat. App. Pub. 2009/0247140 to Gupta (Ex. 1017).....	24
Deeplinking technologies were well-established.....	26
Basic Ad-Network Technology.....	30
Basic Web Site Technology	32
VI. CLAIMS OF the '074 PATENT	42
VII. EXAMINATION HISTORY OF THE '074 PATENT.....	44
VIII. RELEVANT TIMEFRAME FOR DETERMINING OBVIOUSNESS	45
IX. LEVEL OF ORDINARY SKILL IN THE ART.....	46

X.	DETAILED EXPLANATION OF THE REASONS FOR UNPATENTABILITY	46
Ground 1.	Claims 1-30 were obvious over Oliver, Agrawal, and Khanna.	46
A.	Overview of the Ground.....	47
1.	Overview of Oliver (two-tap method)	48
2.	Overview of Agrawal (ad network)	52
3.	Overview of Khanna (mobile app deeplinking)	54
B.	Rationale (Motivation) Supporting Obviousness.....	56
C.	<i>Graham</i> Factors	60
D.	Reasonable Expectation of Success	61
E.	Analogous Art	62
F.	Claim Mapping.....	62
Ground 2.	Claims 1-2, 11-14, and 20 were obvious as in Ground 1, in further view of Molinet.	111
XI.	SECONDARY CONSIDERATIONS	116
XII.	OATH	118

1. I, Henry Houh, hereby declare as follows:

I. ENGAGEMENT AND COMPENSATION

2. I have been retained by Stodge, Inc. d/b/a Postscript (“Petitioner”) as an expert in the *inter partes* review proceeding described above. I have been asked to provide my technical review, analysis, insights, and opinions regarding the references that form the basis for the grounds of unpatentability set forth in the Petition for *inter partes* review of U.S. Patent No. 11,553,074 (“the ’074 Patent”). For this service, my billing rate is an hourly consulting fee of \$720/hour, and I am reimbursed for actual expenses. My compensation in no way depends on the outcome of this matter.

3. This declaration sets forth my analyses and opinions based on the materials I have considered thus far and the bases for my opinions. I understand that this declaration will be used in the above mentioned *inter partes* review. In forming my opinions, I have considered the following Exhibits:

Exhibit No.	Description
1001	U.S. Patent No. 11,553,074 (“ the ’074 patent ”).
1004	File History of U.S. App. Ser. No. 17/496590 (issued as the ’074 patent).
1005	U.S. Pat. No. 8,015,615 (“ Agrawal ”).
1006	U.S. Pat. App. Ser. No. 10/375,900 (“ Anderson ”).
1007	U.S. Pat. No. 8,532,283 (“ Haggerty ”).

1008	U.S. Pat. App. Ser. No. 13/461,541 (“ the Haggerty ’541 application ”).
1009	U.S. Pat. App. Pub. 2016/0142858 A1 (“ Molinet ”).
1010	U.S. Prov. App. Ser. No. 62/079,512 (“ The Molinet ’512 Provisional ”).
1011	U.S. Pat. App. Pub. 2015/0178784 A1 (“ Oliver ”).
1012	Cristal, G., “Ad Serving Technology, Understand the marketing revelation that commercialized the internet”, ISBN: 1484867572 (2014)(“ Ad Serving Technology ”).
1013	U.S. Pat. App. Pub. 2015/0142568 (“ Hsu ”).
1015	File History of U.S. App. Ser. No. 15/986,569 (abandoned).
1016	Firtman, M., “ <i>Programming the Mobile Web, Second Edition</i> ”, O’Reilly (2013).
1017	U.S. Pat. App. Pub. 2009/0247140 (“ Gupta ”).
1019	RFC 5724, “URI Scheme for Global System for Mobile Communications (GSM) Short Message Service (SMS)”, January 2010.
1020	WO 2005/062596A1 (“ Helkio ”).
1021	Flanagan, D. “ <i>JavaScript: The Definitive Guide, Sixth Edition</i> ”, O’Reilly Media, Inc. (2011).
1022	Complaint for Patent Infringement in <i>Attentive Mobile Inc. v. Stodge Inc. d/b/a Postscript</i> , Case No. 1-23-cv-00087 (D. Del. Filed Jan. 25, 2023).
1023	Scheiner, M., <i>Attentive Gets the Message Out</i> , available at https://crm.org/news/attentive-gets-the-message-out
1024	Levine, B., <i>Led by TapCommerce founders, startup Attentive launches SMS/MMS marketing platform</i> , available at https://martech.org/led-tapcommerce-founders-startup-attentive-launches-smsmms-marketing-platform/

1025	Ha, A, <i>TapCommerce's founders are back with Attentive</i> , a messaging startup that's raised \$13M, available at https://techcrunch.com/2018/02/08/attentive-launch/
1028	Mobile Marketing Association, "US Consumer Best Practices for Messaging", version 7.0, (2012), available at https://www.mmaglobal.com/documents/us-consumer-best-practices .
1032	U.S. Pat. App. Pub. 2014/0379482 (" Demsey ").
1033	RFC 3986, "Uniform Resource Identifier (URI): Generic Syntax", January 2005.
1034	U.S. Pat. App. Pub. 2013/0111328 A1 (" Khanna ").
1035	Excerpt from assignment database for Khanna.

II. SUMMARY OF OPINIONS

4. It is my opinion that:

- Claims 1-30 were obvious over Oliver in view of Agrawal and Khanna.
- Claims 1-30 were obvious as in Ground 1, in further view of Molinet.

III. QUALIFICATIONS

5. I hold a Ph.D. in electrical engineering and computer science from the Massachusetts Institute of Technology ("MIT"), awarded in 1998. I also hold a Master's Degree in electrical engineering and computer science from MIT, awarded in 1991, and two Bachelor's degrees (the first in electrical engineering

Patent No. 11,553,074

and computer science and the second in physics) also from MIT, awarded in 1989 and 1990, respectively.

6. I have approximately 25 years of work experience related to topics in networking, Internet communications, Web development, and Internet advertising. I am currently a technical consultant specializing in Social Networking, Web 2.0, Web Site Development, Data Networking, Optical Networking, Telecommunications, Media Streaming and Voice over IP.

7. I have extensive experience with Web programming and advertisement technologies in particular. From 1994-1996, I was the founder and CEO of Agora Technology Group, where I conceived and oversaw development of targeted advertising-supported web sites. From 1997-1999, I was a Software Engineer and Senior Scientist and Engineer for NBX (acquired by 3Com), where I worked on IP telephony (telephony using Internet protocol). From 1999-2004, I worked for Teradyne and later its spinout (Empirix). During this period, I held the position of Chief Technologist, Engineering Manager, of the Web Application Test Group. In that position, I was the Chief Architect for all Web testing products, re-architected the core testing product, and helped write a JavaScript interpreter.

8. In 2006, I was the Vice President of Operations and Technology for PodZinger Inc. There, I upgraded the capability of a consumer-facing search site and redeployed a web site from the company to a co-location facility. I was also

responsible for evaluating advertisement models of revenue. From 2007-2008, I was Vice President of Technology for the Delta Division of BBN Technologies, where I oversaw the development and deployment of a new fully-featured social networking web site. From 2008-2009, I was the Chief Technology Officer for Eons. With Eons, I integrated a product I created at BBN that Eons acquired. I integrated the product into the Eons social networking platform. I also helped evaluate advertisement platform offerings and brought the “Boom Network” advertising network to market.

9. My qualifications and publications are set forth more fully in my curriculum vitae, attached as Ex. 1003.

IV. MY UNDERSTANDING OF THE RELEVANT LAW

A. Claim Construction

10. I understand that a claim in an *inter partes* review is to be interpreted according to standards applied by district courts. First, the language of the claims themselves is of primary importance in the effort to determine precisely what it is that is patented. The terms used in a claim are generally given the ordinary and customary meaning that the terms would have to a person of ordinary skill in the art in question at the time of the alleged invention, unless the term is expressly defined in the patent. The person of ordinary skill in the art reads the claim in the context of the entire patent, including the specification. Next to the language of

the claims, the specification is the single best source for interpreting the claim terms. The claims may also be interpreted using the record of correspondence between the patent applicant and the Patent Office, and also with reference to other sources of evidence that can help to define the meaning of terms to a person of ordinary skill in the relevant time frame. I define the relevant time frame in ¶68, below.

B. Anticipation

11. I understand that a claim in an issued patent can be invalid if it is anticipated. In this case, “anticipation” means that there is a single prior art reference that discloses every element of the claim, arranged in the way required by the claim.

12. I understand that an anticipating prior art reference must disclose each of the claim elements expressly or inherently. I understand that “inherent” disclosure means that the claim element, although not expressly described by the prior art reference, must necessarily be present based on the disclosure. I understand that a mere probability that the element is present is not sufficient to qualify as “inherent disclosure”.

C. Obviousness

13. I understand that a claim in an issued patent can be invalid if it is obvious. Unlike anticipation, obviousness does not require that every element of

the claim be in a single prior art reference. Instead, it is possible for claim elements to be described in different prior art references, so long as there is motivation or sufficient reasoning to combine the references.

14. I understand that a claim is invalid for obviousness if the differences between the claimed subject matter and the prior art are such that the subject matter as a whole would have been obvious at the time the alleged invention was made to a person having ordinary skill in the art to which said subject matter pertains.

15. I understand, therefore, that when evaluating obviousness, one must consider obviousness of the claim “as a whole”. This consideration must be from the perspective of the person of ordinary skill in the relevant art, and that such perspective must be considered as of the “time the invention was made”.

16. The level of ordinary skill in the art is discussed in ¶69 below.

17. The relevant time frame for obviousness, the “time the invention was made”, is discussed in ¶68, below.

18. I understand that in considering the obviousness of a claim, one must consider four things. These include the scope and content of the prior art, the level of ordinary skill in the art at the relevant time, the differences between the prior art and the claim, and any “secondary considerations”. I understand that these four

things were specified by the U.S. Supreme Court in a case called *Graham v. John Deere*, and are therefore called the “*Graham* factors”.

19. I understand that “secondary considerations” include real-world evidence that can tend to make a conclusion of obviousness either more probable or less probable. For example, the commercial success of a product embodying a claim of the patent could provide evidence tending to show that the claimed invention is not obvious. In order to understand the strength of the evidence, one would want to know whether the commercial success is traceable to a certain aspect of the claim not disclosed in a single prior art reference (i.e., whether there is a causal “nexus” to the claim language). One would also want to know how the market reacted to disclosure of the invention, and whether commercial success might be traceable to things other than innovation, for example the market power of the seller, an advertising campaign, or the existence of a complex system having many features beyond the claims that might be desirable to a consumer. One would also want to know how the product compared to similar products not embodying the claim. I understand that commercial success evidence should be reasonably commensurate with the scope of the claim, but that it is not necessary for a commercial product to embody the full scope of the claim.

20. Other kinds of secondary considerations are possible. For example, evidence that the relevant field had a long-established, unsolved problem or need

that was later provided by the claimed invention could be indicative of non-obviousness. Evidence that others had tried, but failed to make an aspect of the claim might indicate that the art lacked the requisite skill to do so. Evidence of copying of the patent owner's products before the patent was published might also indicate that its approach to solving a particular problem was not obvious.

Evidence that the art recognized the value of products embodying a claim, for example, by praising the named inventors' work, might tend to show that the claim was non-obvious.

21. I further understand that prior art references can be combined where there is an express or implied rationale to do so. Such a rationale might include an expected advantage to be obtained, or might be implied under the circumstances. For example, a claim is likely obvious if design needs or market pressures existing in the prior art make it natural for one or more known components to be combined, where each component continues to function in the expected manner when combined (i.e., when there are no unpredictable results). A claim is also likely invalid where it is the combination of a known base system with a known technique that can be applied to the base system without an unpredictable result. In these cases, the combination must be within the capabilities of a person of ordinary skill in the art.

22. I understand that when considering obviousness, one must not refer to teachings in the specification of the patent itself. One can, however, refer to portions of the specification admitted to be prior art, including the “Background” section. Furthermore, a lack of discussion in the patent specification concerning how to implement a disclosed technique can support an inference that the ability to implement the technique was within the ordinary skill in the prior art.

V. TECHNICAL INTRODUCTION

23. I have reviewed the '074 patent. It is largely directed to a known business method, namely that of automating the process of enrolling customers in subscriptions using prefilled and preaddressed messages. To do this, the '074 patent claims the use of known web advertising technology in conjunction with its business method.

24. By the time the first application leading to the '074 patent was filed on May 26, 2017, smartphones had been in widespread use for nearly a decade. With the rise of smartphone use, text-based messaging (*e.g.*, “SMS” messaging) became a primary form of communication. This form of communication was naturally adopted by advertisers, who wanted to send advertising texts to consumers. Advertising texts, however, could be subject to legal restrictions, contractual restrictions or standards that require or recommend that an advertiser first obtain a user’s consent prior to sending advertising texts. (Ex 1001, 6:49-57).

25. For example, I refer to the Mobile Marketing Association’s 2012 publication entitled “U.S. Consumer Best Practices for Messaging”. (Ex. 1028). This document states that it “is a compilation of accepted industry practices, common wireless carrier policies, and regulatory guidance that have been agreed upon by representative member companies from all parts of the off-deck ecosystem.” (Ex. 1028, p. 004). The document provides guidance (in 2012) concerning the need to obtain consent and proof of possession of a mobile device prior to sending marketing text messages. (Ex. 1028, pp. 008-009). The portion of table 1.4 shown at the bottom of page 008 is reproduced here:

1.4 Opt-In

	Guideline
1.4-1	Content providers must obtain opt-in approval from subscribers before sending them any SMS or MMS messages or other content from a short code.
1.4-2	Program flow and information must not be misleading in any way.
1.4-3	Recurring standard rate programs require a single opt-in.
1.4-4	When opt-in to a recurring program occurs via the web or other non-mobile point of origination, the content provider must obtain verification that the subscriber is in possession of the handset being opted-in to the service.
1.4-5	For recurring standard rate programs, subscribers should indicate their willingness to participate in a program and receive messages from the program as follows:
1.4-6	1. Subscriber initiates opt-in to a recurring Standard Rate Program by responding to a call to action (CTA): i.) Subscriber may send a Mobile Originated (MO) message from their handset to the short code. ii.) Subscriber may initiate opt-in from a web interface

(Ex. 1028, p. 008, Table 1.4). This table provides opt-in guidelines for short code SMS marketing messages, which are messages to a short code number, rather than a standard ten-digit telephone number. The guidelines state (1.4-4) that when opt-in occurs “via the web or other non-mobile point of origination”, then the content provider must verify that the subscriber is in possession of the handset (smartphone)

being opt-into. In other words, if someone fills in a web form that signs a phone number up for marketing messages, the marketer must separately verify that the owner of the phone actually wants to receive the messages. This was usually done by sending a confirming text, and asking the user to respond with “y” or “n” to opt-in. However, this confirming text is not necessary where there is a mobile point of origin—i.e., where a user does not sign up with a web form, but rather by sending a text message to the marketer in the first instance. In that case, the marketer receives the opt-in communication directly from the user’s phone, which also serves as proof that the user possesses the device for which marketing communications are requested. For example, such a text message might be sent in response to a “call to action” or “CTA” (see 1.4-6). A CTA could be an advertisement, as shown on page 016 of the document:

Call to Action:

Program sponsor →
Service Description →
Frequency of Messaging →
Customer Support Info →
Opt Out Info →
Additional Carrier Costs →
Terms & Conditions →

The following is advertised:

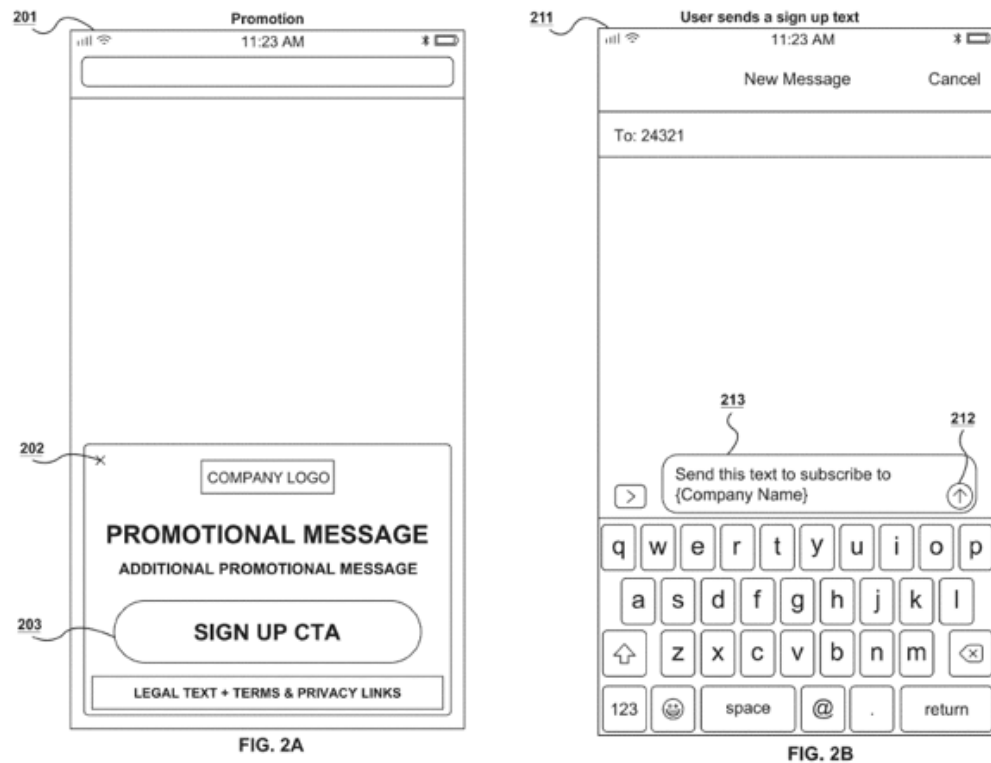
Upmobile Ski Alerts!
Send us the resort name, we'll send you the snow conditions. Txt
'Mammoth' to 12345 to receive ongoing alerts for Mammoth resort.
Get 10 msgs/month.
Text HELP for help.
To stop text STOP.
Msg&Data Rates May Apply.
T&Cs avail at www.mammoth.com/mobile.

26. Thus, in 2012, five years before the filing of the first application leading to the '074 patent, it was already industry-standard to have advertisements that asked a user to send a text message to a marketing company, and to have the

text message serve as both an opt-in communication and proof that the user possessed the mobile device with the phone number for which marketing messages were solicited. This is already a short step to the teachings of the '074 patent.

27. The '074 patent uses a similar “call to action” in the form of web-based advertisements. But instead of encouraging a user to text an opt-in message to a certain telephone number, the advertisement encourages a user to click on a link. (Ex. 1001, 6:27-34). Clicking the link from a mobile device causes the device’s text-messaging app to open. (Ex. 1001, 6:27-45). Upon opening, the text-messaging app presents the user with a pre-filled text message. (Ex. 1001, 6:41-45). The pre-filled text message provides both a destination telephone number and a message body, such that the user need only click “send” to send the text. (Ex. 1001, 6:45-57). When sent (to a designated server), the text message signs a user up for a promotion, such as a further promotional message. (*Id.*).

28. The process is shown in Figs. 2A and 2B of the '074 patent, reproduced here:



(Ex. 1001, Figs. 2A-2B). Figure 2A on the left shows a mobile device screen 201, with an ad 202. (Ex. 1001, 6:58-7:41). The ad 202 has a clickable link 203. (*Id.*). When the consumer clicks the link, the device opens the text-messaging app (Fig. 2B), which has a pre-filled destination number 214 and a pre-filled body 213. (*Id.*). The user need only click the send button (212) to send the text and sign up to receive marketing texts. (*Id.*). In this way, the user only needs to take *two actions* to sign up: clicking the link 203 in the ad, and then clicking the send button 212 in the text-messaging app. (Ex. 1001, 6:58-7:41). Because only two user actions are required, the '074 patent refers to this approach as a “two-tap” approach. (Ex. 1001, 2:33-37,

6:53-57).

29. I understand that the Patent Owner considered this “two-tap” method to be the point of novelty for the ’074 patent. (Ex. 1014, p. 2). The ’074 patent states that

“Known methods allow a user to open a vendor’s application or a vendor’s website on a mobile device to select a product or service. The user can then provide payment information via the vendor’s application or website to complete a transaction with the vendor. With these known methods, however, the user often pauses its previous activities (e.g., viewing a website, reading an email) on the mobile device and is redirected to the vendor’s application or website to purchase the products and services via many user input (e.g., clicks, or screen taps). This time-consuming and burdensome process results in many users leaving the purchase before the transaction is completed.”

(Ex. 1001, 1:20-43). The “two-tap” approach allegedly avoided potential mistakes in typing a text message and provided a record of consent with a single message. (Ex. 1014, p. 2).

30. Based on my review of the prior art, however, it is clear that the Patent Owner was not the first to employ a “two-tap” method. Even in the patent literature, there are several examples of two-tap methods, as I explain here in the next few paragraphs:

Two-tap example: U.S. Pat. App. Pub. 2015/0178784 to Oliver (Ex. 1011)

31. Oliver, which is the subject of Ground 1, below, also describes a two-tap method. Oliver teaches a streamlined method of generating customer leads. (Ex. 1011, Title, ¶0009). In Oliver, a potential customer interacts with an advertisement for a business. (Ex. 1011, ¶0034). The advertisement contains something that the user can interact *with* (using a mobile device), which Oliver calls a “passive entity”. (Ex. 1011, Abstract, ¶0007). The “passive entity” is sometimes a scannable QR code, but Oliver makes clear that a passive entity can be anything that can store URI information and be activated by a user’s mobile device. (Ex. 1011, ¶¶0027-0029, 0007, Abstract). The link information that Oliver encodes in a “passive entity” follows standard URI schemes (*e.g.*, “smsto:”), as shown in ¶0028 of Oliver, reproduced here:

[0028] smsto: 543-21: This is where the message goes
and ID=### info etc. To receive your reward TAP
SEND!

(Ex. 1011, ¶0028).

32. When the user activates a passive entity containing the smsto: URI (a *first* user interaction), the user's mobile device creates a pre-populated text message, as shown in Fig. 2 of Oliver, reproduced again at right. (Ex. 1011, ¶¶0027-0029). If the user then taps the “send” button on such a message (a *second* user interaction), the text will be sent, signing the user up for a promotion. (Ex. 1011, ¶0007).

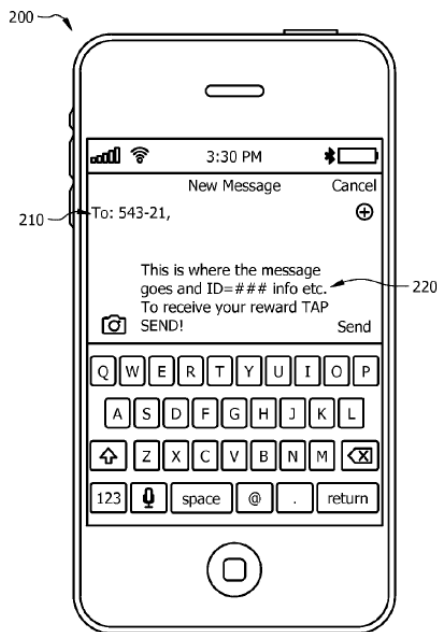


FIG. 2

33. Oliver explains that limiting the user actions to two improves the user response process:

“the disclosed principles **streamline the response process** by simply requiring **only two actions** by the user: the scan or other means of activating the passive entity, **and the action of sending** the prepopulated message generated by the activation of the passive entity.”

(Ex. 1011, ¶0009)(Emphasis added).

Two-tap example: U.S. Pat. No. 8,532,283 to Haggerty (Ex. 1007)

34. Haggerty teaches a two-tap method for SMS marketing. Haggerty teaches a “two-tap” business method similar to that of the ’074 patent. In Haggerty, a user can interact with an advertisement. (Ex. 1007, 14:41-49). The interaction can be scanning a QR code, but can also be clicking a hyperlink. (Ex.

1007, 14:41-49, *see also* 1:36-62, 2:1-5, 8:51-56, 10:59-11:21, 16:26-31, Fig. 3C, 4G). Once clicked, the link can cause the user's mobile device to pre-populate an SMS message. Haggerty states:

“[U]pon reading the QR code using a smart phone, a **pre-populated text message** addressed to the destination address **with the contents indicated therein** is presented to the user, along with a prompt for **confirmation for sending the message.**”

(Ex. 1007, 10:1-5)(Emphasis added). Note that Haggerty teaches that a QR code is not necessary, and that “[i]n general, any of the actions that may be initiated by a QR code **could also be associated with a link.**” (Ex. 1007, 11:19-21)(Emphasis added). A PHOSITA would understand a “link” to mean a clickable link in a Web page, implemented, for example, using an HTML anchor tag (<a>) or JavaScript to open a URI similar to the URIs described in RFC 5724 (Ex. 1019).

35. The pre-populated text message is sent, after confirmation by the user, to a “call center” for processing, as shown in Fig. 4B. (Ex. 1007, Fig. 4B, 12:39-47). The “call center” is not limited to handling or dealing with *voice* calls, however, and Haggerty also considers an SMS text message to be an “SMS call”. (Ex. 1007, Figs. 4B-4C, 4:38-56). Upon receiving the SMS text message, the call center processes the message according to the particular advertising campaign, and sends a response SMS. (Ex. 1007, 12:39-67). An example message flow is shown

in Fig. 4B, reproduced below, with yellow highlighting added to show the text messages:

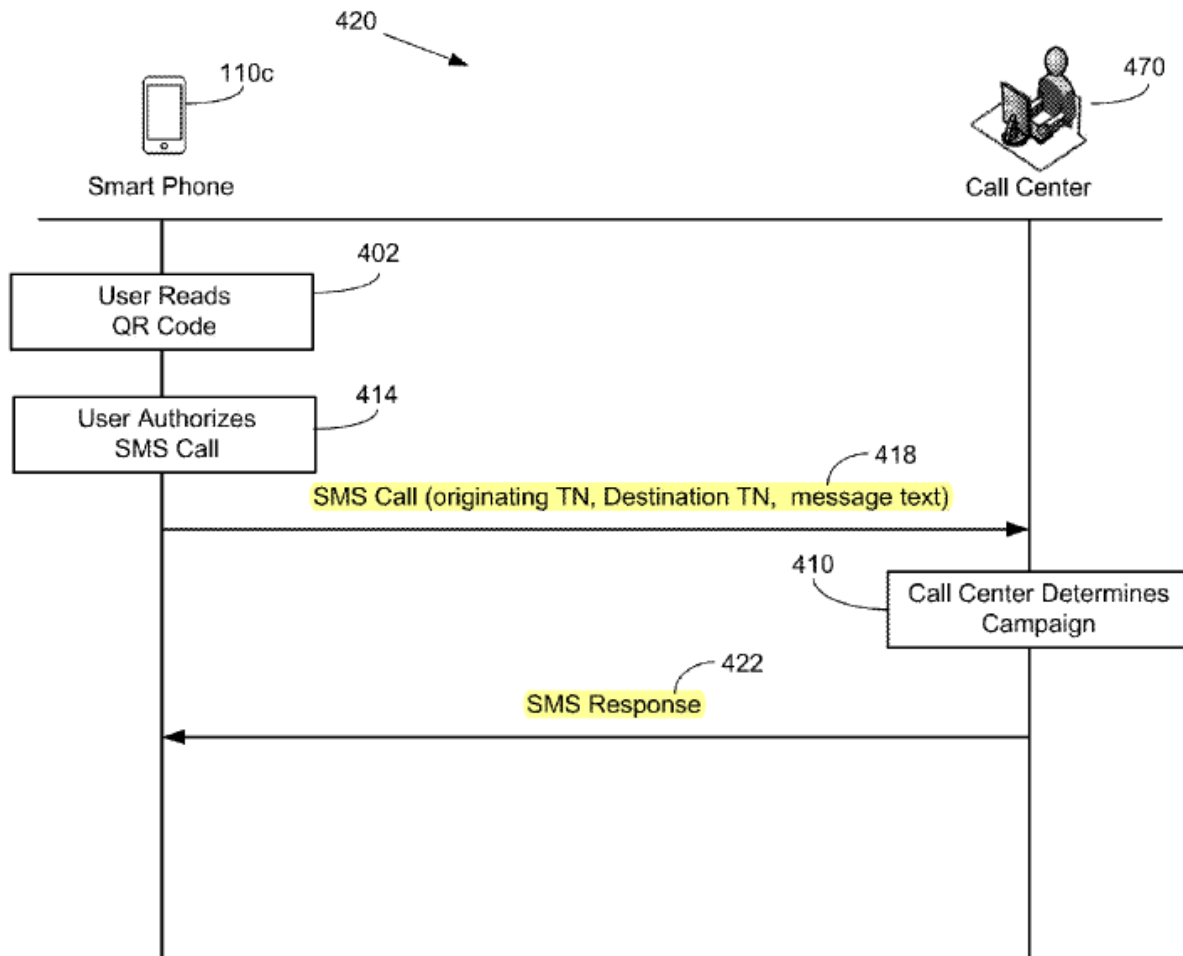


FIG. 4B

(Ex. 1007, Fig. 4B, 12:39-67)(highlighting added).

36. Just like the '074 patent, Haggerty teaches a user response to an advertisement with only two actions: activating a link, and then sending a pre-populated SMS. (Ex. 1007, 10:1-5, 14:41-49, 1:36-62, 2:1-5, 8:51-56, 10:59-11:21, 16:26-31, Figs. 3C, 4G). Haggerty further states that its method provides a

convenient way for a user to respond to an advertisement (Ex. 1007, 1:53-62) with lowered risk of mistyping information in the SMS (Ex. 1007, 9:48-55).

Two-tap example: U.S. Pat. App. Pub. 2015/0142568 To Hsu (Ex. 1013)

37. Hsu also teaches a two-tap method. In Hsu, a user downloads an ad to a browser. (Ex. 1013, ¶0031). When the user clicks on the ad, an advertising server determines whether there is a “landing page” for the ad, and if there is not, receives user agent data from the mobile device. (Ex. 1013, ¶0032-0038). Using the user agent data, the server sends a “message launching command” to the mobile device, which will launch a message app with a pre-populated message and telephone number. (Ex. 1013, ¶¶0038-0040). Hsu states:

“In this embodiment, the selected message launching command may be implemented using program syntax such as: smsto:?phonenumber=”0987654321”&body=”I am interested in xxx product. Please provide detailed information”. In the program syntax, “phonenumber” is used to place the preset telephone port number, and “body” is used to place the preset text, but the present disclosure is not limited in this respect. Program syntax of the message launching command may vary with the different operating systems installed at the mobile devices.”

(Ex. 1013, ¶0040). Hsu further states:

“Accordingly, upon a user trigger (e.g., clicking/ touching) of an advertisement displayed by the mobile device 1, messaging function that corresponds to the operating system (e.g., iOS, android,

Windows Phone, etc.) or the brand of the mobile phone (particularly, a smart phone) may thus be launched, and a text that corresponds to the advertising activity may be loaded from a variety of preset texts. The user only needs to trigger a "send" button of the messaging interface for expressing interest in the specific product shown in the landing page, or in the advertisement displayed by the mobile device 1. After completion of message transmission, the phone number of the user is automatically acquired by the advertising server 2, and the relevant information may be provided to the host advertiser 4 for subsequent sale activities or services.”

(Ex. 1013, ¶0042).

Two-tap example: U.S. Pat. App. Pub. 2009/0247140 to Gupta (Ex. 1017)

38. Gupta also teaches a “two-tap” method. In Gupta, advertisements can be paired with “mobile advertisement actions”. (Ex. 1017, ¶0034). As Gupta states,

“[a] mobile advertisement action may be a clickable interactive object embedded in an advertisement, such as a button or a **link**. The mobile advertisement action may represent **one or more operations that may be performed by a mobile device**, such as email, **text** messaging, phone calling, or generally any operation that may be performed by a mobile device and may provide may provide [sic] enhanced advertising capabilities to a revenue generator A 110A. The mobile advertisement action may include data specific to the revenue generator A 110A, such as the phone number, address, or generally any information specific to the revenue generator A 110A.

The mobile advertisement action may include a description such as ‘Click to call,’ ‘**Click to SMS,**’ ‘Click to coupon,’ ‘Click to download,’ ‘Click to survey,’ ‘Click to make reservation,’ ‘Click to attend event,’ ‘Click to buy,’ or generally any description that may describe the functionality provided by them mobile advertisement action.”

(Ex. 1017, ¶0035)(Emphasis added).

39. When a user clicks on a mobile advertisement action, “the service provider 130 may cause the mobile device of the user AA 120AA to perform the specified action, such as by providing instructions to the mobile device and causing the instructions to be executed.” (Ex. 1017, ¶0036). For example, the device may open a pre-populated text message:

“If the mobile device supports more than one messaging protocol, the service provider 130 may attempt to determine a default or preferred protocol. At block 425 the service provider 130 may determine the messaging address of the revenue generator A 110A for the determined messaging protocol, such as an email address, a **text messaging** number, or generally any messaging address. At block 426 the service provider 130 may **cause the mobile device of the user AA 120AA to create a message addressed to the revenue generator A 110A**, such as by generating code capable of causing the mobile device to generate the message, communicating the code to the mobile device and causing the code to be executed on the mobile device.”

(Ex. 1017, ¶0069)(Emphasis added).

40. One example of such an action can be a “click to message” action.

“Another mobile action may include a ‘click to message’ action 710. Clicking a ‘**click to message**’ action 710 allows user AA 120AA to **send a message directly to the contact phone number** or email of revenue generator A 110A. The message may be any of the available messaging systems such as short message service (SMS) or Multimedia Messaging Service (MMS). Additionally, the message may be sent from the email account of user AA 120AA. The message may be any message such as a question or response to the advertisement from the user AA 120AA to the revenue generator A 110A. The message may also be a message with an integrated picture, sound, or file that the user AA 120AA wishes to provide to the revenue generator A 110A. Revenue generator A 110A may need to provide a phone number, email address, or other contact information to receive the message from user AA 120AA.”

(Ex. 1017, ¶0088)(Emphasis added). The messages can be to enroll in marketing programs. (Ex. 1017, ¶¶0091, 0096-0097).

Deeplinking technologies were well-established

41. To pre-populate text messages, such references often used available technology to link to mobile apps. (e.g., Ex. 1011, ¶¶0027-0029)(Ex. 1013, ¶0040)(Ex. 1010, pp. 021-022, 015). Mobile app links were clickable links on a webpage that could be directed at the opening screen of a mobile app, but could

also be directed *within* the app (*e.g.*, to information about a specific product or service). In that case, such links are called “deeplinks”, because they are directed to a point “deep” within the app, not just the opening screen. (Ex. 1034, ¶0030).

42. Deeplinking technologies for mobile apps were well-established years before 2017. For example, it was well-known to use Uniform Resource Identifiers (“URIs”) as deeplinks to open a text-messaging app and pre-populate a text-message. As the textbook “*Programming the Mobile Web*” stated in 2013, “[w]e all like the Short Message Service; that’s why **mobile browsers generally offer the ability to invoke the new SMS window from a link**. To do this, we have two possible URI schemes, sms:// and smsto://”. (Ex. 1016, p. 557)(Emphasis added). These URI schemes were used at the start of a link—so just as a standard web URL might begin with http://, a link to a text-messaging app would begin with sms://. These URI schemes were used to pass both the destination phone number and the message body to the text-messaging app (thereby “deeplinking” to the app). (Ex. 1016, pp. 557-558)(Ex. 1013, ¶0040)(Ex. 1011, ¶¶0027-0029). The major smartphone operating systems (iOS and Android) in the relevant timeframe had such SMS URI implementations. (Ex. 1010, p. 015).

43. An appropriate URI protocol (*e.g.*, “sms:”) to open text-messaging applications with pre-populated text and destination numbers was standardized by

the year 2010 in IETF RFC 5724. (Ex. 1019, pp. 007-008, §2.2, p. 010, §2.5).

RFC 5724 provides examples of the use of such URIs in §2.5, reproduced here:

2.5. Examples of Use

```
sms:+15105550101
```

This indicates an SMS-message-capable recipient at the given telephone number. The message is sent using the user agent's default SMS delivery method.

```
sms:+15105550101,+15105550102
```

This indicates SMS-message-capable recipients at the given telephone numbers. The identical message should be sent to both recipients using the user agent's default SMS delivery method.

```
sms:+15105550101?body=hello%20there
```

In this case, a message (initially being set to "hello there", which may be modified by the user before sending) will be sent via SMS using the user agent's default SMS delivery method.

(Ex. 1019, pp. 007-008). This protocol is simple and could have been easily used to produce a wide variety of pre-populated text messages. Employing such a URI causes a pre-populated text message to open in the “default SMS delivery method”, i.e. the default SMS app on a smartphone. Note that after the user activates the URI (e.g. by clicking on a link as described above), the message is not sent without the user confirming the send. (Ex. 1019, p. 005)(“In either case, simply activating a link with an ‘sms’ URI SHOULD NOT cause a message to be sent without prior user confirmation.”).

44. The URIs specified in RFC 5724 could be used in clickable website links, such that when a user clicked on the link, a pre-populated SMS message would appear in the user's text-messaging app. (Ex. 1020, pp. 009:33-010:33). Numerous prior art references discussed such URIs. (Ex. 1016, pp. 557-558)(Ex. 1013, ¶¶0040)(Ex. 1011, ¶¶0027-0029)(Ex. 1020, pp. 009:33-010:33)(Ex. 1010, pp. 015, 021).

45. *Programming the Mobile Web* was published by O'Reilly, a well-known publisher of books related to software and networks. (Ex. 1016, p. 001-003). It bears a copyright date of 2013 and a first release date of 2013-03-15. (Ex. 1016, p. 004). Experts in this field would reasonably rely on it, including for its copyright and release dates as publication dates.

46. RFC 5724 is a publication of the Internet Engineering Task Force (IETF), which is an international community of interested parties traditionally responsible for the evolution and maintenance of Internet architecture and function. RFCs are documents that propose or adopt technical standards. RFC 5724 has the status of a "Internet Standards Track document". Experts in this field would reasonably rely on RFCs, including for their publication dates, which in the case of RFC 5724 was January 2010. (Ex. 1019, p. 001). RFC 5724 has the IETF maturity level of "Proposed Standard" (<https://datatracker.ietf.org/doc/rfc5724/>), which means it "is generally stable, has resolved known design choices, is believed

to be well-understood, has received significant community review, and appears to enjoy enough community interest to be considered valuable.”

(<https://datatracker.ietf.org/doc/html/rfc2026#page-10>).

Basic Ad-Network Technology

47. The claims of the '074 patent also pair the known two-tap business method to standard Internet advertising technology. The standard advertising technology used by the '074 patent can be explained with a short background. In the early days of the Web, website publishers could contract directly with advertisers or ad agencies, who would provide ads (*e.g.*, “banner ads”) to place on a website. (Ex. 1012, p. 090). As the textbook *Ad Serving Technology* states:

“In traditional buying, the Agency buyer makes a call to a contact they have acquired at the Publisher site where they are interested in buying media space. On both sides of the phone, the Publisher and Agency agree on a particular ‘buying-rate’ or ‘cost model’ for the Advertiser’s campaign. The Agency buyer may negotiate a position so that the Publisher is responsible for performance of the campaign and in doing so the final price of the ad space may be tied to this figure (more on actual cost models and costs in Chapter 2 - Publishers). The relationship between the media buyer and the Publisher seller in most instances will have been formed over many months in which time the Publisher would meet with the buyer to provide updates about the Publisher’s media space and technology integrations.”

(Ex. 1012, p. 090).

48. *Ad Serving Technology* (Ex. 1012) bears a copyright date of 2014, and is a textbook concerning the Internet-based technology used to serve ads. Experts in this field would reasonably rely on it, including for its copyright date as a publication date.

49. When placed on webpages, such ads usually had links, such that a user who clicked on the ad would be directed to the advertiser's home page. (Ex. 1012, pp. 055-056). This system had a variety of limitations. First, most website publishers did not have the expertise to seek out advertisers. Only the largest publishers regularly had ads, but even these ads could not be quickly changed, may not have been placed optimally, and could not be targeted to specific users. Furthermore, it was difficult to perform accounting tasks like recording the number of views of an ad.

50. To address these concerns, *ad networks* arose. Ad networks would contract with website publishers, through a simple, one-time sign-up process, for the right to place ads on the publisher's site. To place ads on a publisher site, ad networks would transmit so-called "ad tags" to publishers. (Ex. 1012, p. 051-054). The ad tags were placeholder segments of code (HTML and/or JavaScript) that could be placed into publisher webpages.

Basic Web Site Technology

51. To explain ad tags, a short background of Web site technology is appropriate. A Web page is basically a file that can be downloaded from a server to a user device, using HyperText Transfer Protocol (“HTTP”). Communication by the HTTP protocol allows for a request for a file using a URL (*e.g.*, <http://www.uspto.gov>). Such a request, such as an HTTP GET request, will transmit a number of fields, including the Request field, including the method (here: GET), a path (a directory path), the version of HTTP, the Host field (such as www.uspto.gov), the User-Agent field, which includes information about the requesting device such as the operating system and browser, and the Accept field, which can specify the desired form of the response. An HTTP request may also include cookie data, if the server at the host domain has previously set a cookie at the user’s browser. A cookie is a small amount of data that can be set by a server to be stored on the user device. Cookies can contain name-value pairs, and once a cookie is set for a given Web site domain (and not expired or erased), the cookie data will be sent back to the domain that set the cookie in subsequent HTTP communications to the domain.

52. When a Web server receives a proper request, it will send back a Web page. A typical Web page is a text file written in HyperText Markup Language (HTML). An HTML file can have normal text that is readable by the user on their

browser that can also be “marked up” with HTML tags. Tags are simple special text set apart using less-than (<) and greater-than (>) signs. As a simple example, suppose an HTML file has the normal text sentence “A family lives in that house”. If the Web site designer wanted to have the text “that” appear in bold, this could be accomplished by adding tags, as follows: “A family lives in that house.” When a browser interprets this text, it will read the characters “”, and understand that “” is not text to be displayed, but rather means that the following text should be written in bold, until an end tag (“”) is reached.

53. Many tags beyond are possible, allowing complex Web pages to be constructed. For example, an tag requests that the browser load and place an image on the page. The tag can include parameters that specify further how the tag works. For example, a “src” parameter can be included within the < > marks to provide the browser with a URL that indicates the location of the image, anywhere on the Web, to load. For example, the tag:

```

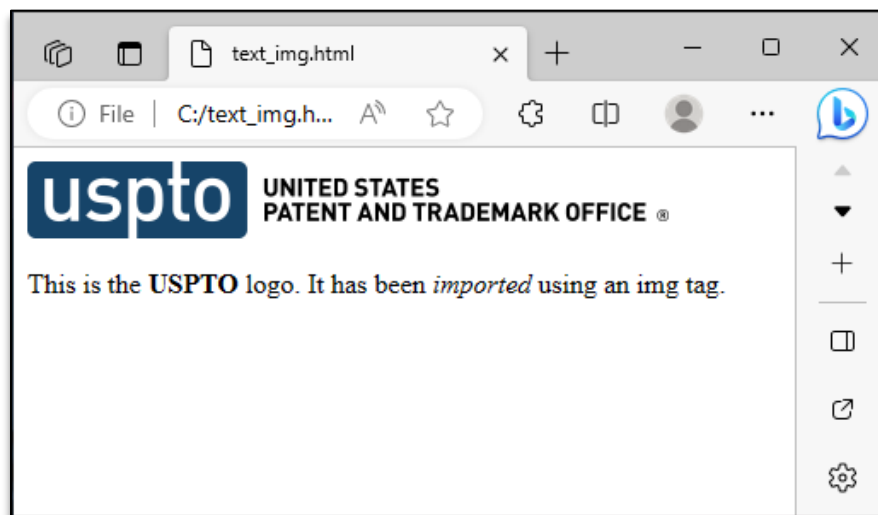
```

will cause a browser to download the USPTO logo from a USPTO server, and place it within a page. To take this img tag and make it into HTML document, one can simply copy the tag into a new file in a simple text editor, like Notepad on a Windows computer, save the file to the desktop with a filename like “test.html”, and

then double-click on the file or open it directly with a Web browser's File->Open command. For example, if one pastes the following into a file and opens it with a browser:

```
  
  
<br><br>This is the <b>USPTO</b> logo. It has been  
<i>imported</i> using an img tag.
```

the browser will display something like the following:



54. In the code above, the img tag causes the browser to issue a request to a USPTO Web server to retrieve the USPTO logo. The
 tag inserts a line break. The and tags indicate the start and end of bold text. The <i> and </i> tags the start and end of italic text. To view all the tags in an HTML

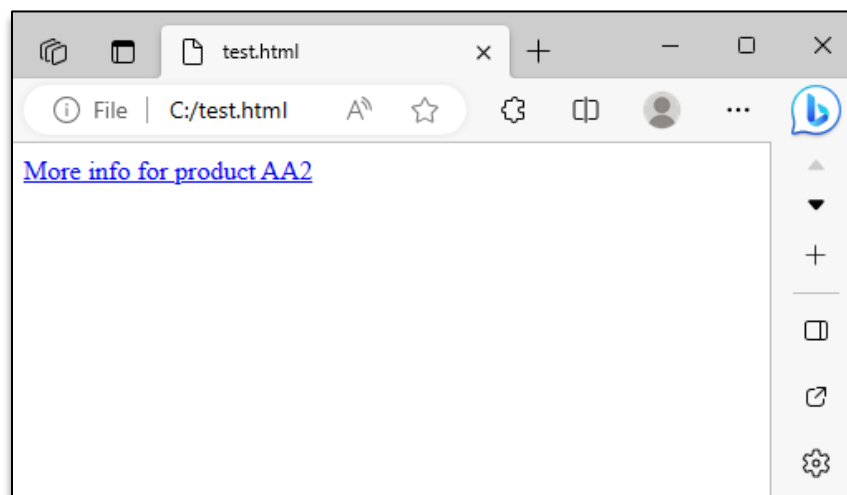
document, one can generally load the page in a browser, and use a “view source” function (for example, by right clicking on the browser window).

55. In order to make the HTML document into a Web page, the HTML file is placed in the document directory of a web server. Then, the HTML document can be accessed by web browsers over the Internet as a Web page.

56. To add a mobile app deeplink, one could use an anchor tag “<a>” together with a url parameter that specified a known deeplink format. For example, one can paste the following link into the text file of a Web page:

```
<a  
href="sms://+3490322111/&body=Interested%20in%20Product%20AA2">  
More info for product AA2</a>
```

Opening this file with a Web browser will result in a Web page that looks like the following, which shows only the text “More info for product AA2”, underlined to show that it is a clickable link:



57. If one clicks on this link with a desktop computer web browser that does not have click-to-text capability, nothing will happen. However, clicking the link in a mobile browser on an Apple smartphone running iOS (one can email oneself the file and open the email attachment) will cause a pre-populated text message, directed to the telephone number (349) 032-2111 and with the message body “Interested in Product AA2” to open. Performing similar actions (with potentially slightly different formatting) on Android smartphones will produce a similar result.

58. One type of special tag that can be used in a Web page is `<script>`. The `<script>` tag specifies the beginning of program code, usually in a language called “JavaScript”. The end of the program code will be designated with a `</script>` tag. The program code, called a “script,” can either be written directly into the file containing the HTML, or can be loaded using a “src” parameter, just as with the `img` tag.

59. JavaScript is a high-functionality language, available since the 1990s, that can use most familiar programming constructs. (Ex. 1012, pp. 051-054)(Ex. 1021, pp. 019-036). As *Ad Serving Technology* explained:

“JavaScript [¶]JavaScript is called a client-side language. It runs or ‘executes’ in the user’s browser and allows for a whole host of important web page tricks such as popup windows and browser manipulation (like a command to print something on the page).

JavaScript can gather and send a user's information and [sic] to a remote server so that the server is capable of responding by sending out a new piece of information (like making a previously hidden part of the website, suddenly appear). JavaScript is the most common method of ad delivery that third party ad servers use. The flexibility of the JavaScript code allows the ad to be pulled directly into a Publisher's web page from the third party ad server and make decisions about what ad to display in real time. JavaScript can do interesting things like work out what time it is, or find out what browser the user is using: information like this can be passed back into the ad server when the ad loads allowing for some impressive customisation inside the ad itself."

(Ex. 1012, p. 053-054).

60. JavaScript can also access and alter the rest of the Web page file using the Document Object Model ("DOM"). The DOM is a model of the Web page that is created when a page is loaded. (Ex. 1021, pp. 379-429). At the time of loading, the browser will look for tags, load any additional resources that are needed (like images or scripts that have src parameters), construct a DOM, and execute scripts in the page. (Ex. 1012, p. 054)(Ex. 1021, pp. 332, 021, 060).

61. The use of scripts allowed an ad network to have some control over the placement of ads *at the time a page was being viewed*. An ad tag (script) could be placed into a Web page and would execute when the user viewed the page. By using a script, the ad tag could cause the browser to request an ad from the ad-

network server at the time the user viewed the page. The ad-network server would choose an ad in realtime and send the ad back for insertion into the webpage. For example, an ad tag is shown (between the <script> and </script> tags) in the prior-'900 application (Ex. 1034):

```
<iframe id='google_ads_frame' src='about:blank' hight=80 ... >
...
content of the page
...
<script>
extract important key words like title, meta tags, headers etc, till we
have reached 1K byte limit.
frames['google_ads_frame'].src=http://pagead.google.com/ads?url=<
url>&contents=word1+word2+....
</script>
```

(Ex. 1006, 026).

62. The ad tag (the script) uses an <iframe> to display an ad. An iframe is a tag that sets up a separate content area on a web page, which can, for example, have its own scrollbars and load its own Web page within another Web page. In the case above, the iframe has an src parameter that is set to about:blank, which will initially cause a blank page to load within the area of the frame on the page. The iframe also has an “id” parameter, which gives the frame a name (here: “google_ads_frame”) that can be accessed using JavaScript. Later in the web page file, there is a <script> tag indicating JavaScript. Part of the JavaScript is simply described (not shown), and that part of the script will gather information about the Web page. The next part of the JavaScript sets the src parameter of the iframe

named “google_ads_frame” to a location at a Google ad server. This causes the iframe to reload, and to send a request (called an ad request) to the Google ad server. (e.g., Ex. 1012, p. 061-062). The request for an ad will include information collected from the Web site, as indicated by the parameters in the src URL, namely the key value pairs appearing after the question mark in the URL (“url=<url>&contents=word1+word2+...””) as was typical for parameter passing in a URL in the relevant timeframe. (Ex. 1021, pp. 361, 034, 584)(Ex. 1012, p. 568, 583, 588). In principle, JavaScript can pass any information accessible to it to an ad server in such a way. When iframe is reloaded and the ad request is sent, the Google ad server will select and send back a targeted ad. In the specific example above, the <script> is separate from the iframe, but scripts can also be found within an iframe.

63. Ad tags in general are shown in Fig. 1.5 of *Ad Serving Technology*, reproduced here with its caption:

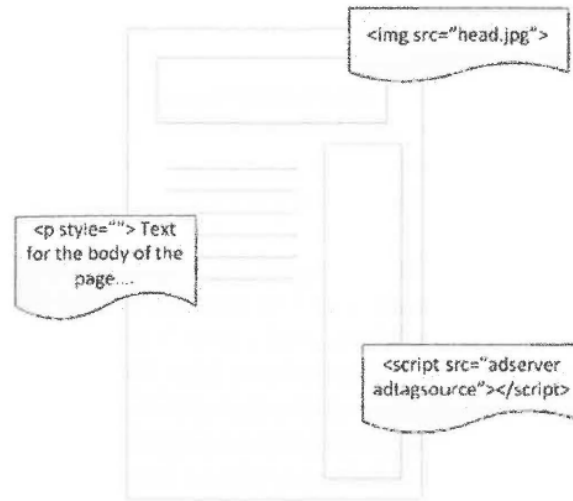


Figure 1.5: Behind the scenes a standard web page is made up of a series of HTML tags. An Ad tag is no different, which can sit on any web page to display an ad to the user. In this instance (bottom right) the HTML tag contains a reference to JavaScript.

Here, the page is shown in light grey in the background, and contains tags, such as `<img>` and `<p>`. One of the tags is a `<script>` tag, which has a `src` parameter that points (conceptually) to “adserver adtagsource”, which is a stand-in for a URL that specifies a location of a JavaScript file at an ad server. When the browser loads the Web page, the browser will see that the `<script>` tag refers to a script on another server, issue a request to retrieve the script, and execute the script after it is received.

64. Ad tags generally originated from the ad network servers, not the publishers of a particular page. As such, ad networks would send ad tags to publishers. The publishers would then incorporate the ad tags into their Web pages, which would then be served to users by publisher web servers. That is, when a user downloads a Web page from a publisher, the Web page will contain an ad tag, that will execute when the user accesses the page. When the ad tag

executes, it will typically contact an ad-network server, and ask it to provide an ad to incorporate into the Web page. This is shown, for example, in Fig. 5 of Demsey, reproduced here with added red-dashed outlines.

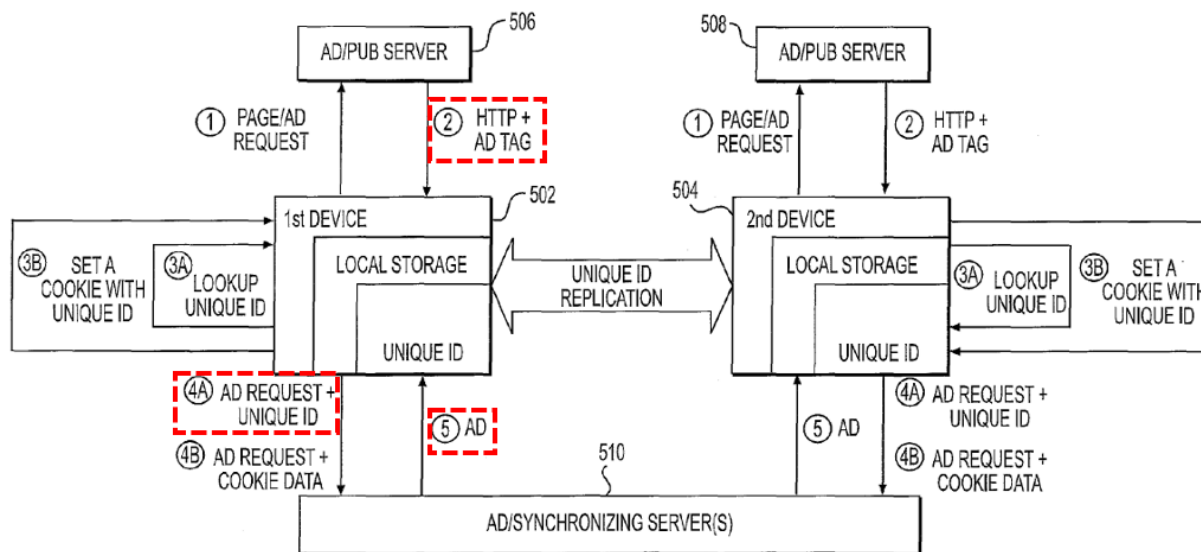


FIG. 5

(Ex. 1032, Fig. 5). As shown in Fig. 5, a Publisher server (506) sends (2) a Web page via HTTP in response to a page request (1). The page is received by the user device 502, which loads the page as described above. While loading the page, the ad tag, which contains JavaScript, executes and causes the browser to collect some information about the user (here, a unique ID and cookie information) and send an ad request containing this information to a server 510. The server 510 selects an appropriate ad and sends it back to the user device, where the browser on the user's device incorporates the ad to the correct position and size in the page. (Ex. 1032, ¶¶0045-0048).

65. The ad-network process using ad tags was advantageous for several reasons. With older technology (no ad tags), an advertiser would only know *which website* would contain the ad, not *which users* would eventually see the ad. The advertiser would have to guess the characteristics of a target audience for the ad from the content of the webpage. Having ad tags that execute *when the user views a page*, however, allowed the ad tag to collect information about the *specific user* and the *user's device*, and send this information to the ad-network server. This, in turn, allowed the ad-network server to target an ad directly to the *specific user*, formatted correctly for the specific user's device (which could be a desktop or smartphone, for example). Furthermore, because the publisher did not need to contract with specific advertisers, but only with the ad network, even small publishers could receive ads appropriate for their sites.

66. The '074 patent uses this standard ad-network technology to serve an ad to user, where the ad has a clickable link that pre-populates a text message. How this technology relates to the claim language will be discussed next.

VI. CLAIMS OF THE '074 PATENT

67. The '074 patent has 30 claims with four independent claims. (Ex. 1001). The independent claims are similar, and are directed to methods performed by mobile devices and servers, and corresponding code. Independent claim 1 can be divided into roughly three parts, corresponding to (1) *standard ad-network*

technology that uses an ad tag to gather user data to send to a server, (2) *standard deeplinking technology* that provides a way to link to a text-messaging application, and (3) the “two-tap” business method. Claim 1 is shown in the table below, with added numbers (1[a] [1b], etc.):

Claim limitation	Category
“1[a]. A method, comprising:	Preamble
“[1b] receiving, from at least one web server, an integration tag included in a webpage;	Ad-network
“[1c] displaying the webpage using a web browser application of a mobile device;	Ad-network
“[1d] sending, from the mobile device, to the at least one web server, and in response to the integration tag executing, (1) a mobile device identifier that identifies the mobile device, and (2) a request to retrieve an identifier associated with at least one of a product or a service presented via the webpage;”	Ad-network
“[1e] receiving, at the mobile device, from the at least one web server, and in response to sending the mobile device identifier and the request to retrieve the identifier, a first signal causing a frame to be displayed with the webpage via the web browser application, the frame including a promotional message associated with joining a subscription list;”	Ad-network
“[1f] sending, to the at least one web server, an indication of a first user interaction with the promotional message;”	Ad-network
“[1g] receiving, from the at least one web server and in response to at least one of (1) the integration tag executing or (2) the indication of the first user interaction, a uniform resource identifier (URI) that deeplinks to and causes the mobile device to automatically (1) switch from rendering the web browser application to rendering a text messaging	Deeplinking / Business method

application and (2) generate a text message with a pre-populated phone number and a pre-populated message body containing a request to join the subscription list;”	
“[1h] sending the text message including the message body prepopulated with the request to join the subscription list in response to a second user interaction with the text messaging application such that the mobile device is subscribed to the subscription list in response to the text message being sent to the pre-populated phone number; and”	Business method
“[1i] receiving an installment of a subscription associated with the subscription list without sending an opt-in communication after sending the text message containing the pre-populated message body.”	Business method

68. In claim 1, a mobile device (*i.e.*, a user’s phone) receives an “integration tag” included in a website. The “integration tag” is configured to cause the sending of a mobile device identifier to a server, which then delivers a response. This is standard ad-network technology. The response can contain a deeplinking URI, which is simply a link in an ad that deeplinks to a mobile app. This is standard deeplinking technology. Clicking on the deeplink causes a pre-filled text-message to appear that provides consent for further marketing activities, as was well-known in the art.

VII. EXAMINATION HISTORY OF THE ’074 PATENT.

69. I understand that during prosecution of the claims that issued as the ’074 patent, the Examiner rejected the claims for having certain technical defects as well as for double patenting, and also in view of prior art not at issue in this declaration. (Ex. 1004, pp. 147-191, 269-303). I also understand that the

Examiner also noted certain prior art considered to be “pertinent” but which was not relied upon. (Ex. 1004, pp. 56-57, 193). I understand that in ultimately allowing the claims after substantial amendments, the Examiner stated that “[t]he prior art discloses a variety of systems and methods for providing deeplinks that can be used to open any application. Additionally, the prior art discloses the use of hashtags within a deeplink and the use of integration tags. Finally, the prior art discloses a variety of systems and methods for subscribing a user through a promotional message.” (Ex. 1004, p. 55). However, the Examiner allowed the claims because, in his view, “[t]he known prior art fails to disclose or suggest each and every limitation together as claimed. Additionally, the examiner cannot determine a reasonable motivation, either in the known prior art or the existing case law, to combine the known limitations to render the claimed invention.” (Ex. 1004, p. 55)

VIII. RELEVANT TIMEFRAME FOR DETERMINING OBVIOUSNESS

70. I understand that obviousness must be evaluated at the time the alleged invention was made. From the first page of the '074 patent (left column, field “[60] Filed”), I understand that the first patent application (a provisional application) was filed on May 26, 2017. I will assume for the purpose of this declaration that May 26, 2017, is the date the purported invention was made, with the understanding that my testimony will also be applicable to some time prior to

May 26, 2017, approximately in the early-to-mid 2017. I may refer to this time period as the “relevant time frame”, and my testimony concerning obviousness is directed to this timeframe, even if I occasionally do not explicitly use a past tense.

IX. LEVEL OF ORDINARY SKILL IN THE ART

71. I understand that the legal question of obviousness must be assessed from the perspective of a person having ordinary skill in the art (“PHOSITA”) at the time of the invention. In my opinion, a PHOSITA in the relevant timeframe would have had a Bachelor’s Degree in computer science and three years of experience in Web-based programming methods, where a higher level of education may substitute for experience and vice-versa.

X. DETAILED EXPLANATION OF THE REASONS FOR UNPATENTABILITY

Ground 1. Claims 1-30 were obvious over Oliver, Agrawal, and Khanna.

72. It is my opinion that claims 1-30 were obvious over U.S. Pat. App. Pub. 2015/0178784 A1 (“**Oliver**”)(Ex. 1011) in view of U.S. Pat. No. 8,015,615 **Agrawal** (Ex. 1005) and U.S. Pat. App. Pub. 2013/0111328 (“**Khanna**”)(Ex. 1034).

73. I understand that Agrawal and Khanna were not of record during the prosecution of the application leading to the ’074 patent. I understand that Oliver was of record, but was not substantively examined. I understand that Oliver was

used to reject certain (different) claims during prosecution of the abandoned '569 application, to which the '074 patent claims priority. (Ex. 1029). I further understand that Oliver was used in combination with other references not raised by the present petition, and was not overcome by the Applicant. (Ex. 1029). Instead, the application was abandoned. (Ex. 1029, pp. 001-002).

A. Overview of the Ground

74. Oliver teaches a “two-tap” method for enrolling users in promotions, where a single user action opens a pre-filled text message, which then only needs to be sent. (Ex. 1011, ¶¶0027-0029, ¶0007). In my opinion, it would have been obvious to implement Oliver’s business method using well-known ad-serving technology, by adding the techniques of Agrawal (an ad-network patent) and Khanna (a mobile-app deeplinking patent).

75. Thus, in the combination, Oliver teaches the known “two-tap” business method, Agrawal provides the standard ad-network functionality, and Khanna provides the standard deeplinking functionality. Agrawal also provides a server that can be easily extended and modified to encompass the functions of Oliver and Khanna, although in principle, the combination could be carried out on any of the servers of the three references.

1. Overview of Oliver (two-tap method)

76. Oliver teaches a “two-tap” business method similar to that of the ’074 patent. In Oliver, a potential customer interacts with an advertisement for a business. (Ex. 1011, ¶0034). The advertisement contains something that the user can interact *with* (using a mobile device), which Oliver calls a “passive entity”. (Ex. 1011, Abstract, ¶0007). The “passive entity” is sometimes a scannable QR code, but Oliver makes clear that a passive entity can be anything that can store link information and be activated by a user’s mobile device. (Ex. 1011, ¶¶0007, 0024, Abstract). The link information that Oliver encodes in a “passive entity” follows standard URI schemes (*e.g.*, “smsto:”), as shown in ¶0028 of Oliver, reproduced here:

[0028] smsto: 543-21: This is where the message goes
and ID=### info etc. To receive your reward TAP
SEND!

(Ex. 1011, ¶0028).

77. When the user activates a passive entity (the first user interaction) containing the smsto: URI, the user's mobile device creates a pre-populated text message, as shown in Fig. 2 of Oliver, reproduced again at right. (Ex. 1011, ¶¶0027-0029). If the user then taps the "send" button on such a message (the second user interaction), the text will be sent, signing the user up for a promotion. Oliver explains:

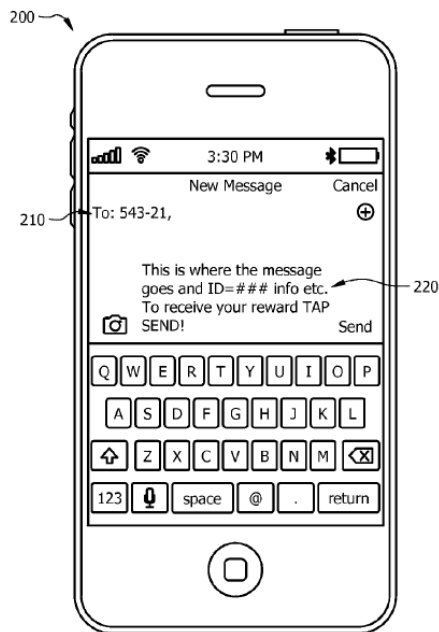


FIG. 2

“As used herein, a ‘passive entity’ is defined to include images, codes, tags, or any other unique identifiers that may be encoded with uniquely identifying information. As used herein, the term ‘activating’ is defined to include either an action by a user that scans, reads, or otherwise inputs the unique information contained in the passive entity into a device operated by the user or the mere presence of a user near a passive entity while equipped with a passive entity receiver, which will input the unique information contained in the passive entity into a device operated by the user. After activating the passive entity, **the user is only required to tap SEND to respond to the promotion.**”

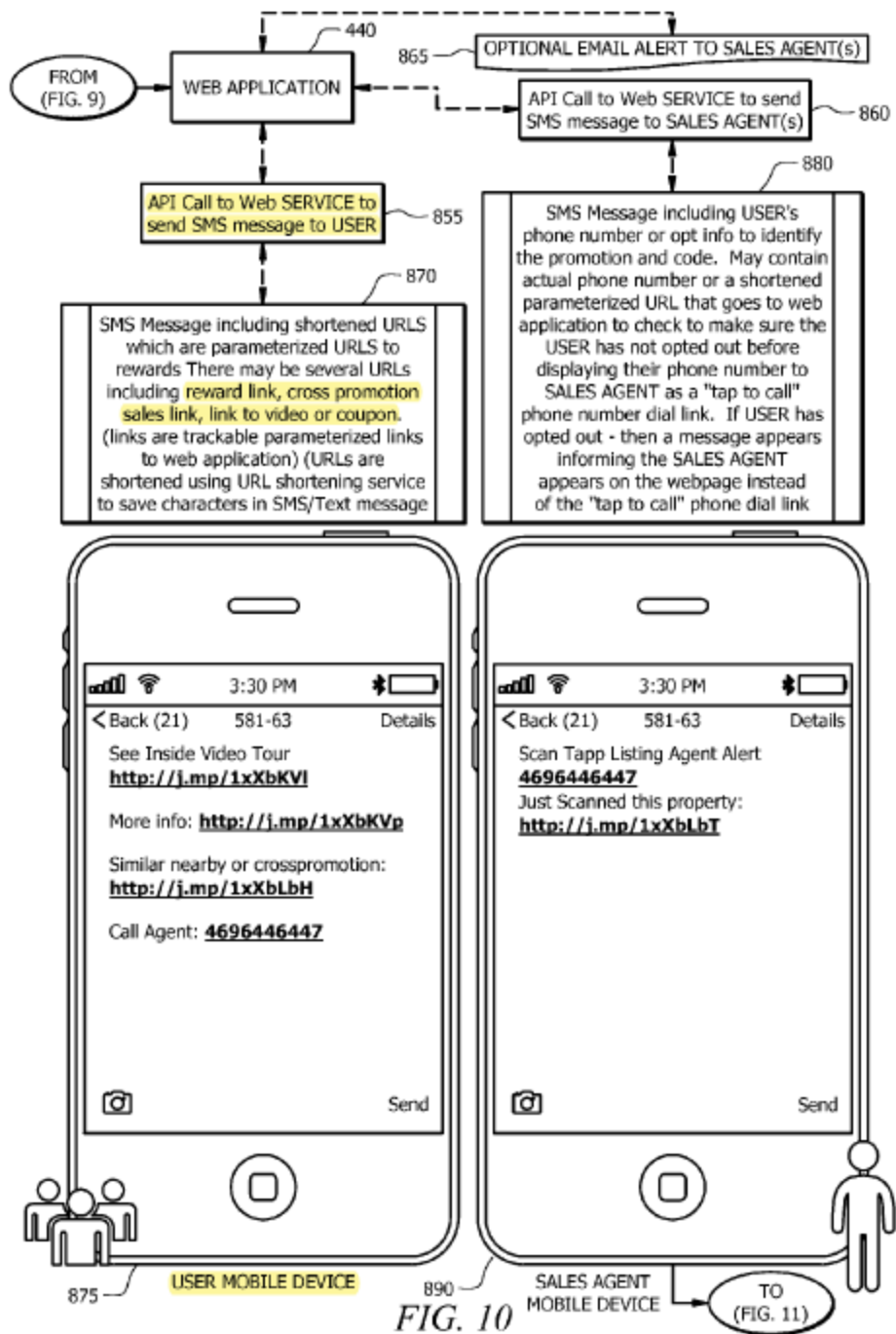
(Ex. 1011, ¶0007)(Emphasis added).

78. Oliver explains—just like the '074 patent—that limiting the number of user actions to two improves the user response process:

“the disclosed principles **streamline the response process** by simply requiring **only two actions** by the user: the scan or other means of activating the passive entity, **and the action of sending** the prepopulated message generated by the activation of the passive entity.”

(Ex. 1011, ¶0009)(Emphasis added).

79. When the text message is sent to the server and the user is registered, Oliver teaches that the server sends response messages to the user offering discounts, rewards, incentives, etc., as shown in Fig. 10 of Oliver, reproduced here with relevant highlighting added:



(Ex. 1011, Fig. 10, ¶¶0051-0053).

80. While Oliver assumes user interaction with an ad, and teaches that the advertisement can be on the Internet, Oliver does not concern itself with the

technology to *deliver* ads to a user on the Internet. (See Ex. 1011, ¶¶0008, 0034, 0036, 0077). This standard technology, however, was disclosed in two Google patent publications: **Agrawal** and **Khanna**.

2. Overview of Agrawal (ad network)

81. **Agrawal** describes an example of a known ad-network system. Ad-network systems were explained above in the Introduction, and generally have a central ad-network server to coordinate activities, advertisers who wish to place ads, publishers who have websites on which ads can be placed, and users who download the websites and see the ads. Such a system is shown in Fig. 1 of **Agrawal**, reproduced here:

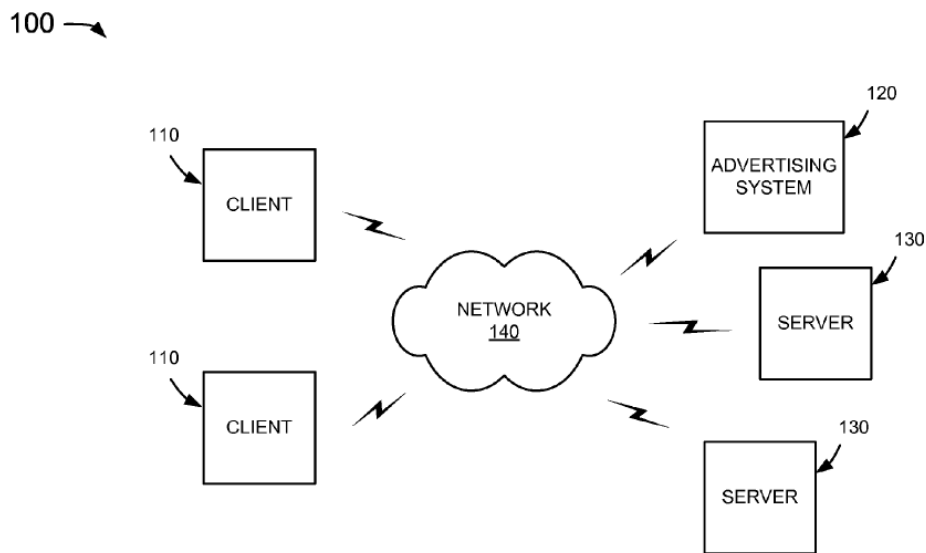


FIG. 1

(Ex. 1005, Fig. 1, 3:62-35). In Fig. 1, the central coordinating server is the “advertising system 120”. (Ex. 1005, 1:62-65, 10:5-6). The users are the “client”

devices 110, which can be mobile devices. (Ex. 1005, 4:10-16). The various publishers of webpages are represented by servers 130. (Ex. 1005, 4:18-20).

82. In Agrawal's system, a user will download a webpage from a server. (Ex. 1005, 5:29-32, 3:46-58). The webpage has an "advertising link", which is code (*e.g.*, JavaScript) that is executed when the user's device loads the webpage. (Ex. 1005, 5:34-39, 3:52-55). Agrawal incorporates by reference another Google patent application ("the '900 application, Ex. 1006)(Ex. 1005, 5:58-63) that further describes the code (ad tags) included in a web page. Upon execution of the code (when the user's browser loads the webpage), the browser sends to the advertising system 120 an "ad query request". (Ex. 1005, 5:29-39). Along with the ad query request, the browser also sends data about the user and the device to the advertising system 120. (Ex. 1005, 5:64-6:46). The advertising system 120 stores the user data, selects one or more ads, and sends the ads back to the browser for display. (Ex. 1005, 5:45-63, 7:54-58). The ads will typically have links that the user can click, as shown in Fig. 4 (links 410) of Agrawal, reproduced here:

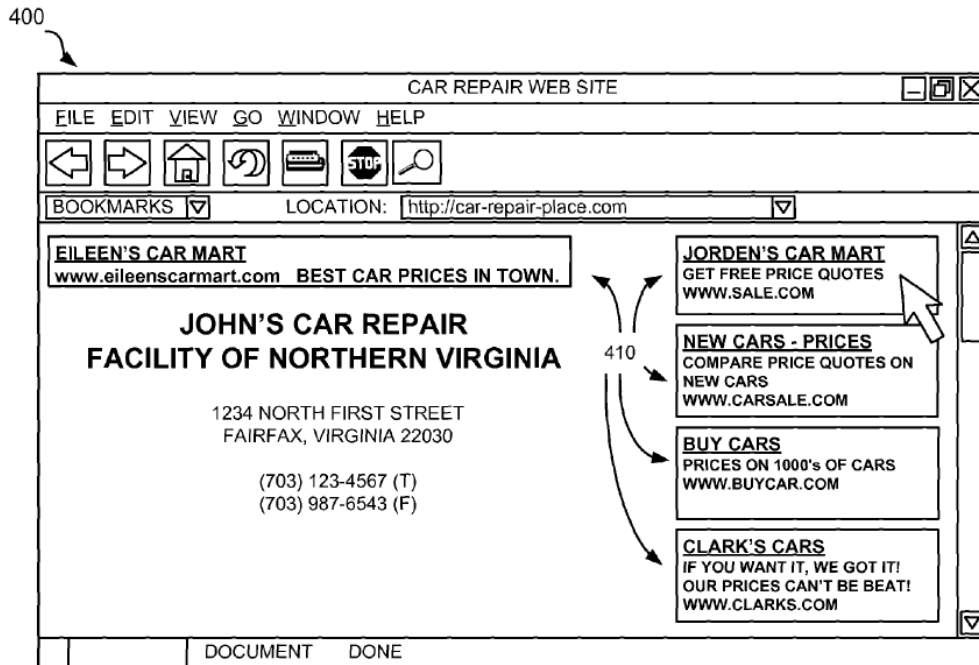


FIG. 4

(Ex. 1005, Fig. 4, 8:12-37).

83. Agrawal does not teach specifically that its advertising links can be links that would open mobile apps, but this functionality was provided by another Google patent publication, **Khanna**.

3. Overview of Khanna (mobile app deeplinking)

84. A natural extension of Agrawal's ad-network is provided by **Khanna**. Khanna teaches a way to allow mobile app deeplinking using links in ads that are served by an ad network (like Agrawal's). (Ex. 1034, ¶¶0031-0044). Like Agrawal's system, Khanna's system also has a central server, called the Advertisement Management System 110 ("AMS 110"), shown in Fig. 1, reproduced here:

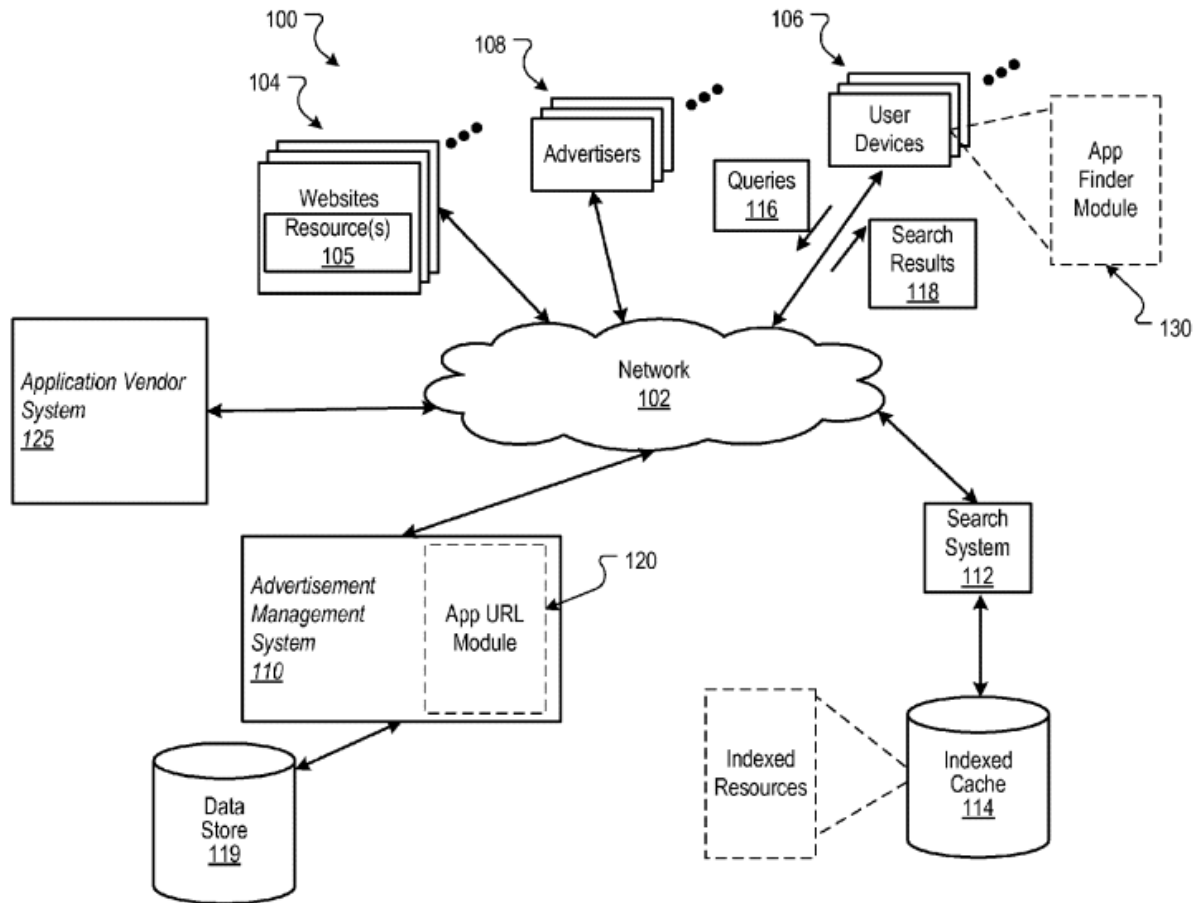


Fig. 1

(Ex. 1034, Fig. 1, ¶0031). The system also has user devices 106 (e.g., smartphones), which can download “resources” 105 (e.g., websites). (Ex. 1034, ¶¶0033-0035, 0103). The websites have advertisement slots that are intended to be filled with ads. (Ex. 1034, ¶0035). When a user views a webpage on a browser, the webpage—just as in Agrawal—makes a request to the central server for advertisements to fill the ad slots. (Ex. 1034, ¶0038). And just as in Agrawal, the AMS 110 responds to an ad request by sending back a response with the ads. (Ex. 1034, ¶0040).

85. To this basic ad-network system, Khanna adds mobile app deeplinking capability. (Ex. 1034, Abstract, ¶0025). Specifically, Khanna allows the ads to have links that specify a target mobile app, and a target product within the mobile app. (Ex. 1034, ¶0025). The links are generated when the central server selects an ad. (Ex. 1034, ¶0044). The central server uses an application URL module 120 (shown above in Fig. 1) to generate or modify the linked URL in the ad such that it will open a target mobile app to a desired starting point relating to the desired product (Khanna states that “product” a product could be, for example, information about a song and artist, ¶0051). (Ex. 1034, ¶0044). Khanna also provides code on the client device to check whether the target application is installed (¶¶0028, 0067), and if so, to carry out the deeplinking function (¶¶0046-0047).

B. Rationale (Motivation) Supporting Obviousness

86. In my opinion, it would have been obvious implement Oliver’s business method by presenting Web links in Internet ads, using an ad system such as Agrawal’s, and to augment such an ad system with Khanna’s techniques to deeplink to mobile applications. I incorporate my opinions above, particularly relating to the Background of the technology prior to May 2017.

87. **First**, it was obvious to implement Oliver’s method using the system of Agrawal (as augmented by Khanna). Oliver’s method would have appealed to

businesses seeking a streamlined method for acquiring the contact information of potential customers, and for acquiring the potential customer's consent to receive marketing communications and proof of possession of the mobile device. (Ex. 1011, ¶¶0004-0006, 0009).

88. A PHOSITA would have found the combination of Oliver with the system of Agrawal/Khanna advantageous. Specifically, a PHOSITA would have desired to enhance the reach of Oliver's ads by using a Web advertising system such as Agrawal's, which allows ads to be targeted at users likely to find the subject matter appealing. (Ex. 1005, 5:45-63). To make use of Agrawal in Oliver's business method, a PHOSITA would have found it obvious to use the concept of mobile app deeplinks that pre-populated text messages (like Oliver's) as links (*e.g.*, links 410) in Agrawal's web ads. (Ex. 1005, 8:20-44)(Ex. 1011, ¶¶0027-0029). Oliver already suggests this, by stating the "passive entity" that contains the link can be essentially anything with which a mobile device can interact. (Ex. 1011, Abstract, ¶¶0007, 0035, 0033, 0048). Oliver further teaches that passive entities can be incorporated into advertising (¶0034), can be incorporated into "electronic visual media" on the Internet, (¶0034), and encode links configured to open a text-messaging application and pre-populate a text message. (Ex. 1011, ¶¶0027-0029). A PHOSITA would have recognized that Agrawal's system provides electronic visual media in the form of advertisements,

and advertisements have links that use URIs. (Ex. 1005, 8:20-44, Fig. 4, 1:18-26).

A PHOSITA further would have found Agrawal's authentication system advantageous to carry out Oliver's express desire to authenticate the user and impose limits on the user's access to promotions. (Ex. 1011, ¶0047).

89. **Second**, it would have been obvious to augment Agrawal's system with Khanna together. Both references arise from the same company (Ex. 1005, Cover)(Ex. 1035), and, as I describe above in ¶¶81-85, both broadly describe similar ad-network systems having user mobile devices that download webpages from publisher web sites, send ad requests to a central advertising management server, and receive ads in response.

90. Both references are intended for use with web-based advertising and teach targeting responsive ads to the content of a webpage. (Ex. 1005, 5:45-63)(Ex. 1034, ¶¶0044, 0003, 0038-0039). Agrawal describes an ad-network central server that provides ads to webpages that can be read on mobile devices. (Ex. 1005, 4:39-5:2, 5:25-8:12). Khanna picks up at this point, operating on ads, to be incorporated into a webpage for reading on a mobile device, that are served from a central server based on an ad request. (Ex. 1034, ¶¶0031, 0038-0039). Given the common origin of the patents and the natural technological relationship between the two references, it would have been obvious to a PHOSITA to extend Agrawal's capability with Khanna's. Both Khanna and Agrawal also express

flexibility in the arrangement of the software and the user and server devices that make up their systems. (Ex. 1034, ¶¶0108, 0122, 0125)(Ex. 1005, 9:65-10:43, 4:23-35).

91. Additionally, a PHOSITA with Agrawal's system would have desired the mobile-app-deeplinking technology of Khanna, because Khanna states that "[a]llowing publishers or advertisers to create content items that deep-link directly to the content within their applications may lead to better user experience as well as create opportunities for the publisher to drive more business." (Ex. 1034, ¶¶0025). Similarly, a PHOSITA considering Khanna's system would have desired to use the system of Agrawal, in order to be able to serve advertisements and to track users and user interactions with advertisements (Ex. 1005, 1:27-33), as expressly desired by Khanna (Ex. 1034, ¶¶0043, 0004). A PHOSITA considering Khanna's system also would have desired to use the system of Agrawal, in order to reduce losses caused by malicious activities. (Ex. 1005, 1:34-58).

92. It also would have been obvious to carry out the server-based software techniques of Oliver, Khanna, and Agrawal at Agrawal's advertising system 120 and/or servers 130, and specifically at a single server, as taught by Agrawal. (Ex. 1005, 4:28-35). Agrawal teaches that its server functionality can be expanded with other server-based functions. (Ex. 1005, 4:15-35, 9:62-10:13). Each of Oliver and Khanna, moreover, provides functionality with a server (like Agrawal's advertising

system), in the form of Khanna's Advertisement Management System 110 and Oliver's Web service 430. (Ex. 1011, ¶0031)(Ex. 1034, ¶¶0031, 0038-0039).

Oliver requires no specific hardware and teaches that its functionality can be implemented via a server. (Ex. 1011, ¶0031, Fig. 9). Oliver, Agrawal, and Khanna all express flexibility in the arrangement of their systems. (Ex. 1005, 4:8-35, 9:65-10:43, 5:3-21)(Ex. 1011, ¶¶0101, 0031, 0025, 0033, 0036)(Ex. 1034, ¶¶0108, 0120-0122). It would have been obvious to combine the functionalities of these references into a single "server" as that term is understood in the '074 patent (Ex. 1001, 4:13-5:47) in order to obtain the benefits of each system as described above.

93. The combination further would have been obvious as the application of known techniques (Oliver and Khanna), for their known functions, in order to improve Agrawal, with no unpredictable results.

C. *Graham* Factors

94. I understand that consideration of the obviousness of the claims requires a consideration of the so-called Graham factors, as I explain above under ¶18. The Graham factors are the level of ordinary skill in the art at the relevant time, scope and content of the prior art, the differences between the prior art and the claim, and any "secondary considerations". I discuss the **level of ordinary skill** in the art above in ¶69.

95. The **scope and content of the prior art** are discussed throughout the Ground.

96. The **differences between the prior art and the claims** are discussed above in the overview of the ground, and in the claim mapping, below.

97. **Secondary considerations** are discussed in ¶¶222-224, below.

98. This discussion of the Graham factors applies similarly to each of the grounds in this declaration.

D. Reasonable Expectation of Success

99. In my opinion, a PHOSITA in the relevant timeframe would have had a reasonable expectation of success in using the prior art in the manner discussed in this petition. Web development, client server communications, ad-network technologies, and mobile-device programming were well-established fields prior to 2017. The techniques of the prior art (and the '074 patent) are implemented in software, are compatible, and were predictable in the relevant timeframe (May 2017). A PHOSITA would have been able to make any necessary modifications to implement the Ground, and in particular would have been able to implement Khanna's software techniques in Agrawal's system, and then to carry out Oliver's method, combining the functions of the servers in each reference. A PHOSITA would have reasonably expected to be able to combine these functions at Agrawal's central server. When I use the term "central server", I am referencing

the somewhat more general concept of a central coordinating server, such as the AMS 110 in Khanna, or the advertising system 120 in Agrawal, which serve the purpose of coordinating activities with respect to users and other server functions, while recognizing that the functions of the servers of all three references could easily be combined into a server function under owned and maintained by a single entity, *e.g.*, Agrawal's advertising system 120.

E. Analogous Art

100. In my opinion, Oliver, Agrawal, and Khanna are analogous art because they are in the same field as the '074 patent (Internet and mobile device technology). (Ex. 1001, 1:20-25)(Ex. 1011, Abstract, ¶¶0003-0006)(Ex. 1005, Abstract)(Ex. 1034, Abstract). Furthermore, the methods of Agrawal, Khanna, and Oliver as described above would have been reasonably pertinent to the problems facing the named inventors, for example, the problems of providing an application deeplinking system and a more convenient text-message signup process. (Ex. 1001, 1:20-42).

F. Claim Mapping

101. This section maps the challenged claims to the relevant disclosures of the prior art, where the claim text appears in bold-italics, and the relevant mapping follows the claim text. I have added numbering and lettering in brackets (*e.g.*, 1[a], [1b]) to certain claim limitations, to facilitate the discussion. Each mapping for a

(narrower) dependent claim also applies to the claim limitation of a superior claim that is being narrowed.

CLAIM 1

“1[a]. A method, comprising:

102. It is my opinion that the combination renders obvious a method comprising the steps outlined below.

“[1b] receiving, from at least one web server, an integration tag included in a webpage;”

103. In the combination, Agrawal teaches that client devices request webpages (referred to as “documents”) that contain integration tags.

(a) receiving, from at least one web server...a webpage

104. Agrawal teaches that its client devices 110 (shown in Fig. 1) download webpages from the Internet. (Ex. 1005, 3:11-12, 3:46-55, 5:29-32). The webpages are called “documents” in Agrawal. (Ex. 1005, 3:46-55).

105. As would be readily apparent to a PHOSITA, webpages downloaded from the Internet are obviously downloaded from “**web servers**”, because that was standard functioning of the Internet. In Agrawal, there are two kinds of web servers disclosed: Agrawal’s server(s) 130 and Agrawal’s advertising system 120, both shown in Fig. 1 of Agrawal, reproduced below with added red-dashed boxes:

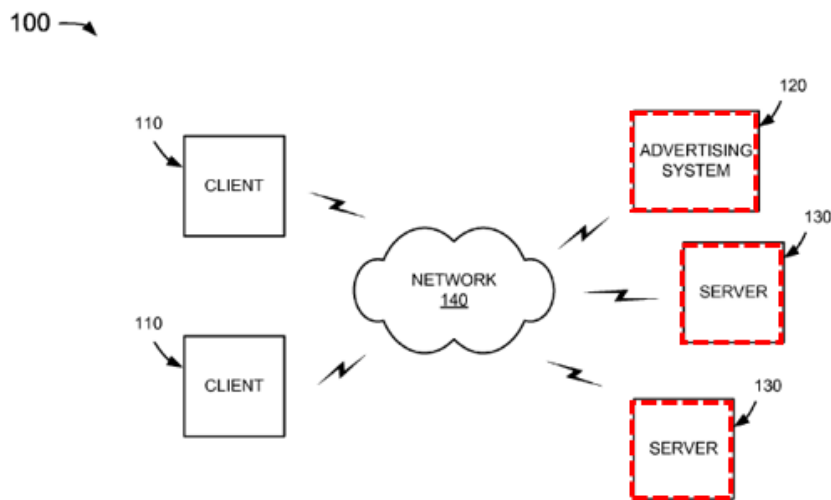


FIG. 1

(Ex. 1005, Fig. 1).

106. Agrawal’s servers 130 are **web servers**. (Ex. 1005, 3:62-4:35)(*see also* Ex. 1006, 011:1-26).¹ Specifically, Agrawal teaches that “servers 130 may include server entities that gather, process, search and/or maintain documents in a manner consistent with the principles of the invention” (Ex. 1005, 4:18-20), where “a common document is a web page” (Ex. 1005, 3:51-52). Agrawal’s advertising system 120 is also a **web server**. (Ex. 1005, 4:16-35)(“Advertising system 120

¹ Citations to Ex. 1006 (the incorporated ’900 application) are in the format {page}:{lines}, with {page} indicating the three-digit numbers (bottom) added.

may include a **server** entity that maintains and provides advertising links.”)(Emphasis added).

107. Agrawal makes clear that the servers 130 and the advertising system 120 can be combined into and implemented as a single server. (Ex. 1005, 4:28-35). In my opinion, it would have been obvious to carry out the server-based software techniques of Oliver, Khanna, and Agrawal at a single server as taught by Agrawal for the reasons I discuss above in ¶92.

(b) integration tag included in a webpage

108. The combination renders obvious **receiving**, from at least one web server, **an integration tag included in a webpage**. In Agrawal, an **integration tag** is code executed by a user’s (client device 110’s) browser when the user’s device **receives** a webpage from a web server, *e.g.*, the combined server I discuss above in ¶107. (Ex. 1005, 5:29-39, 4:10-35)(*see also* Ex. 1006, 026:29-32, 011:7-26, 022:15-023:5, 022:1-6, 021:6-022:21, 013:1-3, 010:26-31, Fig. 4, Fig. 2, Fig. 1).

109. Agrawal first explains that webpages often included embedded code in the form of JavaScript:

“In the context of the Internet, a common document is a web page. **Web pages often include** textual information and may include embedded information (such as meta information, images, hyperlinks, etc.) and/or **embedded instructions (such as**

Javascript, etc.).”

(Ex. 1005, 3:51-55). In the quote above, the referenced “browser” is a web browser on a consumer user device that displays and interprets web pages accessed by the user. (Ex. 1005, 1:18-24, 5:64-6:3, 7:1-3, 8:13-37, Figs. 1 and 4). The “code” is the **integration tag**, because it performs the functions of an integration tag as that term is used in the ’074 patent,

110. Next, Agrawal explains that when a user accesses a web page (“document”), the user’s browser will execute a “piece of code”:

“if a user accesses a document that provides advertising links, an advertisement query request may be generated when the user requests access to the document. In one implementation, the advertisement query request may be automatically generated via the browser used by client 110. **The browser may execute a piece of code, which may cause the advertisement query request to be generated** in response to client 110’s request for the document.”

(Ex. 1005, 5:29-39)(Emphasis added). The “code” is the integration tag. Note that, in the quote above, the referenced “browser” is a web browser on a consumer user device that displays and interprets web pages accessed by the user. (Ex. 1005, 1:18-24, 5:64-6:3, 7:1-3, 8:13-37, Figs. 1 and 4).

111. In my opinion, Agrawal’s code is an “integration tag” because it performs the functions of an integration tag as that term is used in the ’074 patent.

(*compare* Ex. 1001, 4:63-5:3, 11:22-29)(*see also* discussion under limitation [1d] below for the functions carried out by Agrawal's code limitations).

112. Because Agrawal teaches that webpages often include embedded instructions (Ex. 1005, 3:51-55) and that the user's browser "executes a piece of code" when it accesses a webpage (Ex. 1005, 5:29-39), in my opinion, a PHOSITA would have understood, and found it obvious, that the code (integration tag) is **included in a webpage**. As Agrawal indicates (Ex. 1005, 3:51-55), including JavaScript code in a webpage was a standard practice in the relevant timeframe. As I explain above (§58), <script> tags containing JavaScript are executed when a page loads. (Ex. 1012, p. 054)(Ex. 1021, pp. 332, 021, 060).

113. Furthermore, as explained above, Agrawal incorporates by reference (Ex. 1005, 5:58-63) the '900 application (Ex. 1006), which teaches specific examples of code (Ex. 1006, 026:18-028:31) that are "provided to content servers for insertion into all their Web pages." (Ex. 1006, 026:031-034)(*see also* Ex. 1006, 016:30-107:11). These tags disclosed in the incorporated '900 application perform the same function as the code in Agrawal's specification: generating a request for the servers to provide an advertisement. (Ex. 1006, 026:18-028:31)(Ex. 1005, 5:37-40, Fig. 3). In my opinion, these additional disclosures further support the obviousness of including Agrawal's code in a webpage.

"[1c] displaying the webpage using a web browser application of a

mobile device;”

114. Agrawal’s user devices (client devices 110) can be, for example, “a wireless telephone” or “a personal digital assistant (PDA),” which are **mobile devices**. (Ex. 1005, 4:10-16). These mobile devices have **web browser applications** that **display webpages** accessed by the user. (Ex. 1005, 1:18-58, 3:11-25, 5:29-39, 5:64-6:3, 6:30-46, 7:1-3, 8:13-37, Figs. 1 and 4). Agrawal teaches sending advertisements to the **mobile device** for **display** in a **webpage using a web browser application**, thus also demonstrating display of the webpage itself. (Ex. 1005, 8:20-32, Fig. 4).

“[1d.1] sending, from the mobile device, to the at least one web server, and in response to the integration tag executing, (1) a mobile device identifier that identifies the mobile device,”

115. In my opinion, the combination renders this limitation obvious.

(a) a mobile device identifier that identifies the mobile device

116. First, Agrawal teaches collecting **a mobile device identifier that identifies the mobile device**. I note here that, in the ’074 patent:

“the mobile device identifier can be a user agent field collected by the web browser, a platform type or an operating system type operated at the user mobile device. In some implementations, the mobile device identifier can be a user identifier (e.g., a user phone number, a user mobile device identifier, user impression data, and/or any information identifying the user and/or user mobile device).”

(Ex. 1001, 18:34-41).

117. Agrawal teaches that its mobile devices send information that qualifies as a “**mobile device identifier**” per the ’074 patent. (Ex. 1005, 5:45-6:31, 8:20-44). The collected information includes “a type of operating system running on client 110” (Ex. 1005, 6:24-25), “the type of browser being used, the version of the browser being used” (Ex. 1005, 6:4-5), “an Internet Protocol (IP) address associated with client 110” (Ex. 1005, 6:21-22), and “information relating to the user that uses client 110” (Ex. 1005, 6:26-27).

118. Furthermore, in my opinion, the operating system and browser information for the mobile device is the “user agent” information referred to by the ’074 patent as a type of “mobile device identifier”. (Ex. 1001, 18:34-41). The term “user agent” refers to a variety of possible information about the mobile device, operating system and browser that can be sent along with an HTTP request in what is known as the User-Agent field of the HTTP header. (Ex. 1013, ¶10048)(Ex. 1010, p. 022)(Ex. 1016, pp. 328). The user agent information is also accessible by JavaScript operating on a Web page. (Ex. 1021, pp. 364-365, 992).

(b) sending, from the mobile device, to the at least one web server, and in response to the integration tag executing

119. In my opinion, Agrawal teaches and renders obvious that the mobile device identifier is **sent from the mobile device to the at least one web server in response to the integration tag executing**. In overview, this works as follows:

when the user's browser accesses a webpage, the user's browser executes Agrawal's "piece of code" (the integration tag), which is included therein, as discussed above under limitation [1b]. (Ex. 1005, 5:25-44). The execution of the integration tag produces an advertisement query request, which is a request—sent from the mobile device to the advertising system 120—for the system 120 to provide a suitable advertisement for the webpage. (*Id.*). Together with this request, the mobile device sends the mobile device identifiers described above. (Ex. 1005, 2:14-15, 3:29-42, 6:30-46, Fig. 3, step 320, 5:64-6:46, 5:3-21, 10:5-9).

120. More specifically, Agrawal teaches that information constituting the mobile device identifier is collected by the execution of the "piece of code" (the **integration tag**, as explained above under limitation [1b]). Agrawal states:

"The **browser** may **execute a piece of code**, which may **cause a predetermined combination of the above browser/client/user information to be collected**. The information that is collected may include a number of different browser information items, a number of different client information items, and/or a number of different user information items."

(Ex. 1005, 6:34-46)(Emphasis added).

121. Furthermore, the code in the webpage generates an advertisement query request. (Ex. 1005, 5:29-39). Agrawal explains:

"[A]n advertisement query request may be generated when the user

requests access to the document. In one implementation, the advertisement query request may be **automatically generated via the browser** used by client 110. **The browser may execute a piece of code, which may cause the advertisement query request to be generated** in response to client 110's request for the document."

(Ex. 1005, 5:29-39)(Emphasis added). Thus, execution of the integration tag causes the **mobile device** to **send** the advertisement query request.

122. The advertisement query request is an HTTP request (i.e. a web request) directed to advertising system 120 (**at least one web server**) as a direct result of the **execution** of the "piece of code" (**integration tag**). (Ex. 1005, 6:40-43, 5:29-39, 4:16-22, 5:3-65, Fig. 3, 10:3-6). The data collected about the mobile device is also "collected with the advertisement query request", from which a PHOSITA would have understood that piece of code is collecting the information to include with the advertisement query request to be sent to the advertising system 120. (Ex. 1005, 2:14-15, 3:29-42, 6:30-46). The advertising system 120 receives the collected information corresponding to the mobile device identifier. (Ex. 1005, Fig. 3, step 320, 5:64-6:46, 5:3-21, 10:5-9).

"[Id.2] "sending, from the mobile device, to the at least one web server, and in response to the integration tag executing, . . (2) a request to retrieve an identifier associated with at least one of a product or a service presented via the webpage;"

123. In my opinion, the combination renders this limitation obvious.

(a) sending, from the mobile device, to the at least one web server,
and in response to the integration tag executing

124. As explained under claim limitation [1d.1], Agrawal teaches that when a user's **mobile device** loads a webpage, the code (**integration tag**) **executes** and thereby causes an advertisement request query to be generated. (Ex. 1005, 5:29-39).

(b) a request to retrieve an identifier

125. In my opinion, Agrawal's advertisement query request is a **request to retrieve an identifier associated with at least one of a product or a service presented via the webpage**. This is because Agrawal's mobile devices 110 will provide information about the webpage in which the "piece of code" is executing to the Advertisement System 120, in the form of the advertisement query request. Upon receipt of the advertisement query request, advertising system 120 responds with an advertising link associated with a "unique identifier" that is associated with a product or service offered by the webpage.

126. More specifically, Agrawal first teaches that the advertisement query request will contain information about the webpage containing the piece of code that issues the request, as follows:

"[t]he advertisement query request may include information for identifying which advertising links to retrieve for the accessed document. In one implementation, the advertisement query request may include information that causes advertising links that relate to

the content of the document accessed by the user to be retrieved. In this situation, the advertising links may relate to the content of the document. For example, if a user of client 110 accesses a document relating to flowers, advertising links relating to buying flowers over the Internet may be provided. The selection of appropriate advertising links to provide may, for example, be determined by identifying a keyword that represents the content of the document and using the keyword to select the appropriate advertising links. One technique for providing an advertising link based, at least in part, on the content of a document (or web page) is described in U.S. patent application Ser. No. 10/375,900, entitled ‘Serving Advertisements Based on Content,’ and filed Feb. 26, 2003, the entire contents of which are expressly incorporated herein by reference.

(Ex. 1005, 5:45-63, *see also* 3:11-18, 8:20-44).

127. Agrawal next explains that the advertising system 120 will respond to the advertisement query request by sending back an ad (which Agrawal calls an “advertising link”) which can be associated with a unique **identifier**:

“The browser may execute a piece of code, which may cause the advertisement query request to be generated in response to client 110's request for the document. Each advertising link that is returned in response to the advertisement query request 40 may be associated with an **identifier that uniquely identifies the advertising link for this particular transaction (i.e., for this particular user's request for the document and the provisioning of the document and**

associated advertising links).”

(Ex. 1005, 5:39-44, *see also* 6:55-58)(Emphasis added). That is, Agrawal teaches that the Advertising system 120 responds to an ad **request** by **retrieving** a unique **identifier** that identifies the unique webpage, advertising link, and unique user involved in a transaction. Thus, the advertisement query request is a **request to retrieve an identifier**. Agrawal also specifically states that “the advertisement click request may include the unique identifier that is associated with the selected advertising link”. (Ex. 1005, 6:57-58). Because the advertisement click request is a request sent later *from* the mobile device to the server (Ex. 1005, 6:47-58), the fact that the unique identifier is included in the advertisement click request demonstrates that the unique identifier was first **retrieved by** the advertising system 120, and then sent to (*i.e.*, **retrieved by**) the mobile device accessing the webpage.

(c) associated with at least one of a product or a service presented via the webpage

128. In my opinion, Agrawal’s unique identifier is **associated with at least one of a product or a service presented via the webpage** for two reasons. First, Agrawal specifies that the unique identifier “uniquely identifies the advertising link for this particular transaction (*i.e.*, for this particular user’s request for the document and the provisioning of the document and associated advertising links.” (Ex. 1005, 5:39-44). This indicates that the unique identifier is associated with the

document (the webpage). The document, furthermore, can be a document offering a product or service, such as “buying flowers over the Internet” or “car repair” which are at least one of a product or a service. (Ex. 1005, 5:52-54, 8:20-27, Fig. 4). This is shown, for example, in Fig. 4, reproduced here:

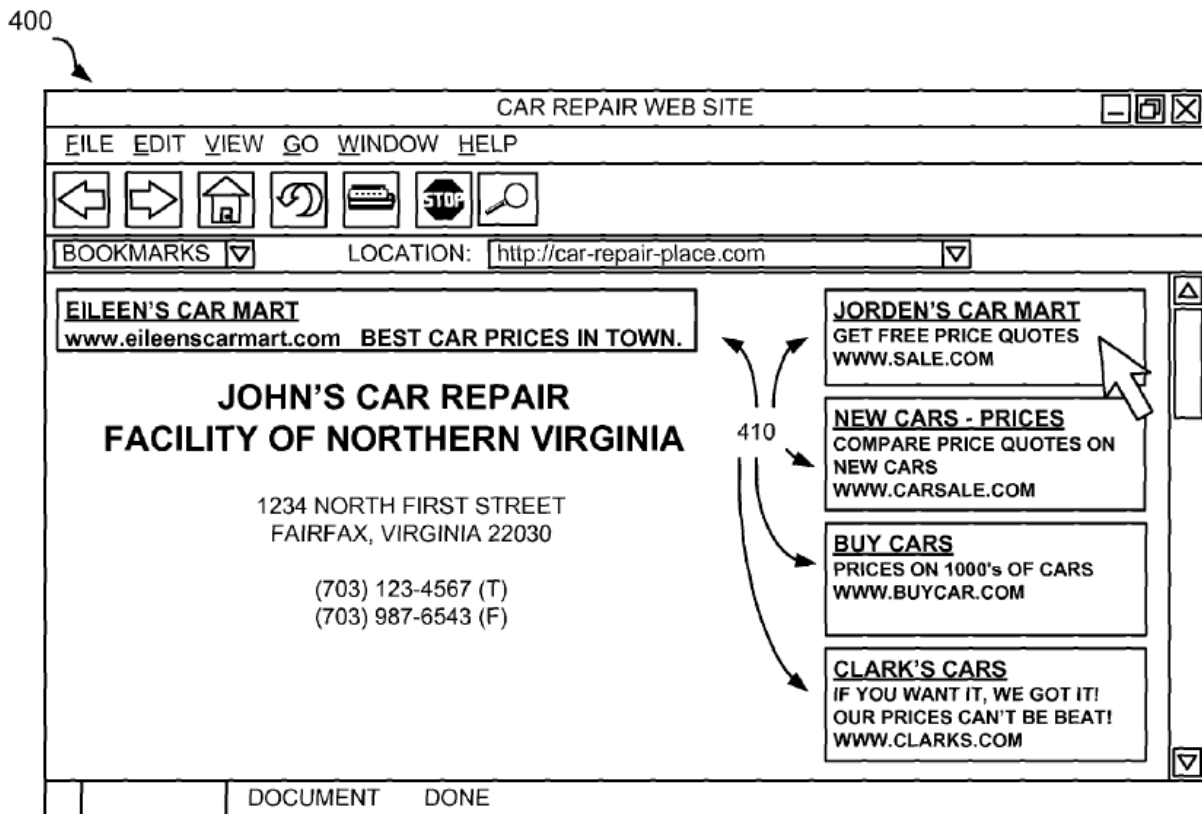


FIG. 4

(Ex. 1005, Fig. 4, 8:13-27). Because the unique identifier is associated with the webpage itself, it is also associated with a product or service presented via the webpage.

129. Second, the unique identifier is associated with one or more advertising links returned by the advertising system 120 in response to the

advertisement query request. (Ex. 1005, 5:39-44). The advertising links are associated with products and services. (Ex. 1005, Fig. 4, 8:13-27).

130. Because the advertising links are incorporated into the webpage, the advertising links are also associated with at least one of a product or a service presented via the webpage, in turn making the unique identifier **associated with at least one of a product or a service presented via the webpage**. Specifically, the unique identifier is correlated with a unique transaction and thus also correlated with the content associated with that unique transaction.

“[1e] receiving, at the mobile device, from the at least one web server, and in response to sending the mobile device identifier and the request to retrieve the identifier, a first signal causing a frame to be displayed with the webpage via the web browser application, the frame including a promotional message associated with joining a subscription list;”

131. In my opinion, the combination renders this limitation obvious by carrying out Oliver’s business application using the technology of Agrawal and Khanna.

(a) receiving, at the mobile device, from the at least one web server, and in response to sending the mobile device identifier and the request to retrieve the identifier, a first signal

132. As explained above under limitation [1d], Agrawal teaches that a **mobile device sends** an advertisement query request (together with a **mobile device identifier**) that is a **request to retrieve the identifier**.

133. **In response** to this request, advertising system 120 and/or servers 130 (**at least one web server**), send, and the **mobile device receive[s]**, for display together with the web page that the user accessed, an ad that displays a link. (Ex. 1005, 3:11-18, 8:20-32, 3:56-4:9, 5:25-63, 8:13-37, 1:18-21). Agrawal's Fig. 4, reproduced again below, shows a webpage with several advertising links 410:

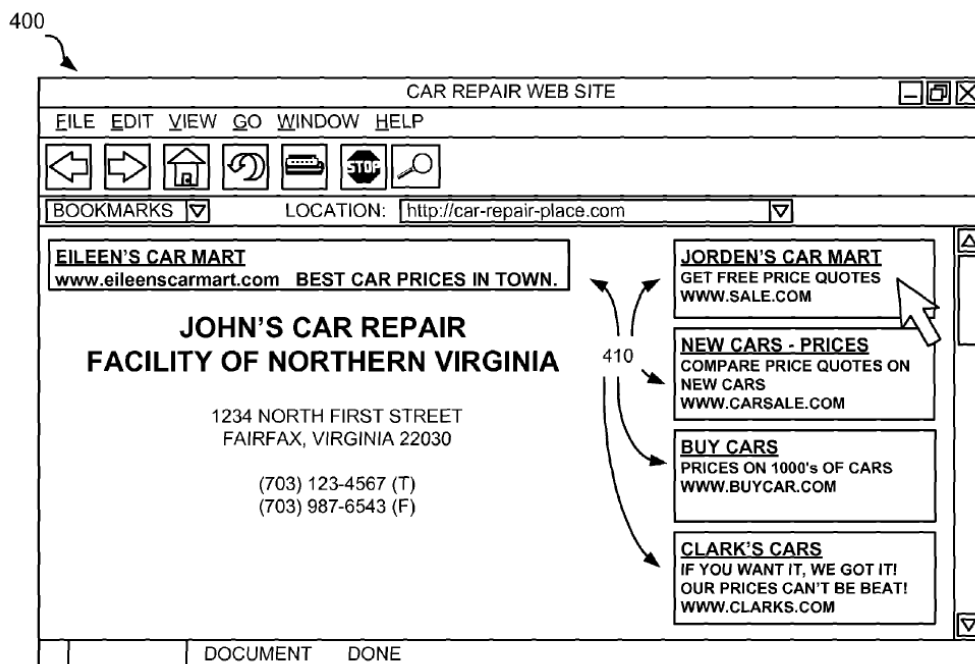


FIG. 4

(Ex. 1005, Fig. 4, 8:13-30). Advertising system 120 and/or servers 130 thus send and the mobile device **receive[s]** a **first signal** sent over a network such as the Internet which **caus[es]** a **promotional message to be displayed with the webpage via the web browser application**. Petitioner notes that in response to an ad request, Khanna similarly teaches returning an advertisement (called a “content item”). (Ex. 1034, ¶¶0035-0039).

(b) causing a frame to be displayed with the webpage via the web browser application, the frame including a promotional message associated with joining a subscription list

134. In my opinion, the combination also renders obvious causing a frame to be displayed with the webpage via the web browser application, the frame including a promotional message associated with joining a subscription list.

135. As taught in Agrawal's expressly incorporated '900 application, the **integration tag** (code) can cause a **promotional message** to display in an iframe (inline frame), which is a **frame**. An iframe defines a region within the document in which a browser can display a separate embedded document. An iframe was a well-known HTML tag frequently used in the relevant timeframe to display ads. (Ex. 1012, pp. 191, 442).

136. Ad Serving Technology teaches that iframes were a known technology for implementing ad tags (a form of integration tag) in the relevant timeframe. (Ex. 1012, p. 191)(“The ‘ad tags’ themselves are short pieces of code that simply reference the location of the ad on the third party ad server. This code is most commonly generated as JavaScript although it can also be generated as an iframe (more HTML friendly script) or for added flexibility, standard HTML.”)(Emphasis added). These ad tag iframes fetch ads for display within the ad tag iframe. (Ex. 1012, p. 442)(“Some Publishers (particularly in the exchange space - See Chapter 8) will serve ads from within an HTML iframe.”)(Emphasis

added). I note here that JavaScript can be used within an iframe. (e.g., Ex. 1006, 031:19-25, Fig. 9B). I discuss the use of iframes in further detail above in ¶¶59-60.

137. The '900 application incorporated into Agrawal provides the following example iframe tag used together with JavaScript to display advertising:

```
<iframe id='google_ads_frame' src='about:blank' hight=80 ... >
...
content of the page
....
<script>
extract important key words like title, meta tags, headers etc, till we
have reached 1K byte limit.
frames['google_ads_frame'].src=http://pagead.google.com/ads?url=<
url>&contents=word1+word2+....
</script>
```

(Ex. 1006, 026:18-028:31, *see also* Figs. 9A-9C). With reference to this exemplary iframe, the script (denoted with <script>) sets the source of the iframe to the Google advertising server located at <http://pagead.google.com/ads>. To this source URL is added the information collected by the script in the form of parameters. Setting the source of the iframe to the URL “instructs the browser to reload the frame.” (Ex. 1006, 026:30-34). The reloading of the frame causes an advertisement request to issue together with information collected by the script provided as parameters. The request elicits a response with an advertisement, which **caus[es] the frame including a promotional message to be displayed with the webpage via the web browser application.**

138. In my opinion, a PHOSITA would have understood how to use iframes predictably, and would have found it obvious to use an iframe to display a

promotional message (ad) consistent with the other teachings of Agrawal because this was a known technique for displaying ads that would have been implementable without unpredictable results. (*See, e.g.*, Ex. 1005, 10:18-29, Fig. 4).

139. The combination renders obvious that **the advertisement includ[e] a promotional message associated with joining a subscription list**. I first note that the informational content of the advertisement, as understood by a human being reading it, is unrelated to the technical functioning of the invention.

140. Oliver nonetheless renders this limitation obvious. Oliver expressly teaches providing Web advertisements with a “passive entity” that a user can use to pre-populate a text message. (Ex. 1011, ¶0034). The text message can be followed up with the delivery “by response message or a later time” of “any type of promotional information”, including coupons, discount codes, entries in sweepstakes, milestones in games, marketing promotions, etc. (Ex. 1011, ¶0025)(Ex. 1011, Figs. 8-10, ¶¶0051-0054). Oliver further explains that its methods “provide streamlined entry into sweepstakes and other giveaways or contest[s].” (Ex. 1011, ¶0047, Figs. 8-10, rn 825-875, ¶¶0051-0054, 0025). Where such benefits are offered, to be provided in subsequent text messages, it would have been obvious to say this in the ad to induce the user to sign up. Oliver also teaches that its purpose is “lead generation” (Ex. 1011, Title), including the

acquisition of contact information for the use in “targeted marketing efforts”. (Ex. 1011, ¶0006, *see also* ¶¶0003-0005, 0046, 0050-0052). Using and offering a subscription list for such purposes was a standard advertising technique in the relevant timeframe.

“[1f] sending, to the at least one web server, an indication of a first user interaction with the promotional message;”

141. Agrawal teaches that a user may select an “advertising link” by clicking on it. (Ex. 1005, 1:18-30, 3:11-25, 3:56-58). This click is a **first user interaction with the promotional message**. A user click generates an “advertisement click request” which is an **indication** of that first user interaction. (Ex. 1005, 6:47-58). This advertisement click request is **sen[t]** to advertising system 120 (**at least one web server**), which can compare the advertisement click request to the previously-received advertisement query request to verify the legitimacy of the advertisement click request. (Ex. 1005, 10:4-13, 7:45-8:13).

“[1g.1] receiving, from the at least one web server and in response to at least one of (1) the integration tag executing or (2) the indication of the first user interaction, a uniform resource identifier (URI) that deeplinks to and causes the mobile device to automatically (1) switch from rendering the web browser application to rendering a text messaging application and”

142. In my opinion, the combination renders this limitation obvious. Agrawal teaches that “[a] ‘link,’ as the term is used herein, is to be broadly

interpreted to include any reference to/from a document from/to another document or another part of the same document”, where a “document is to be broadly interpreted to include any machine-readable and machine storable work product.” (Ex. 1005, 3:46-58). Thus, Agrawal teaches that its advertisements can have “links” in a broad sense, which would include URIs that cause mobile-app deeplinking.

143. While Agrawal does not specifically teach using mobile-app deeplinking, Khanna, another publication from Google, teaches mobile-app deeplinking from ads served from an ad server. (*see* Overview, above). Khanna would have been obvious to use for the reasons provided above under the section entitled “Rationale (Motivation) for the Combination”.

144. Specifically, Khanna’s Advertisement Management Server 110 (implemented at Agrawal’s advertising system 120 in the combination) (**at least one web server**), receives ad requests when users browse web sites (just as in Agrawal). (Ex. 1034, ¶¶0035-0039). In response to an ad request, Khanna teaches returning an advertisement (called a “content item”). (Ex. 1034, ¶¶0035-0039). In addition, Khanna teaches that the content item (advertisement) can be augmented by defining and sending a **uniform resource identifier (URI) to the mobile device**, where the **URI deeplinks** to an application which is different from the

user's web browser. (Ex. 1034, ¶¶0043-0044, *see also* ¶¶0025-0030, 0051, 0038-0041, Fig. 2). Khanna explains:

“The advertisement management system 110 may also include an application URL module 120. The application URL module may be used to generate or modify a formatted URL for a content item....In some implementations, the application URL module may generate a deep-linking application specific landing page URL based on unformatted data elements associated with the content item received from an advertiser 108.”

(Ex. 1034, ¶0044)(Emphasis added). A URL (as used in Khanna) is simply a more-specific form of a **URI**, which complies with RFC 3986. (Ex. 1033). URIs are discussed in further detail at ¶¶40-44 above. Khanna's URIs **deeplink** to an application which is different from the user's web browser because the second application is a specific application associated with the content item, and the browser first checks to see if the application is installed before launching it. (Ex. 1034, ¶¶0067, 0005).

145. To **switch from rendering the web browser application to rendering** another application, Khanna teaches using “**deep-linking**” technology. (Ex. 1034, ¶¶0015, 0025-0030, 0044). In my opinion, “rendering” here simply means drawing the application screen on a display device. In the combination, the second application is a **text messaging application**, as taught by Oliver. (Ex.

1011, ¶¶0027-0029, Fig. 2, ¶¶0051, 0012, 0033, 0035, 0037, cls. 7, 14)(*see also* Ex. 1034, ¶0084).

146. Khanna teaches that when a user clicks on a portion of the content item (advertisement) that activates the URI, the URI **causes the mobile device to automatically switch** from a first mobile application to a different mobile application. Khanna states:

“When a user selects (e.g., clicks on) the content item, if the application is installed, the application may be launched with the associated product related data elements as inputs so that the content (e.g., a landing page) related to the product is displayed in the application.”

(Ex. 1034, ¶0027). Khanna teaches that this transition is **automatic** after detecting user input. (Ex. 1034, ¶¶0006, 0008, 0010, 0013-0014, 0046, 0062-0063).

Khanna states:

“The application may be launched **automatically** when the application is determined to be previously installed.... In some implementations, the launch may be initiated by user input (e.g., input received through pointing device 304).”

(Ex. 1034, ¶0063)(Emphasis added). Oliver likewise teaches the use of URIs (*e.g.*, using the smsto: schema) to cause a transition to a text-messaging app. (Ex. 1011, ¶¶0027-0029). URIs are described further above in ¶¶40-44.

147. In my opinion, it would further have been obvious to cause a user click on an advertising link in Agrawal (*i.e.* a **promotional message**) to **switch from rendering the web browser application to rendering a text messaging application**, as taught by Oliver, for the reasons provided above under ¶¶86-93. (Ex. 1011, ¶¶0027-0029, Fig. 2, ¶¶0051, 0012, 0033, 0035, 0037, cls. 7, 14). Such a use of URIs was well-known in the relevant timeframe, as discussed in the Introduction.

148. As explained under claim limitation [1b], Agrawal's integration tag executing results in the generation of an advertisement query request. As explained under claim limitation [1d.2], an advertising link is returned in response to an advertisement query request. Because, in the combination, Agrawal's advertisement link includes a URI, the **URI is receiv[ed] from the at least one web server and in response to the integration tag executing.**

“[1g.2] a uniform resource identifier (URI) that deeplinks to and causes the mobile device to automatically . . . 2) generate a text message with a pre-populated phone number and a pre-populated message body containing a request to join the subscription list;”

149. In my opinion, the combination renders this limitation obvious by carrying out Oliver's business method using the functionality of Agrawal and Khanna.

150. As explained above under limitation [1g.1], Khanna describes deeplink URIs for inclusion into an advertisement, where the deeplinks (when clicked by a user) cause the mobile device to transition to a different application. (Ex. 1034, ¶¶0044-0046). This different application, moreover, can receive input data that was encoded into the link. (Ex. 1034, ¶0061). For example, Khanna states:

“[I]f the application was previously installed 215, **the application may be launched 230 with a data element identifying a product** associated with the content item as an input parameter. For example, where the content item includes **a URL with data elements that specify a song**, the application may start up by loading a landing page by an advertiser’s server that displays information regarding the song and an icon that when selected will allow the user to complete a purchase transaction that will result in the download of the song to the user’s device.”

(Ex. 1034, ¶0061)(Emphasis added). It is clear that the data elements are incorporated into the deeplink URI, such that they are configured to guide the target application to a specific starting point, with the correct information. (Ex. 1034, ¶0044). Oliver similarly teaches using URIs to transition to a text-messaging app, where the URI encodes information into the URI itself in order to pre-populate a text message. (Ex. 1011, ¶¶0027-0029).

151. In my opinion, it would have been obvious, for reasons discussed above under ¶¶86-93, to implement Oliver’s business method together with Khanna and Agrawal. Oliver’s business method includes using application deeplinks to **automatically generate a text message with a pre-populated phone number and a pre-populated message body**. (See, e.g., Ex. 1011, ¶¶0026-0028, 0033, 0037, 0046, 0047). For example, Fig. 2 of Oliver, reproduced below, shows a mobile device 200 with a pre-populated text message (in a text-messaging application) having an address 210 and a message body 220:

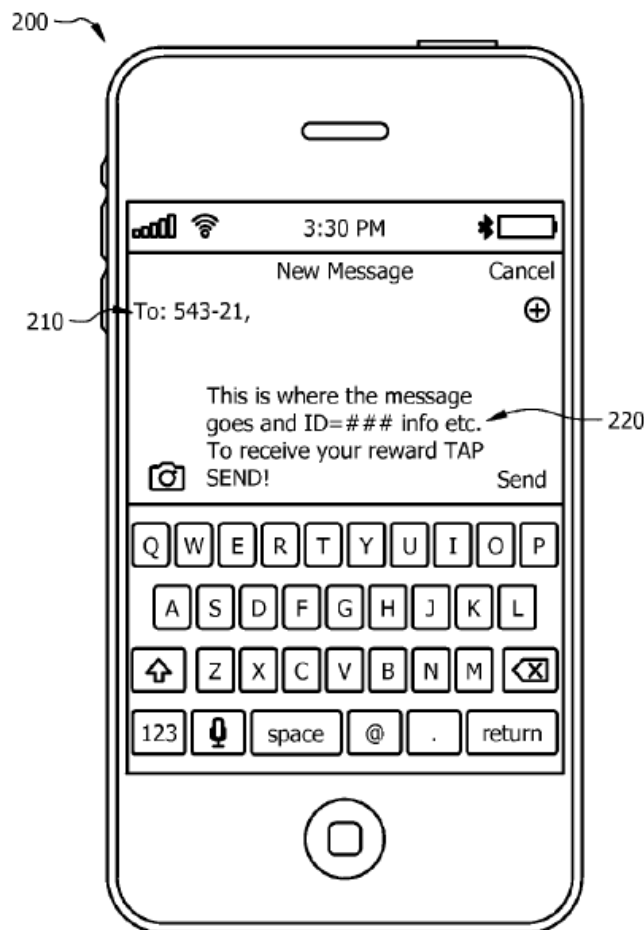


FIG. 2

(Ex. 1011, Fig. 2, ¶¶0027-0030, 0012). The address can be a destination number or short code (**phone number**). (Ex. 1011, ¶¶0008, 0013, 0026, 0029, Fig. 3).

152. Oliver explains that the pre-populated text message arrives from a mobile-app deeplink in a format exemplified as: “Smsto: 543-21: This is where the message goes and ID=# info etc. To receive your reward TAP SEND.” (Ex. 1011, ¶¶0028). This was a well-known form of link for deeplinking to an SMS (text messaging) application. (*citing* Ex. 1016, p. 584); *see also* paragraphs 40-44, above. Oliver teaches that when the deeplink is activated, a “pre-coded message body 220 and destination number or short code 210 will populate the user’s screen 200 using the device’s SMS application” and that this information is “**automatically**” prepopulated and the resulting message “**automatically generate[d]**.” (Ex. 1011, ¶¶0029, 0026, 0027).

153. Oliver teaches that the deeplink can be activated by scanning a QR code, but also that it can be activated by any type of “passive entity”, including “images” on the Internet. (Ex. 1011, ¶¶0034, 0007, 0024). In my opinion, it would have been apparent to PHOSITA that a deeplink, such as the ones used by Oliver, could also be activated by other interactions, as explained in Khanna. (Ex. 1034, ¶¶0027, 0063). Using Khanna’s deeplinks to activate a prepopulated text message as taught conceptually by Oliver would result in **automatically**

generat[ing] a text message with a pre-populated phone number and a pre-populated message body.

154. It would further be obvious for the **pre-populated message body** to **contain[] a request to join the subscription list** for the reasons set forth under claim limitation [1e].

“[1h] sending the text message including the message body prepopulated with the request to join the subscription list in response to a second user interaction with the text messaging application such that the mobile device is subscribed to the subscription list in response to the text message being sent to the pre-populated phone number; and”

155. As explained under claim limitation [1g.2], the combination renders obvious **generating a text message with a pre-populated phone number and a pre-populated message body containing a request to join a subscription list.**

Oliver teaches that once the prepopulated text message is generated, “the users [sic] only action required is to press or tap ‘Send.’” (Ex. 1011, ¶0029)(*see also* Ex. 1011, ¶¶0026, 0033). In my opinion, a PHOSITA in the relevant timeframe would have understood and found obvious that the tap on the “send” button of a messaging application would trigger **sending** of the **text message** to the **pre-populated destination phone number**. Devices in the relevant timeframe had user interfaces, such as touch screens. When a user interacted with a touch screen, the device’s hardware and operating system would detect the user input. (*e.g.*, Ex. 1011, Fig. 1,

¶0007). This tapping send is a **second user interaction with the text messaging application**.

156. Oliver teaches that the reception of the prepopulated text message **subscribe[s] the mobile device** in a promotion. For example, the message can “provide streamlined entry into sweepstakes and other giveaways or contest[s].” (Ex. 1011, ¶0047, Figs. 8-10, rn 825-875, ¶¶0051-0054, 0025). In my opinion, it would be obvious that **in response to a text message** containing a request about a product or service, (e.g., to receive coupons or discounts, Ex. 1011, ¶0025) (see claim limitation [1g.2]) that the **mobile device** be in fact **subscribed to the subscription list** in order to facilitate “targeted marketing” as taught by Oliver. (Ex. 1011, ¶0006).

“[1i] receiving an installment of a subscription associated with the subscription list without sending an opt-in communication after sending the text message containing the pre-populated message body.”

157. As shown in Figs. 8-10 of Oliver, enrollment in a promotion takes place without receiving any additional information from the mobile device. (Ex. 1011, Figs. 8-10, ¶¶0051-0054, ¶0009). Oliver explains that this “two-step” processing allows the user to respond to ads offering promotions in a “streamlined” fashion. (Ex. 1011, ¶¶0046-0047, 0009, 0023). Accordingly, in my opinion, it would be obvious that enrollment in and participation in the subscription is

completed **without sending an opt-in communication after sending the text message containing the pre-populated message body**. This is because no further communications are needed in order enroll and participate in the subscription. (*See also* Ex. 1004, p. 123). I further understand that, in patent law, a reference need not state a feature's absence in order to disclose a negative limitation.

158. It would also be obvious to arrange the combination to cause **receiving an installment of a subscription associated with the subscription list** after having been enrolled and without any further action by the mobile device user. For example, Oliver teaches sending a reply message to the user's mobile device which can include "coupons or discount codes delivered to the user either by response message or a later time, instant-win game chances, sweepstakes entries, . . . registration of a purchased product, etc." (*Id.*). (Ex. 1011, ¶¶0025, 0039-0040, 0045, 0050, 0053). In my opinion, a PHOSITA would recognize that a reply message could be used to deliver **an installment of a subscription associated with the subscription list** to the mobile device.

CLAIM 2

"2. The method of claim 1, wherein a size of the frame relative to a size of the webpage is configurable."

159. In my opinion, the combination renders obvious making the size of the frame relative to a size of the webpage (i.e., the size of the advertisement) configurable. The '900 application, which is incorporated into Agrawal, teaches

that ad serving parameters include “an **absolute position** of the ad on the page on which it is served, a **position** (**spatial** or temporal) of the ad relative to other ads served, an **absolute size of the ad**, [and] a **size of the ad relative to other ads**.”

(Ex. 1006, 015:28-016:16)(emphasis added)(*see also* Ex. 1006, 042:3-20, 013:24-32, 006:16-18, 010:29-31)(*see* Ex. 1005, Fig. 4)(depicting ads of different sizes).

In my opinion, and as a PHOSITA would recognize, the presence of programmable parameters indicates configurability.

160. The incorporated '900 application further explains that the system “transmit[s] information about the ads and how, when, and/or where the ads are to be rendered (e.g., **position**, click-through or not, impression time, impression date, **size**, conversion or not, etc.” (Ex. 1006, 011:18-26, 012:6-20, 08:28-31, 015:28-016:16, 042:3-19)(emphasis added). As set forth above under claim limitation [1e], the '900 application further teaches that advertisements can be displayed within an iframe, which is a **frame**. (Ex. 1006, 026:17-028:31, 029:11-030:27, 031:13-15, 042:3-19, 043:4-14, Fig. 9A)(*see also* Ex. 1012, 165, 191, 442). The examples of iframes shown in the '900 application have configurable sizing parameters such as “height”. (Ex. 1006, 026:18-028:31). Agrawal thus discloses that the **sizes** of ads (and thus the **frames** in which ads are presented) are **configurable**. In my opinion, a PHOSITA would understand that these sizes are configurable **relative to a size of the webpage** because changing the size of the

frame containing the ad changes its size relative to the size of webpage, and because it was obvious to size ads to account for differences in display sizes experienced by each user. It was also known how to display ads of different sizes and it was standard practice to have ads of different sizes on a given webpage where advertisers could pay different amounts for different size ads. (See, e.g., Ex. 1012, pp. 165-169, 178, 284-285).

CLAIM 3

“3. The method of claim 1, wherein the mobile device is subscribed to the subscription list (1) with no user input other than the first user interaction and the second user interaction and (2) without the text messaging application being redirected to a different application after the text message is sent.”

161. The combination renders obvious claim 3. As set forth under claim 1, the mobile device is subscribed to the subscription list after only two user interactions—the **first user interaction** (clicking on an advertisement) and the **second user interaction** (sending the text message). Accordingly, **the mobile device is subscribed to the subscription list with no user input other than** these two interactions. (See also Ex. 1004, pp. 122-123). I further understand that, in patent law, a reference need not state a feature’s absence in order to disclose a negative limitation.

162. In my opinion, it would also (and independently) have been obvious for the **text messaging application** not to be **redirected to a different application**

after the text message is sent so that the user can, for example, view a reply or confirmation message, and/or the **installment of a subscription**. (See, e.g., Ex. 1011, ¶¶0025, 0039, 0050, 0051)(see claim limitation [1i]). I note here that the technical disclosure of the '074 patent does not specifically describe a mobile device being subscribed to the subscription list without the text messaging application being redirected to a different application after the text message is sent.

CLAIM 4

“4. The method of claim 1, wherein the mobile device identifier is based on a user agent field associated with the web browser application.”

163. See above, claim 1, limitation [1.d1]. Agrawal teaches obtaining **user agent** information such as browser type, client (mobile device) type, and operating system information). (Ex. 1005, 5:45-6:46, 8:20-33). As explained above, Agrawal discloses, and it further would have been obvious, to use “user agent” information as user data sent together with the advertisement query request of Agrawal. As I explain under claim element [1d.1], “user agent” refers to any of a variety of data about the user’s device and software that is available to webpages and servers interacting with the user’s device. In my opinion, it would have been obvious to access user agent data using JavaScript executed by a browser by using, for example, the Navigator object’s userAgent field. (Ex. 1021, pp. 364-365, 992)(Ex. 1016, pp. 324-338). It would have been obvious to do so, for example to

identify the operating system of the mobile device to configure a URI, as explained under claim 5, because the user agent states the identity of the operating system directly. (*Id.*).

CLAIM 5

“5. The method of claim 1, wherein: the mobile device identifier identifies a first operating system executing on the mobile device; and the URI is compatible with the first operating system and not a second operating system different from the first operating system.”

164. See claim 1, limitation [1.d1] and claim 4. As set forth under claim 4, Agrawal’s user data (**the mobile device identifier**) can **identify the operating system executing on the mobile device**. (Ex. 1005, 5:45-6:46, 7:19-28, 8:20-33).

165. The combination further renders obvious that **the URI is compatible with the first operating system and not a second operating system different from the first operating system**. It was known in the relevant timeframe that different operating systems could have different URI schema for text messaging. In my opinion, it would have been obvious to configure the URI such that it is specifically compatible with the operating system executing on the mobile device, because otherwise the URI would not work correctly. (*See also* Ex. 1004, pp. 122-123). If a URI is compatible with one operating system, it will not be compatible with another operating system that has a different requirement. In my opinion, it would have been a relatively simple matter in the relevant timeframe to use JavaScript to determine, at the server, the operating system for the mobile device

and, using that information, to configure the URI based on the operating system detected based on known click-to-text URI schema. (Ex. 1016, pp. 324-338, 584-590).

CLAIM 6

“6. The method of claim 1, wherein the first user interaction and the second user interaction are the only user interactions between receiving the first signal causing the frame to be displayed and subscribing the mobile device to the subscription list.”

166. In my opinion, claim 6 is rendered obvious by the combination for the same reasons as claims 1 and 3. Oliver discloses only the interaction of a user with a passive entity, followed by prepopulation of text message and clicking “send”. (Ex. 1011, Fig. 8). As noted above, I further understand that, in patent law, a reference need not state a feature’s absence in order to disclose a negative limitation. Here, the recitation of “the only user interactions...” is a negative limitation.

CLAIM 7

“7. The method of claim 1, wherein the installment of the subscription is received via text message without an entity associated with the at least one webserver charging a fee.”

167. As Oliver explains in conjunction with Figures 8-10, the device is enrolled in the promotion without charging a fee to a user of the mobile device. (Ex. 1011, Figs. 8-10, ¶¶0051-0054).

168. Furthermore, in my opinion, it would have been obvious for Oliver's system to send an installment without charging a fee for user enrollment, because the intent of Oliver's system was the generate leads for further sales with the minimum amount of friction, and charging a fee would obviously reduce user acceptance of the promotion. (Ex. 1011, Title, ¶¶0025, 0046-0047). It similarly would have been obvious to deliver an installment of the subscription for free in order to facilitate user participation in a promotion. As noted above, I further understand that, in patent law, a reference need not state a feature's absence in order to disclose a negative limitation.

CLAIM 8

“8[a]. A non-transitory processor-readable medium storing code configured to be executed by a processor of a mobile device, the code including instructions configured to cause the mobile device to:”

169. As explained under claim limitation 1[a], Agrawal's user devices (client devices 110) can be **mobile devices**. (Ex. 1005, 4:10-16). Agrawal teaches that its client devices execute code (4:39-425:37-39, 6:30-36, 7:33-35), and possess **a non-transitory processor-readable medium** as well as a **processor**, for **executing** the code. (Ex 1005, Fig. 2, 4:39-57).

170. In general, a PHOSITA would have understood that the methods of the prior art were software-based methods that have instructions to be carried out

by a processor, and that the instructions needed to be stored in a non-transitory processor-readable medium.

171. In my opinion, for the reasons described above in ¶¶86-93, it would have been obvious to arrange **code configured to be executed by a processor of a mobile device to include instructions configured to cause the mobile device to perform the functions of the recited in each of claim limitations [8b]-[8i].**

172. The body limitations of claim 8 are similar to those of claim 1, except that claim 8 is directed to the method steps of claim 1 carried out via a non-transitory processor-readable medium storing code which is executed by a processor of a mobile device whereas claim 1 is in method form. These steps are obvious for the same reasons as set forth for claim 1.

CLAIMS 9-14

173. Claims 9-14 are similar to claims 2-7 except claims 9-14 are directed to the steps carried out via a non-transitory processor-readable medium storing code which is executed by a processor of a mobile device whereas claim 1 is in method form. In my opinion, claims 9-14 were obvious for the same reasons as claims 2-7, respectively.

CLAIM 15

“15. The mobile device comprising the non-transitory processor readable medium of claim 8 and the processor.”

174. In my opinion, the combination renders obvious claim 15 is obvious for same reasons as set forth under claim 8.

CLAIM 16

“16[a]. A method, comprising:”

175. In my opinion, the claimed method is obvious for the reasons set forth under claim 1 and as further explained below.

“[16b] defining an integration tag configured to be included in a webpage such that the integration tag executes when a web browser application of a mobile device accesses the webpage;”

176. See above, claim 1, limitation [1b]. The present limitation additionally requires “defining an integration tag” and that the “the integration tag executes when a web browser application of a mobile device accesses the webpage”.

177. In my opinion, the combination renders this limitation obvious. Specifically, because Agrawal teaches an integration tag, it was obvious that the tag needed to be “defined”, because otherwise it would not exist.

178. Furthermore, Agrawal renders obvious that the integration tag (Agrawal’s piece of code) **executes when a web browser application of a mobile device accesses the webpage**. Ex. 1005, 5:32-39). Specifically, as Agrawal teaches, mobile device browsers loading webpages that contain code (*e.g.*,

JavaScript) will execute when the webpage is loading. Ex. 1005, 5:32-39).

Browsers in the relevant timeframe executed scripts present within a webpage at the time of loading the webpage. This is discussed further in ¶¶58-62, above.

“[16c] receiving, from the mobile device and in response to the integration tag executing at the mobile device, (1) a mobile device identifier that identifies the mobile device, and (2) a request to retrieve an identifier associated with at least one of a product or a service presented via the webpage;”

179. The combination renders obvious claim limitation [16c] for the same reasons set forth under claim limitation [1d]. In the combination, advertising system 120 and/or client servers 130 **receive** the claimed information sent by the mobile device. (Ex. 1005, 2:14-15, 3:29-42, 6:30-46, Fig. 3, step 320, 5:64-6:46, 5:3-21, 10:5-9).

“[16d] sending, to the mobile device and in response to receiving the mobile device identifier and the request to retrieve the identifier, a first signal configured to cause a frame to be displayed with the webpage via the web browser application, the frame including a promotional message associated with joining a subscription list;”

180. The combination renders obvious claim limitation [16d] for the same reasons as set forth under claim limitation [1e]. In the combination, advertising system 120 and/or client servers 130 **send[]** the claimed **first signal to the mobile device in response to receiving** the claimed information sent by the mobile device.

“[16e] receiving, from the mobile device, an indication of a first user interaction with the promotional message;

181. In my opinion, the combination renders obvious claim limitation [16e] for the same reasons as set forth under claim limitation [1f]. In the combination, advertising system 120 and/or client servers 130 **receive** the claimed **indication**.

“[16f] sending a uniform resource identifier (URI) to the mobile device that deeplinks to and causes the mobile device to automatically (1) switch from rendering the web browser application to rendering a text messaging application and (2) generate a text message with a pre-populated phone number and a pre-populated message body containing a request to join the subscription list in response to at least one of (1) the integration tag executing or (2) the indication of the first user interaction;”

182. The combination renders obvious claim limitation [16f] for the same reasons as set forth under claim limitation [1g]. In the combination, advertising system 120 and/or client servers 130 **send** the URI to the mobile device.

“[16g] receiving the text message including the pre-populated message body to the pre-populated phone number in response to a second user interaction with the text messaging application;”

183. In my opinion, the combination renders obvious claim limitation [16g] for the same reasons as set forth under claim limitation [1h]. In the combination, advertising system 120 and/or client servers 130 **receive** the text message. As Oliver explains, the “destination number or code is ‘hosted’ at a web service 430 that interfaces with the mobile data and telephone network.” (Ex. 1011, 0008,

0038, 0040), and it was obvious to combine web service 430 with Agrawal's advertising system 120 for the reason provided under the Rationale (Motivation) for the Combination. In the combination, advertising system 120 and client servers 130 are associated with the **pre-populated phone number** and thus host the destination number and **receives the text message**.

“[16h] subscribing the mobile device to the subscription list based on receiving the text message; and”

184. In my opinion, the combination renders obvious claim limitation [16h] for the same reasons as set forth under claim limitation [1i]. In the combination, advertising system 120 and/or client servers 130 **subscribe the mobile device to the subscription list on receiving the text message**.

“[16i] delivering an installment of a subscription associated with the subscription list without receiving an opt-in communication after receiving the text message containing the pre-populated message body.”

185. In my opinion, the combination renders obvious claim limitation [16i] for the same reasons as set forth under claim limitation [1j]. In the combination, advertising system 120 and/or client servers 130 **deliver** an installment of a subscription.

CLAIM 17

“17. The method of claim 16, wherein a location of the frame on the

webpage is configurable.”

186. In my opinion, the combination renders obvious that the location of the advertisement on the webpage (and the frame including the advertisement) be configurable. The '900 application, which is incorporated into Agrawal, teaches that ad serving parameters include “an absolute **position** of the ad on the page on which it is served, a **position** (**spatial** or temporal) of the ad relative to other ads served, an absolute size of the ad, a size of the ad relative to other ads” and that a content server can transmit information regarding the “**position**” of the ads. (Ex. 1006, 015:28-016:22, 011:18-26)(Emphasis added). Furthermore, “an advertiser may be able to specify that its ad is . . . no lower than a certain **position**.” (Ex. 1006, 015:28-015:22)(Emphasis added)(*see also* Ex. 1006, 042:3-20, 013:24-32)(Ex. 1005, Fig. 4)(displaying ads in different positions). In my opinion, it was obvious that the location of the ad on the webpage be configurable so that the ad can be located in a manner so as to be visible to the user and also so that it can be placed high or lower, or in a more or less favorable position on the webpage, depending on how much an advertiser pays for the ad.

CLAIM 18

“18. The method of claim 16, wherein the text message is a first text message, the method further comprising sending, to the mobile device via the user phone number, a second text message confirming the subscription to the subscription list in response to receiving the

first text message.”²

187. In my opinion, the combination renders claim 18 obvious. Oliver discloses providing “any type of promotional information” in a reply message in response to having received a first text message. (*See, e.g.*, Ex. 1011, ¶¶0025, 0039, 0050, Fig. 5). It would be obvious to **send this second (reply) text message to the mobile device via the user phone number** to confirm that the device has in fact been subscribed to the subscription list in response to the user’s request so that the user can be notified that their request was successful.

CLAIM 19

“19. The method of claim 16, wherein the installment of the subscription is delivered (1) with no user input other than the first user interaction and the second user interaction and (2) without redirecting the text messaging application to a different application after the message is sent.”

188. Claim 19 is a combination of claim 16, limitation [16i], which states that an installment of the subscription is delivered, and claim 3, which specifies conditions (1) and (2). The combination thus renders obvious claim 19 for the same reasons as set forth under claim limitation [16i] and claim 3.

² I note that claim language “the user phone number” does not appear earlier in the claim. I treat it as though it refers to the phone number associated with the user’s mobile device for purposes of my opinions.

CLAIM 20

“20. The method of claim 16, wherein the installment of the subscription is delivered via text message to the mobile device without a provider of the subscription service charging a user of the mobile device any fee.”

189. Claim 20 is a combination of claim 16, limitation [16i], which states that an installment of the subscription is delivered, and claim 7, which specifies the negative limitation. The combination thus renders obvious claim 20 for the same reasons as set forth under claim limitation [16i] and claim 7.

CLAIM 21

“21. The method of claim 16, wherein the pre-populated message body complies with opt-in requirements for enrollment in text message subscription services such that the mobile device can be automatically subscribed to the subscription list without sending or receiving an additional authorization from the mobile device.”

190. The combination renders this claim obvious. Regarding the limitation “complies with opt-in requirements for enrollment in text message subscription services”, Oliver teaches using a “two-tap” method as I explain above (*e.g.*, in ¶¶30-33, and 74-80), and the specification of the ’074 patent admits that use of the “two-tap” method complied with opt-in requirements for enrollment in text message subscription services. (Ex. 1001, 6:53-57)(*See also* Ex. 1004, pp. 122-123).

191. Furthermore, in my opinion, it would have been obvious to design the combination to comply with opt-in requirements for enrollment in text message subscription services in order to create a commercially viable system which is legally compliant. As of the relevant time, advertising texts could be subject to well-known legal or industry restrictions that required an advertiser to both obtain a user's consent and provide proof of possession of a smartphone prior to sending advertising texts. (see Ex. 1001, 6:53-57). To prove both consent and possession of the phone, it was by 2012 considered a best practice to request, in an advertisement, that a user sign up for text messages by sending a text to the advertiser. (Ex. 1028, pp. 008-009, 0015-0021). Because this method complies with opt-in requirements, the mobile device **can be automatically subscribed to the subscription list without sending or receiving an additional authorization from the mobile device.** As noted above, I further understand that, in patent law, a reference need not state a feature's absence in order to disclose a negative limitation.

CLAIM 22

“22. The method of claim 16, wherein: the pre-populated message body complies with opt-in requirements for enrollment in text message subscription services such that the mobile device can be automatically subscribed to the subscription list without sending or receiving an additional authorization from the mobile device; and the mobile device is subscribed to the subscription list without sending or receiving an opt-in communication other than the URI and the text message.”

192. Claim 22 combines the limitations of claims 19 and 21, and the combination renders obvious claim 22 for the same reasons as set forth under those claims.

CLAIM 23

193. Independent claim 23 is similar to independent claim 16. Where claim 16 is directed to a method, however, claim 23 recites a processor-readable medium storing code configured to cause the processor of at least one server to carry out actions that correspond to the method steps of claim 16. Thus, claim 23 is effectively the same as claim 16, but with the method steps carried out by a server processor.

194. The following mapping shows how the combination renders the claim obvious, with reference to the mapping of claims 1 and 16.

23[a]. At least one non-transitory processor-readable medium storing code configured to be executed by at least one processor of at least one server, the code including instructions configured to cause the at least one server to:

195. As discussed above under claim 1, limitation [1b] Agrawal's system has an advertising system 120 and servers 130, which can be combined into a single server. (Ex. 1005, 4:28-35). The server has a **processor 220 with at least one non-transitory processor-readable medium** (ROM 240 and storage device 250) **storing code configured to be executed by the processor.** (Ex. 1005, 4:39-

57, Fig. 2, 10:18-34). Agrawal also teaches that client servers 130 “may be similarly configured.” (Ex. 1005, 4:39-57)(*see also* Ex. 1005, 4:10-35, 10:4-13).

196. For the reasons described above in the section entitled “Rationale (Motivation) for the Combination”, in my opinion, it would have been obvious to arrange the **code** of the advertising system 120 and client servers 130 to **includ[e] instructions configured to cause the at least one server to perform the functions** recited in each of claim limitations [23b]-[23i].

“[23b] define an integration tag configured to be included in a webpage such that the integration tag executes when a web browser application of a mobile device accesses the webpage;”

197. *See* above, claim 16, limitation [16b]. It further would have been obvious to have the **at least one server** (specifically advertising system 120) to **define the integration tag**. Agrawal’s “code” (integration tag) causes an advertisement query request and detailed user information to be sent to the advertising system 120. (Ex. 1005, 5:64-6:36). The advertising system 120 needs to properly receive and handle the request and user data. (Because of this, the code (executed by the browser) that sends the query request and user data needs to send *it to the correct place, and in the correct form*, for it to be handled properly by the advertising system 120 (*see also* Ex. 1006, 026:18-33)(showing integration tag code addressing specific ad-server locations in a specific format). Because the correct place and correct form are best-known to the administrators of the

advertising system 120, it would have been obvious for the advertising system 120 to define the code. Furthermore, in operation at scale, many thousands of integration tags may need to be defined in a short period of time. Each tag can have unique characteristics for the particular publisher, such as a unique identifier assigned to the publisher and particular business rules that are executed. (Ex. 1006, 023:1-5). In order to generate a unique identifier and business rules with each integration tag during large-scale operations, it would have been most obvious to configure the advertising system 120 server to define and generate the integration tags. Finally, it was well-known in the relevant timeframe (2017) for content publishers to receive ad tag code from an advertising system, for the purpose of integrating the code into web sites, such that the code will transmit information and request advertising when a consumer's browser downloads and interprets the web site. In my opinion, implementation of this well-known technique into a well-known ad system (Google's) as taught by Agrawal would have been no more than the combination of familiar limitations according to known methods, yielding predictable results, and thus obvious.

“[23c] receive, from the mobile device and in response to the integration tag executing at the mobile device, (1) a mobile device identifier that identifies the mobile device, and (2) a request to retrieve an identifier associated with at least one of a product or a service presented via the webpage;”

198. See above, claim 16, limitation [16c].

“[23d] send, to the mobile device and in response to receiving the mobile device identifier and the request to retrieve the identifier, a first signal configured to cause a frame to be displayed with the webpage via the web browser application, the frame including a promotional message associated with joining a subscription list;”

199. See above, claim 16, limitation [16d].

“[23e] receive, from the mobile device an indication of a first user interaction with the promotional message;”

200. See above, claim 16, limitation [16e].

“[23f] send a uniform resource identifier (URI) to the mobile device that deeplinks to and causes the mobile device to automatically (1) switch from rendering the web browser application to rendering a text messaging application and (2) generate a text message with a pre-populated phone number and a pre-populated message body containing a request to join the subscription list in response to at least one of (1) the integration tag executing or (2) the indication of the first user interaction;”

201. See above, claim 16, limitation [16f].

“[23g] receive the text message including the pre-populated message body to the pre-populated phone number in response to a second user interaction with the text messaging application;”

202. See above, claim 16, limitation [16g].

“[23h] subscribe the mobile device to the subscription list based on receiving the text message; and”

203. See above, claim 16, limitation [16h].

“[23i] deliver an installment of a subscription associated with the subscription list without receiving an opt-in communication after receiving the text message containing the pre-populated message body.”

204. See above, claim 16, limitation [16i].

CLAIMS 24-29

205. Claims 24-29 are similar to 17-22 except claims 24-29 are directed to the steps carried out via a non-transitory processor-readable medium storing code configured to be executed by at least one processor of at least one server whereas claims 17-22 are in method form. In my opinion, claims 24-29 are obvious for the same reasons as claims 17-22, respectively.

CLAIM 30

“30. The at least one server comprising the non-transitory processor-readable medium of claim 23 and the at least one processor.”

206. In my opinion, the combination renders obvious claim 30 is obvious for same reasons as set forth under claim 23. The processor-readable medium is part of the server as discussed above under claim 23, limitation 23[a].

Ground 2. Claims 1-30 were obvious as in Ground 1, in further view of Molinet.

207. In my opinion, claims 1-30 were obvious as in Ground 1, in further view of U.S. Pat. App. Pub. 2016/0142858 A1 (“**Molinet**”)(Ex. 1009).

208. I understand that Molinet was not of record during the prosecution of the application leading to the '074 patent.

A. Overview of the Ground

209. This Ground challenges the claims at-issue under Ground 1, further in view of Molinet. As explained below, Molinet provides certain additional teachings regarding “mobile device identifiers” and customization of deeplinks that is relevant to the independent claims and certain dependent claims. The claims for which Molinet provides additional relevant information, namely independent claims 1, 8, 16, and 23 and dependent claims 4-5 and 11-12 will be specifically addressed under the Claim Mapping section, together with relevant motivation for the combination.

B. Overview of Molinet

210. Molinet teaches a system that can use mobile-app deeplinking with “contextual deeplinks”. (Ex. 1009, ¶¶0016-0018). “Contextual deeplinks” are app deeplinks that are customized for the particular operating environment of a user’s mobile device. (Ex. 1009, ¶0037). This customization is desirable, according to Molinet, because different operating environments may not use or allow all deeplinking protocols. (Ex. 1009, ¶0005). Molinet further incorporates (Ex. 1009, ¶0001) its provisional application, U.S. Prov. App. 62/079,512 (Ex. 1010). For example, the '512 provisional teaches the use of standard URIs for mobile-app

deeplinking used by the iOS and Android operating systems. (Ex. 1010, pp. 021, 015).

C. Analogous Art

211. In my opinion, Molinet is analogous art because it is in the same field as the '074 patent and is reasonably related to the problems facing the named inventors. (Ex. 1009, Abstract, ¶¶0002-0005, 0025-0027).

D. Reasonable Expectation of Success

212. In my opinion, a PHOSITA would have had a reasonable expectation of success in using the teachings of Molinet in the manner described herein for the reasons described in Ground 1.

E. Claim Mapping

CLAIMS 4 AND 11

“4. The method of claim 1, wherein the mobile device identifier is based on a user agent field associated with the web browser application.”

“11. The non-transitory processor-readable medium of claim 8, wherein the mobile device identifier is based on a user agent field associated with the web browser application.”

213. See above, Ground 1, claims 4 and 11. As set forth under claim 4, Agrawal’s user data (**the mobile device identifier**) can identify the operating system and browser type executing on the mobile device. (Ex. 1005, 5:45-6:46, 7:19-28, 8:20-33).

214. Molinet teaches that **user agent** information can be collected as part of the user data. (Ex. 1009, ¶0085)(Ex. 1010, p. 022). As Molinet's incorporated provisional explains, the user agent information sent by a browser refers to the "User-Agent" field of an HTTP request, which included an identification of the operating system, operating system version, and name and version of the browser itself. (Ex. 1010, p. 022)(Ex. 1009, ¶0065). Accordingly, Molinet teaches that the **mobile device identifier is based on a user agent field associated with the web browser application.** (*See also* Ex. 1001, 9:45-10:65).

215. In my opinion, it would have been obvious to collect user agent information as "user data" sent together with the advertisement query request of Agrawal (*see above*, Ground 1, claim 1, limitation [1d]), both because the user agent data is sent with the HTTP header in the ad request (Ex. 1010, p. 022), and because the user agent data could be accessed using JavaScript executed by a browser by using, for example, the Navigator object's userAgent field. Ex. 1021, pp. 364-365, 992)(Ex. 1016, pp. 324-338, 584-590). It would have been obvious to do so, for example to identify the operating system of the mobile device (for example, to configure a URI, as explained under Grounds 1 and 2, claim 5), because the user agent states the identity of the operating system directly. (*Id.*).

CLAIMS 5 AND 12

"5. The method of claim 1, wherein: the mobile device identifier identifies a first operating system executing on the mobile device;

and the URI is compatible with the first operating system and not a second operating system different from the first operating system.”

“12. The non-transitory processor-readable medium of claim 8, wherein: the mobile device identifier identifies a first operating system executing on the mobile device; and the URI is compatible with the first operating system and not a second operating system different from the first operating system.”

216. See above, Ground 1, claims 5 and 12. Molinet teaches that the user data (**mobile device identifier**) **identifies a first operating system executing on the mobile device.** (Ex. 1009, ¶0065)(“These characteristics may include network address (*e.g.*, IP address), operating system version, operating system type, device model information, and screen size.”)(*see also* Ex. 1009, ¶¶0017, 0025, 0037, 0054, 0069, 0085, 0105).

In my opinion, Molinet further renders obvious that **the URI is compatible with the first operating system and not a second operating system different from the first operating system.** Specifically, Molinet specifically teaches that URIs can be customized based on the user data (the “fingerprint”, ¶0017), including the operating system, such that the URI is operable on the particular mobile device. (Ex. 1009, ¶¶0037-0045). Molinet explains:

“The server link module 115 also determines contextual information for the client device 120 based on the characteristics of the client device 120. Based on the contextual information, the server link module 115 transmits a response back to the client device 120 such that the response is operable on the client device 120.”

(Ex. 1009, ¶¶0037)(*See also* ¶¶0045, 0017-0018)(Ex. 1010, pp. 015, 021-022).

Molinet thus teaches that the customized URI is compatible with the mobile device's operating system and is not compatible with a different operating system (*e.g.*, iOS v. Android).

XI. SECONDARY CONSIDERATIONS

217. I am of the opinion that secondary considerations would not weigh against obviousness in this case. I am aware that in a complaint for patent infringement, Patent Owner alleged (Ex. 1022, ¶51) that the patent has “garnered industry praise”, citing three articles (submitted as Exs. 1023-1025). Two of the articles, however, seem to simply quote or characterize statements of the Patent Owner itself (Exs. 1024 and 1025), which a PHOSITA would not find compelling as the honest reaction of a neutral observer to a patent. A third article calls “two-tap” technology “novel”. However, as I explain above in ¶¶30-38, two-tap functionality was well-understood and known to have benefits. (Ex. 1011, ¶¶0007, 0009, 0024, 0027-0029)(Ex. 1007, Abstract, 2:28-31, 10:1-5, 11:19-21, 9:48-55, 13:12-18, 1:53-62)(Ex. 1013, ¶¶0039-0042)(Ex. 1017, ¶¶0026, 0069, 0088).

218. I understand that in a co-pending district court litigation (where the Patent Owner was responding to an argument that the '074 patent was not patentable subject matter), the Patent Owner also alleged copying (Ex. 1014, p. 3),

but provided no details, citing only to an irrelevant paragraph in its complaint. In my view, this does not make a good case for copying.

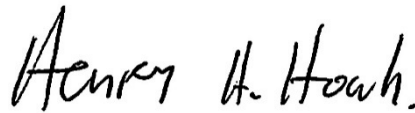
219. Overall, it is my opinion that there is not significant evidence of secondary considerations, much less evidence that would change my opinion about the strong evidence of obviousness over the prior art as discussed above.

XII. OATH

220. This declaration and my opinions herein are made to the best of my knowledge and understanding, and based on the material available to me, at the time of signing this declaration. I declare under penalty of perjury under the laws of the United States of America that all statements made of my own knowledge are true and that all statements made on information and belief are believed to be true. I understand that willful false statements and the like are punishable by fine or imprisonment, or both (18 U.S.C. 1001) and may jeopardize the validity of the application or any patent issuing thereon.

October 31, 2023

Date

A handwritten signature in black ink that reads "Henry H. Houh". The signature is written in a cursive, slightly slanted style.

Dr. Henry Houh