

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. 11,416,897

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 '897 PATENT.....	42
VII. RELEVANT TIMEFRAME FOR DETERMINING OBVIOUSNESS	44
VIII. LEVEL OF ORDINARY SKILL IN THE ART	44
IX. CLAIM CONSTRUCTION	45

A.	Claim 15—“means for displaying....”	45
B.	Claim 15—“means for causing....”	46
X.	DETAILED EXPLANATION OF THE REASONS FOR UNPATENTABILITY	47
Ground 1.	Claims 15-19 and 21-30 were obvious over Oliver in view of Agrawal and Molinet.....	47
A.	Overview of the Ground.....	47
1.	Overview of Oliver	48
2.	Overview of Agrawal.....	51
3.	Overview of Molinet.....	54
B.	Rationale (Motivation) Supporting Obviousness.....	56
C.	<i>Graham</i> Factors	61
D.	Reasonable Expectation of Success	62
E.	Analogous Art	63
F.	Claim Mapping.....	63
Ground 2.	Claims 15-30 were obvious as in Ground 1, in further view of <i>Ad Serving Technology</i>	109
A.	Overview of the Ground.....	110
B.	Analogous Art	110
C.	Claim Mapping.....	111
XI.	SECONDARY CONSIDERATIONS	112

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 US 11,416,897 (the ’897 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. 11,416,897 (“ the ’897 patent ”).
1004	File History of U.S. App. Ser. No. 17/569,265 (issued as the ’897 patent).
1005	U.S. Pat. No. 8,015,615 (“ Agrawal ”).
1006	U.S. Pat. App. Ser. No. 10/375,900 (“ the ’900 application ”).
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 (“ Molinet ”).
1010	U.S. Prov. App. Ser. No. 62/079,512 (“ The Molinet ’512 Provisional ”).
1011	U.S. Pat. App. Pub. 2015/0178784 (“ 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.
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 ").

II. SUMMARY OF OPINIONS

4. It is my opinion that:

- Claims 15-19 and 21-30 were obvious over Oliver in view of Agrawal and Molinet.
- Claims 15-30 were obvious as in Ground 1, in further view of *Ad Serving Technology*.

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. 11,416,897

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 ¶69, 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 ¶70 below.

17. The relevant time frame for obviousness, the “time the invention was made”, is discussed in ¶69, 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 '897 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 '897 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 '897 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, 7:14-24).

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 2021) 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:

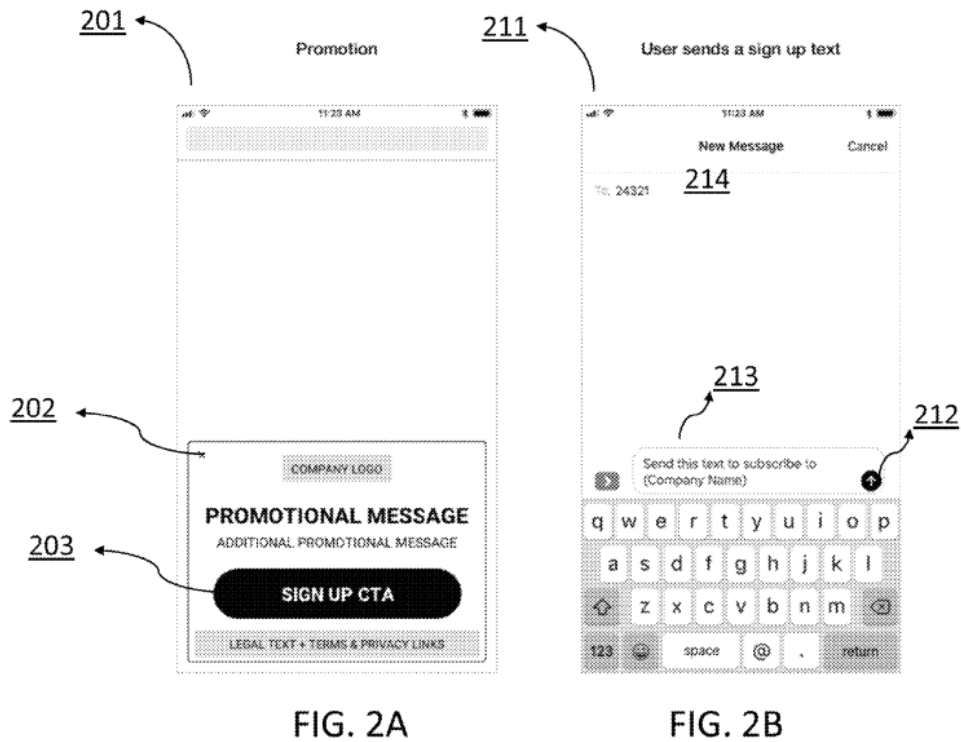
Upmobile Ski Alerts!
Send us the resort name, we'll send you the snow conditions. Txt
'Mammoth' to 12345 to receive ongoing alerts for Mammoth resort.
Get 10 msgs/month.
Text HELP for help.
To stop text STOP.
Msg&Data Rates May Apply.
T&Cs avail at www.mammoth.com/mobile.

26. Thus, in 2012, five years before the filing of the first application leading to the '897 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 '897 patent.

27. The '897 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, 6:60-7:2). Clicking the link from a mobile device causes the device’s text-messaging app to open. (Ex. 1001, 6:60-7:2). Upon opening, the text-messaging app presents the user with a pre-filled text message. (Ex. 1001, 7:7-11). 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, 7:10-22). 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 '897 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, 7:24-59). 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, 6:60-7:66). Because only two user actions are required, the '897 patent refers to this approach as a “two-tap” approach. (Ex. 1001, 3:1-3, 7:19-23).

29. I understand that the Patent Owner considered this “two-tap” method to be the point of novelty for the ’897 patent. (Ex. 1014, p. 2). The ’897 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, 1:30-46). 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:

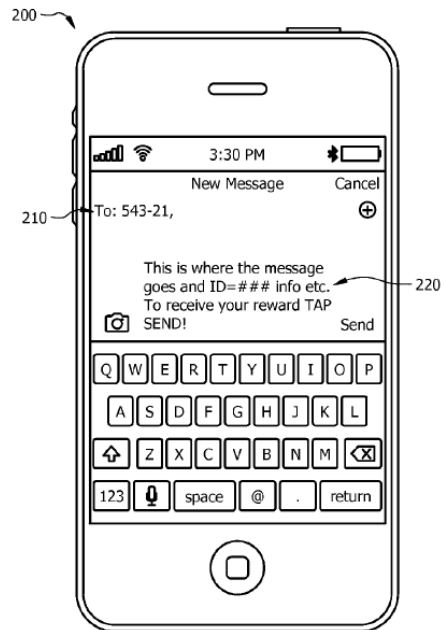
Two-tap example: U.S. Pat. App. Pub. 2015/0178784 to Oliver (Ex. 1011)

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).



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

“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 ’897 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:

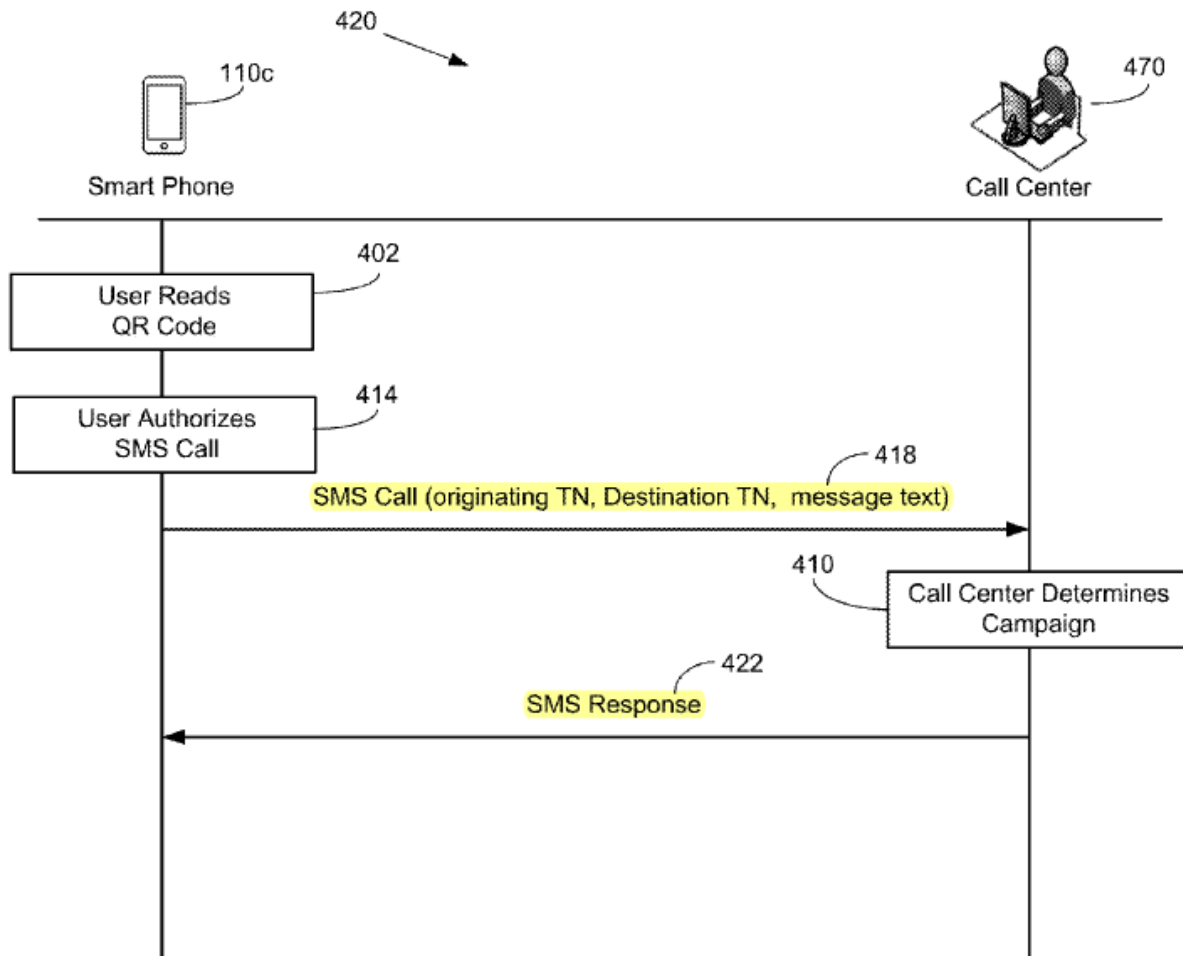


FIG. 4B

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

36. Just like the '897 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 '897 patent also pair the known two-tap business method to standard Internet advertising technology. The standard advertising technology used by the '897 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:

```

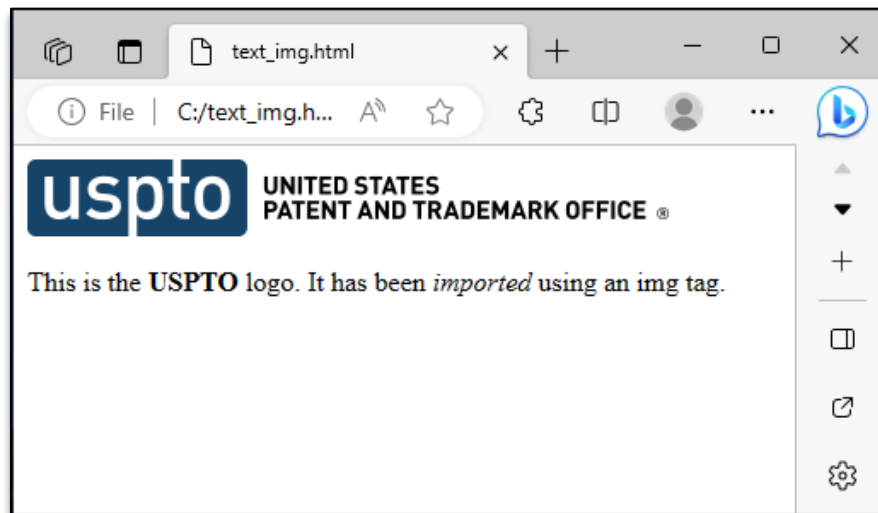
```

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

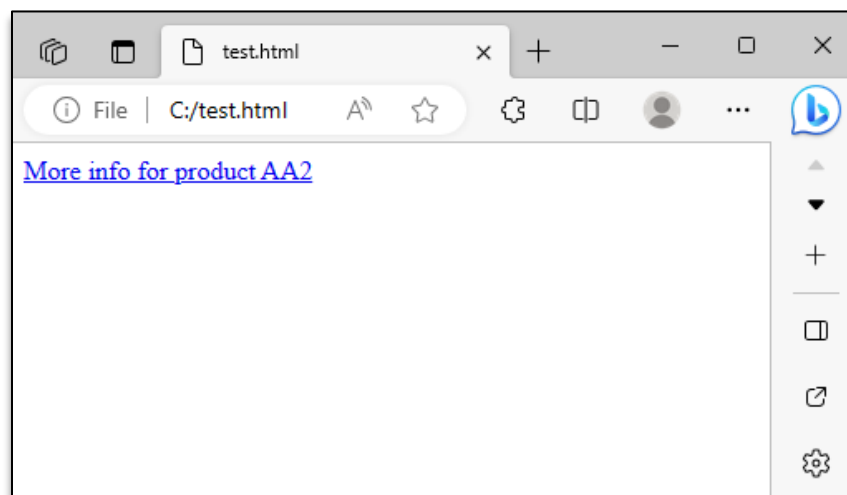
document, one can generally load the page in a browser, and use a “view source” function (for example, by right clicking on the browser window).

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. 1006):

```
<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:

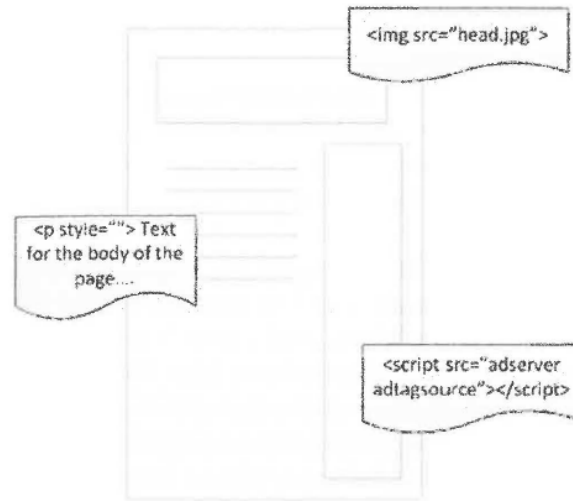


Figure 1.5: Behind the scenes a standard web page is made up of a series of HTML tags. An Ad tag is no different, which can sit on any web page to display an ad to the user. In this instance (bottom right) the HTML tag contains a reference to JavaScript.

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.

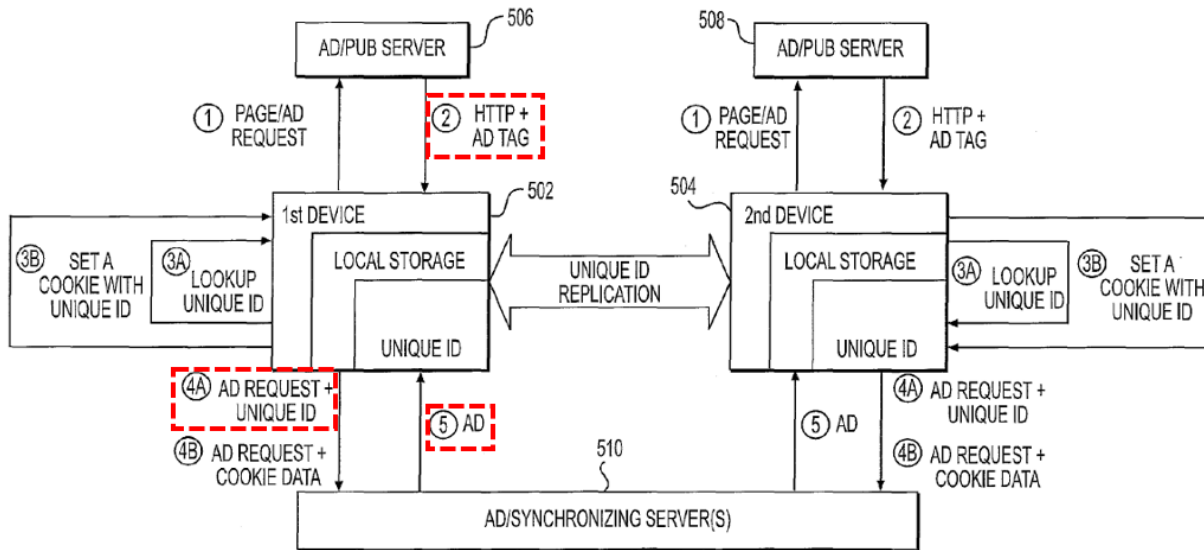


FIG. 5

(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 '897 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 '897 PATENT

67. The petition for *inter partes* review challenges claims 15-30 of the '897 patent. Independent claim 15 can be divided into roughly three parts, corresponding to (1) *standard ad-network technology* that uses a script-based ad tag, (2) *standard deeplinking technology* that provides a way to link to a text-

messaging application, and (3) the “two-tap” business method. Claim 15 is shown in the table below, with added numbers (15[a] [15b], etc.):

Claim element	Category
“15[a]. A non-transitory processor-readable medium storing code configured to be executed by a processor, the code including instructions configured to cause the processor to:”	Preamble
“[15b] send, from at least one server and to a mobile device requesting and/or a webpage, means for displaying an invitation to subscribe to a text messaging subscription service via a first application;”	Ad-network
“[15c] send, from the at least one server and to the mobile device, means for causing (1) the mobile device to transition from the first application to a second application different from the first application in response to the mobile device detecting a first interaction with the invitation, the second application being a messaging application, and”	Deeplinking
“[15d] (2) the second application to automatically populate a custom text message based on the user data;”	Deeplinking / Business method
“[15e] receive, from the mobile device, the custom text message in response to the mobile device detecting a second interaction with the messaging application; and”	Business method
“[15f] deliver an installment of the text messaging subscription service without receiving an opt-in communication after receiving the custom text message.”	Business method

68. In claim 15, a “server” (*e.g.*, the central server of an ad network) sends a means for displaying (*e.g.*, an ad tag) to mobile device. The means for displaying is configured to display a promotion (*e.g.*, an ad). This is standard ad-network technology. When a user clicks on the ad, a server sends a 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. RELEVANT TIMEFRAME FOR DETERMINING OBVIOUSNESS

69. I understand that obviousness must be evaluated at the time the alleged invention was made. From the second page of the '897 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 report 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.

VIII. LEVEL OF ORDINARY SKILL IN THE ART

70. 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.

IX. CLAIM CONSTRUCTION

71. I understand that it is sometimes beneficial to explain (“construe”) the meanings of the terms of claims in a patent. My understanding of the legal rules that apply when construing claim terms in an inter partes review proceeding is provided above in ¶10.

72. Claim 15 has two terms that are written in so-called “mean-plus-function” format. My understanding is that, when such terms are present, a particular law (35 U.S.C. §112(f)) governs their construction, and in particular, that such claim terms “shall be construed to cover the corresponding structure, material, or acts described in the specification and equivalents thereof.”

A. Claim 15—“means for displaying....”

73. Claim 15 recites a “means for displaying an invitation to subscribe to a text messaging subscription service via a first application”. Here, the “function” part of the means-plus-function term is “displaying an invitation to subscribe to a text messaging subscription service via a first application”. I understand that the “means” part of the term must be construed to cover “the corresponding structure, material, or acts described in the specification and equivalents thereof”.

74. It is my opinion that the corresponding structure described in the specification is **an integration tag configured to cause performance of the claimed functions**. Support for the structure corresponding to the recited function is described in the '897 patent at Ex. 1001, claim 16, 1:57-64, 5:29-36, 11:54-66, 12:56-65, 14:61-15:13, 17:9-33.

B. Claim 15—“means for causing....”

75. Claim 15 also recites “means for causing (1) the mobile device to transition from the first application to a second application different from the first application in response to the mobile device detecting a first interaction with the invitation, the second application being a messaging application, and the second application to automatically populate a custom text message based on the user data.” Here, the “function” part of the means-plus-function term is “causing (1) the mobile device to transition from the first application to a second application different from the first application in response to the mobile device detecting a first interaction with the invitation, the second application being a messaging application, and the second application to automatically populate a custom text message based on the user data”. Again, I understand that the “means” part of the term must be construed to cover “the corresponding structure, material, or acts described in the specification and equivalents thereof”.

76. It is my opinion that the corresponding structure described in the specification is **a URI deeplinking to a messaging application and configured to cause performance of the claimed functions**. Support for the structure corresponding to the recited function is described in the '897 patent at Ex. 1001, claim 16, 2:3-6, 2:14-19. 3:13-15, 4:20-26, 3:59-62, 10:22-47, 11:37-41, 19:24-28, 19:44-49, 22:1-18, 22:36-53, 22:65-67, 23:22-28, 23:51-59.

X. DETAILED EXPLANATION OF THE REASONS FOR UNPATENTABILITY

Ground 1. Claims 15-19 and 21-30 were obvious over Oliver in view of Agrawal and Molinet.

77. It is my opinion that claims 15-19 and 21-30 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. 2016/0142858 A1 ("**Molinet**") (Ex. 1009).

A. Overview of the Ground

78. 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. This ground posits that it would have been obvious to implement Oliver's business method using well-known ad-serving technology, as described in two publications: Agrawal and Molinet.

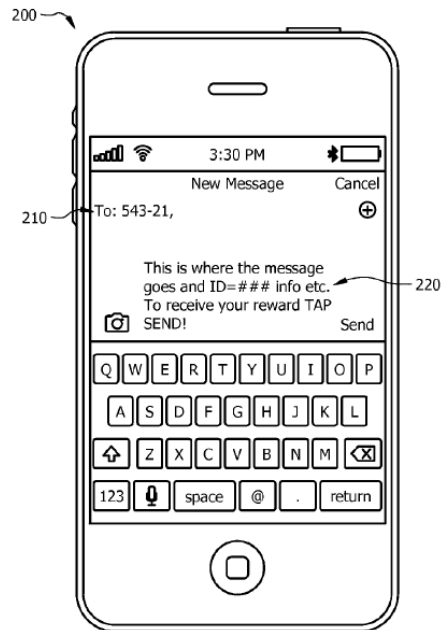
1. Overview of Oliver

79. As I described above, **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 link information and be activated by a user’s mobile device. (Ex. 1011, ¶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).

80. 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).

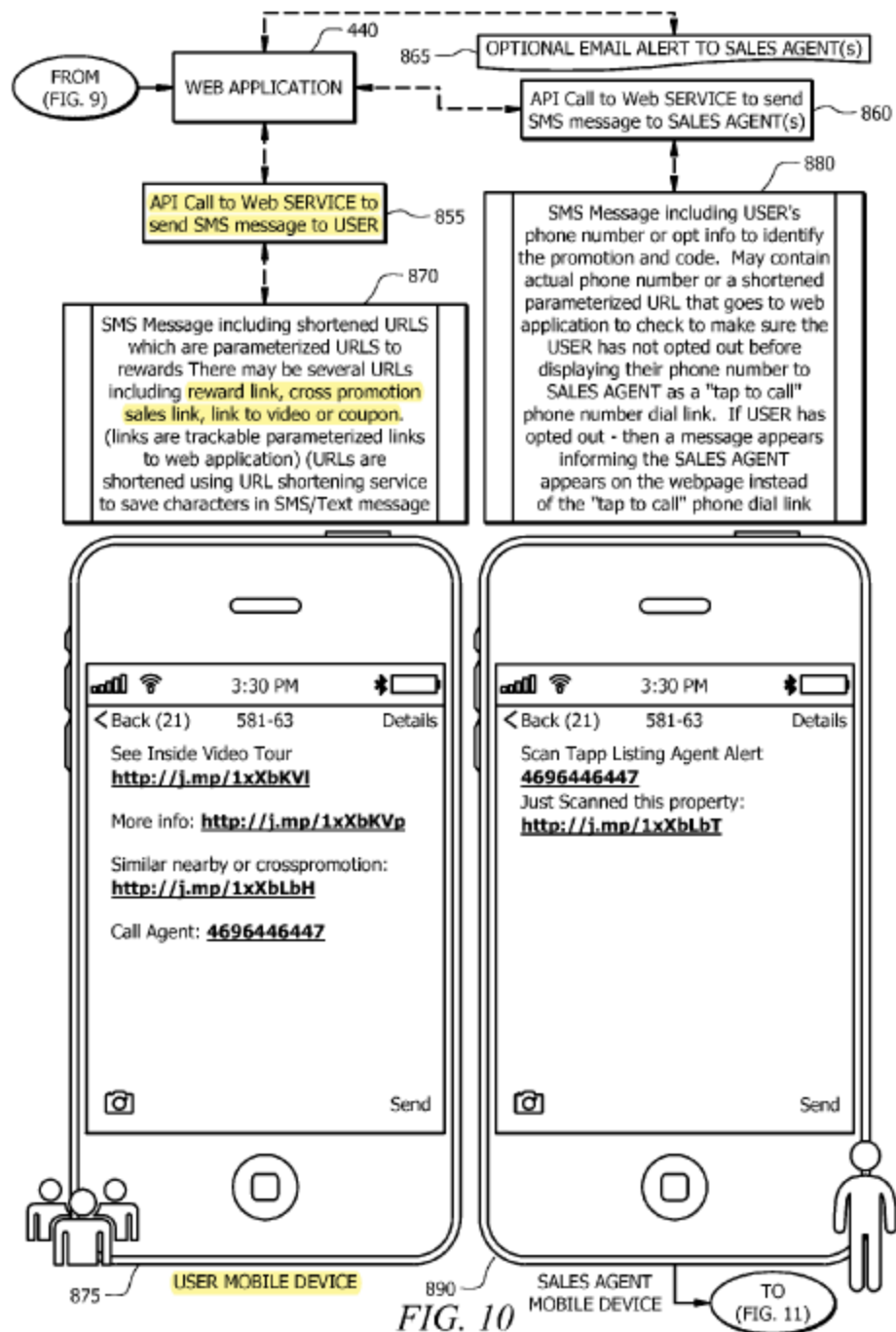


81. Oliver explains—just like the '897 patent—that limiting the user actions to two improves the user response:

“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).

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



(Ex. 1011, Fig. 10, ¶¶0051-0053).

83. While Oliver teaches that the user interacts with an advertisement, 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. This standard technology, however, was disclosed in **Agrawal**.

2. Overview of Agrawal

84. **Agrawal** teaches an example of a known ad-network system. Ad-network systems were described above in the Introduction, and generally have a *central ad-network server* to coordinate activities, *advertisers* who wish to place ads, *publishers* who have servers that serve 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:

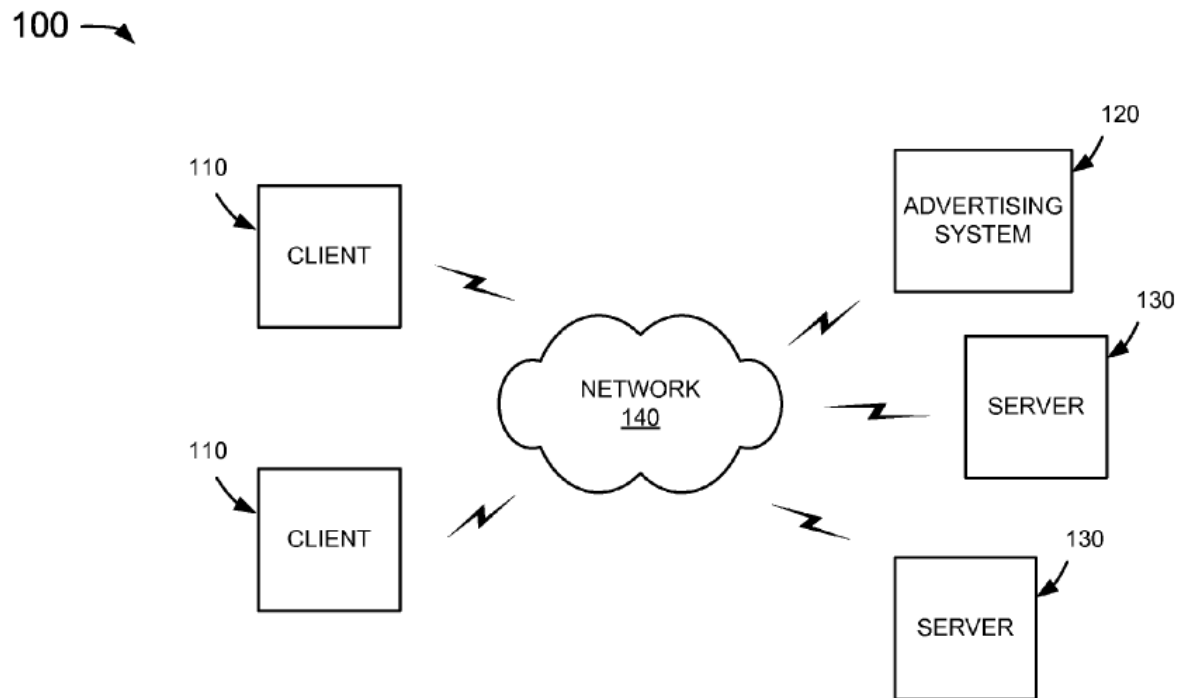


FIG. 1

(Ex. 1005, Fig. 1, 3:62-35). In Fig. 1, the central ad-network server is the “advertising system 120”. (Ex. 1005, 1:62-65, 10:5-6). The users are the “client” devices 110. (Ex. 1005, 4:10-16). The various publishers of webpages are represented by servers 130. (Ex. 1005, 4:18-20). Advertisers are not shown separately in Fig. 1.

85. In Agrawal’s system, a user’s device will contact a publisher server 130 to download a webpage. (Ex. 1005, 5:29-32, 3:46-58). The webpage has code (*e.g.*, JavaScript) that is executed when the user’s device loads the webpage. (Ex. 1005, 5:34-39, 3:52-55). Upon execution of the code (when the user’s browser loads the webpage), the user’s 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 for display, 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 the representation of a user’s browser screen in Fig. 4 (links 410) of Agrawal, reproduced here:

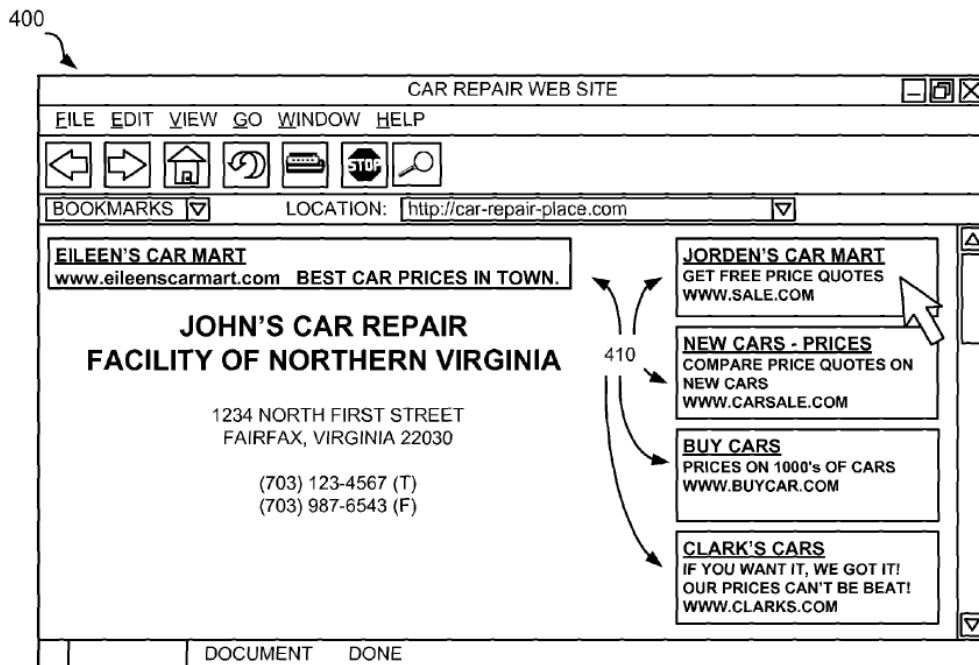


FIG. 4

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

86. Agrawal also has an authentication mechanism that is relevant to the '897 patent dependent claims. Authentication is needed in Agrawal because, if the user clicks an ad link, the ad will be deemed successful and someone will get paid. (Ex. 1005, 1:27-2:3). This potential for payment, however, also creates an incentive for fraud: a malicious user might use bots to generate clicks that are not from a human user. (*Id.*). To filter out such spurious clicks, Agrawal also collects user data *a second time* after an ad link is clicked. (Ex. 1005, 6:47-7:62). By comparing the *second set* of user data to the *first set* of user data (collected when the ad tag executed), Agrawal can authenticate the user, and prevent some fraud. (Ex. 1005, 7:45-8:12). This latter process is shown in Fig. 3 (reproduced below),

which shows a method carried out by the advertising system 120 (Ex. 1005, 5:3-8, 10:4-7):

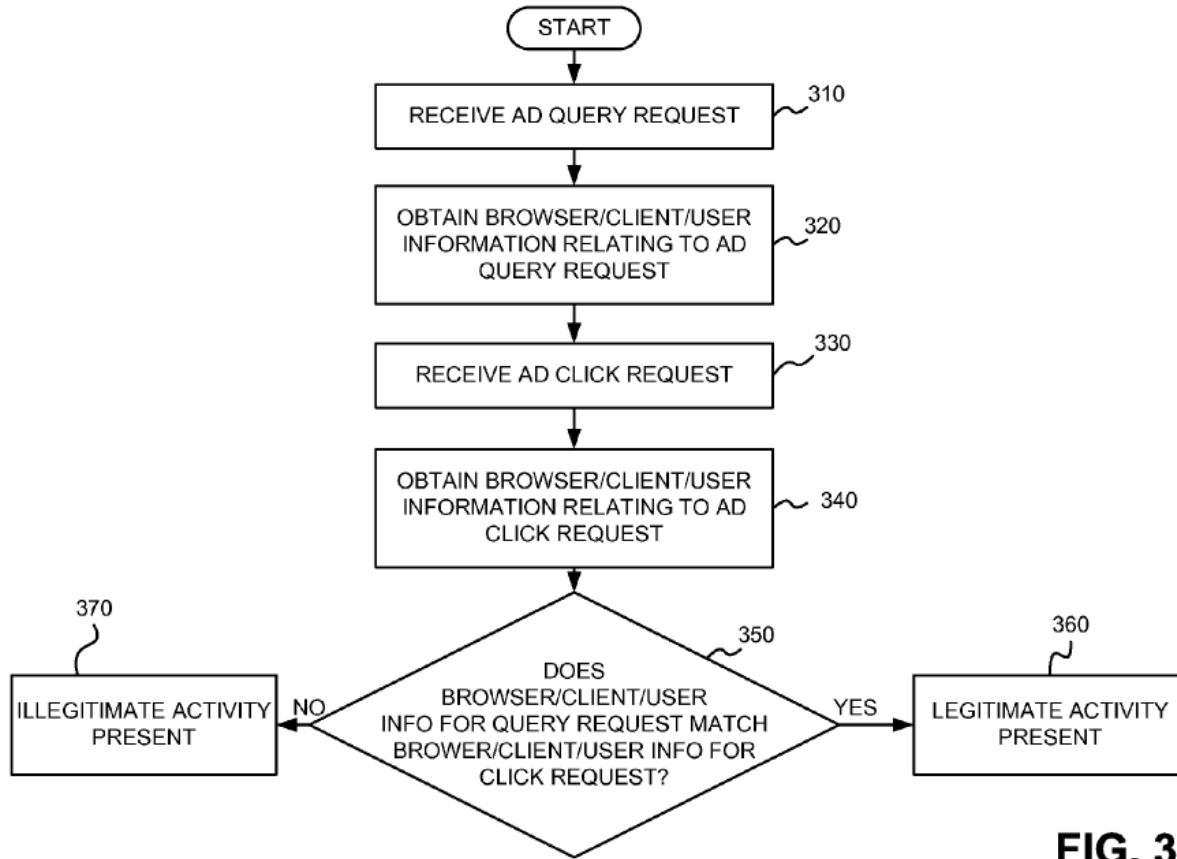


FIG. 3

(Ex. 1005, Fig. 3).

87. Agrawal does not teach specifically that its advertising links can be “deeplinks” that would open mobile apps, but this functionality was well-known in the relevant timeframe and provided by another patent publication, **Molinet**.

3. Overview of Molinet

88. **Molinet** teaches a system for providing working deeplinks for mobile apps. (Ex. 1009, ¶0003). Molinet explains that in an “application

ecosystem” (like a modern smartphone), deeplinks might not be standardized. (Ex. 1009, ¶0005). This created a situation where applications could not easily link to one another, and had particular difficulty linking to specific locations, pages, screens *within* applications (i.e. deeplinking). (Ex. 1009, ¶0005). The desired deeplinking was not easily accomplished in the prior art to Molinet because, as Molinet explains, different mobile devices might use different deeplinking protocols. (Ex. 1009, ¶0005). Molinet explains:

“[A]s the number of applications provided by these ecosystems increase, the environment on a user’s client device becomes less cohesive as unlike with hyperlinks on the Internet, **each application on the client device is unable to easily link to a section of another application** on the client device through the application installation process.....This creates a situation where users are forced to operate each application in semi-isolation. Hence, **what was lacking** was a method of easily directing users to sections within applications and passing data to applications, **regardless of the type of client device the user and whether the application is installed on the client device.**”

(Ex. 1009, ¶0005)(Emphasis added).

89. To address this situation, Molinet provided a system with a central server that facilitates deeplinking. (Ex. 1009, ¶¶0101-0107, Fig. 5, ¶¶0079-0082). To do this, Molinet’s system takes a desired target within an app, and produces a

URL to represent that target location. (Ex. 1009, ¶¶0072, 0025-0026, 0033). The URL could be used in a normal, clickable Web link. (Ex. 1009, ¶¶0026, 0033, 0072, 0088, 0102). When such a link was clicked, the link caused a request to be sent to Molinet's *contextual deeplinking server*. (Ex. 1009, ¶¶0101-0107, Fig. 5, ¶¶0074-0082). The contextual deeplinking server, in turn, accessed information about the type of device the user was using. (Ex. 1009, ¶¶0103-0106, Fig. 5, ¶¶0074-0079). Using this information, Molinet's server formulated a contextually-appropriate deeplink URL (*i.e.* a deeplink URL that works with the operating system and software of the device), and sent that URL back to the mobile device, which would then activate the deeplink to cause the device to transition to a particular state within a different application. (Ex. 1009, ¶¶0082, 0107, 0033, 0036, 0016-0018, Fig. 5)(Ex. 1010, pp. 008-011). The interdiction of the server in the linking process thus ensures that the link will work on any given device. (Ex. 1009, ¶0119).

B. Rationale (Motivation) Supporting Obviousness

90. 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 Molinet's techniques to deeplink to mobile applications. I incorporate my opinions above, particularly relating to the Background of the technology prior to May 2017.

91. **First**, it was obvious to implement Oliver's method using the system of Agrawal and Molinet. Oliver's method would have appealed to businesses that sell goods and services and that seek new potential customers. In particular, it would have been attractive to have a streamlined method for acquiring the contact information, and for acquiring consent to receive marketing communications and proof of possession of the mobile device, from people interested in the business's offerings. (Ex. 1011, ¶¶0004-0006, 0009).

92. A PHOSITA would have found the combination of Oliver with the system of Agrawal/Molinet obvious, where the functionality of Oliver's "web service" (Ex. 1011, ¶¶0033, 0038) is carried out by Agrawal's advertising system 120. By "web service", a PHOSITA understands software running on a server that can interact with other devices on the Internet using Web protocols (*e.g.*, HTTP). Such a Web service would have been easy to place on Agrawal's advertising system 120, which already uses Web protocols. (Ex. 1005, 6:14-17, 6:41-43, 7:13-17, 7:40-42). 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 teaching a two-step sequence for acquiring new leads, in which a potential customer first interacts with a “passive entity”, and second, clicks send on a pre-filled text message. Oliver states that the “passive entity” can essentially be anything with which a mobile device user 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 the Internet also has electronic visual media in the form of advertisements, and advertisements have links that can link to URIs, including (per Molinet) deeplinking into mobile apps. A PHOSITA would have found it advantageous to use Internet ad systems like Agrawal/Molinet in order to increase the reach of Oliver’s marketing efforts, and to target the Oliver’s advertisements to websites likely to attract interested customers, (Ex. 1005, 5:45-63). 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).

93. **Second**, it would have been obvious to extend Agrawal’s system with Molinet’s capabilities. Specifically, a PHOSITA with Agrawal’s system would

have desired the mobile app deeplinking capability provided by Molinet. The PHOSITA would have understood that mobile app deeplinking protocols could differ depending on the mobile operating environment (Ex. 1009, ¶0005), and would have further desired Molinet's ability to "create a contextual deep link that does not cause an error under any context", and to allow apps to use and manipulate links easily independent of context. (Ex. 1009, ¶¶0119-0121)(Ex. 1010, p. 001). A PHOSITA would have desired to allow users to click links in webpages to link to mobile apps (which Molinet expressly teaches, Ex. 1009, ¶¶0026, 0033), such that advertisements for a particular advertiser's products could link directly to that advertiser's app, or to a communication app for communicating directly with the advertiser, without fear of the link generating an error. (Ex. 1009, ¶0119).

94. Furthermore, both Agrawal and Molinet are based on Web technologies, and use communications with a central server to dynamically alter a user's web experience. (Ex. 1005, 5:29-63, 7:54-58)(Ex. 1009, ¶¶0072-0082, 0101-0107). Agrawal uses code on a webpage to collect user information and issue an ad request to a central server, which then returns an advertisement that is inserted into the webpage. (Ex. 1005, 5:29-63, 7:54-58). Similarly, Molinet uses code on a webpage to collect user information and, when a link is clicked, issue a deeplinking request to a central server, which then returns a contextually-

appropriate deeplink URL. (Ex. 1009, ¶¶0016-0018, 0025-0026, 0033, 0072-0082, 0101-0107). Both of these references thus teach a user loading a webpage, code on the webpage sending information to a server, and the server sending back information (*e.g.*, an ad or a URI) that changes how the webpage is experienced by the user. Both references also express flexibility in how their systems are configured, and in particular where and how algorithmic steps are carried out. (Ex. 1005 4:3-9, 4:28-35, 4:28-57, 10:14-42)(Ex. 1009, ¶¶0122-0131, 0019, 0014, 0097, 0108). Given the technical compatibility and similarity between the two references and their own expressed flexibility, it would have been obvious to a PHOSITA to practice Molinet's techniques in Agrawal's system, for example, by performing the functionality of Molinet's servers at Agrawal's advertising system 120, and the functionality of Molinet's user devices at Agrawal's mobile devices

95. In particular, it would have been obvious to carry out the server-based software techniques of Oliver and Molinet at Agrawal's Advertising System 120 and/or servers 130. 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 Molinet, moreover, provides functionality with a server (like Agrawal's advertising system), in the form of Molinet's contextual deeplinking server and Oliver's Web service 430. (Ex. 1009, Fig. 1, Abstract, ¶¶0016-0018)((Ex. 1011, ¶0031). Both Molinet and Oliver also express flexibility in the arrangement of

their systems. (Ex. 1009, ¶¶0122, 0127)(Ex. 1011, ¶¶0101, 0031, 0025, 0033, 0036). It would have been obvious to combine the functionalities of these references into a single “server” as that term is understood in the ’897 patent (Ex. 1001, 4:13-42. 5:14-47) in order to obtain the benefits of each system as described above. The use of server systems was a well-established discipline by 2017, and it was well-known to scale such systems according to demand, and to combine various software functionalities capable of implementing the functionality described in the Agrawal, Molinet, and Oliver.

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

C. *Graham* Factors

97. 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 ¶70.

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

99. The **differences between the prior art and the claims** are discussed in the section entitled “Rationale (Motivation) Supporting Obviousness”, above, and in the claim mapping, below.

100. **Secondary considerations** are discussed in ¶¶178-180, below.

D. Reasonable Expectation of Success

101. 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 '897 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 Molinet's software techniques in Agrawal's system, and then to carry out Oliver's method, combining the functions of the central servers in each reference. When I use the term “central server”, I am referencing the somewhat more general concept of a central coordinating server, such as the advertising system 120 in Agrawal, Molinet's contextual deeplinking server, and Oliver's web service, which serve the purpose of coordinating activities with respect to users and other server functions, while recognizing that the functions of the central server 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. A PHOSITA would have been able to make any necessary modifications to implement the Ground, and in particular would have been able to implement Molinet's software techniques in Agrawal's system, and then to carry out Oliver's method.

E. Analogous Art

102. Oliver, Agrawal, Molinet are analogous art because they are in the same field as the '897 patent (Internet and mobile device technology). (Ex. 1001, 1:26-29)(Ex. 1005, 1:10-50)(Ex. 1009, Abstract, ¶¶0003-0005). Furthermore, the methods of Oliver, Molinet and Agrawal 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, 1:42-46).

F. Claim Mapping

103. 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 elements, 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 15

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

104. It is my opinion that the combination renders obvious claim 15 of the '897 patent. The preamble is shown in the prior art by Agrawal's advertising system 120, shown in Fig. 1 of Agrawal, reproduced below with an added red-dashed box:

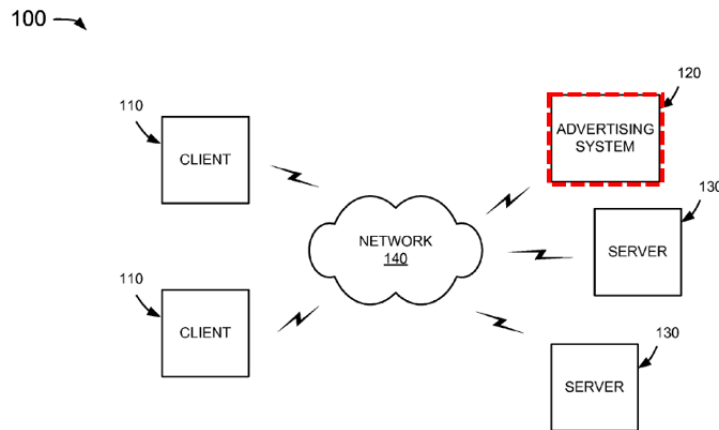


FIG. 1

(Ex. 1005, Fig. 1).

105. The advertising system 120 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).

106. For the reasons described above in ¶¶90-96, it would have been obvious to arrange the **code** of the advertising system 120 to **include instructions configured to cause the system to** perform the functions recited in each of the claim limitations below.

“[15b] send, from at least one server and to a mobile device requesting and/or a webpage, means for displaying an invitation to subscribe to a text messaging subscription service via a first application;”

107. The combination renders this element obvious.

(a) means for displaying an invitation to subscribe to a text messaging subscription service via a first application

108. The combination renders obvious a mobile device with a means for displaying an invitation to subscribe to a text messaging subscription service via a first application. In Agrawal, the means for displaying is code (e.g., JavaScript) executed by the user’s browser on a mobile device. (Ex. 1005, 5:29-39, 4:10-15). This code is used to generate an advertisement query request that results in the display of an advertisement. (Ex. 1005, 5:29-39). Agrawal explains:

“In one implementation, the advertisement query request may be generated in response to a user accessing a document, such as a web page, with which advertising links are associated. For example, 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). Here, the referenced “browser” is a **first application** on a user device that interprets webpages accessed by the user, including ads on those pages. (Ex. 1005, 1:18-24, 5:64-6:3, 7:1-3, 8:13-37, Figs. 1 and 4). The code (**means for displaying**) that generates the advertisement query request can be, for example, JavaScript instructions. (Ex. 1005, 3:53-59)(Ex. 1006, 026:8-32, 027:10-16, 027:17-028:31, 029:1-5, 029:11-030:1-9, 030:28-30, Figs. 8, 9A-9C).¹ As I explain above (¶60), <script> tags containing JavaScript are executed when a page loads. (Ex. 1012, p. 054)(Ex. 1021, pp. 332, 021, 060). The code is an integration tag (see the claim construction in ¶¶73-74, above) because it is code that carries out the functions of an integration tag in the '897 patent (Ex. 1001, 5:29-36), as explained below.

¹ Citations to Ex. 1006 are to an application (“the '900 application) incorporated-by-reference into Agrawal at Ex. 1006, 5:58-63). Such citations are in the format {page}:{lines}, with {page} indicating the three-digit numbers (bottom) added per 37 C.F.R. §42.64(d)(2)(i).

109. The code (**means for displaying**) causes the generation of an advertisement query request to the server, which sends back an ad for display within the webpage. (Ex. 1005, 3:11-19, 5:25-63). Ads displayed within the webpage are shown as ads 410 in Fig. 4, reproduced here:

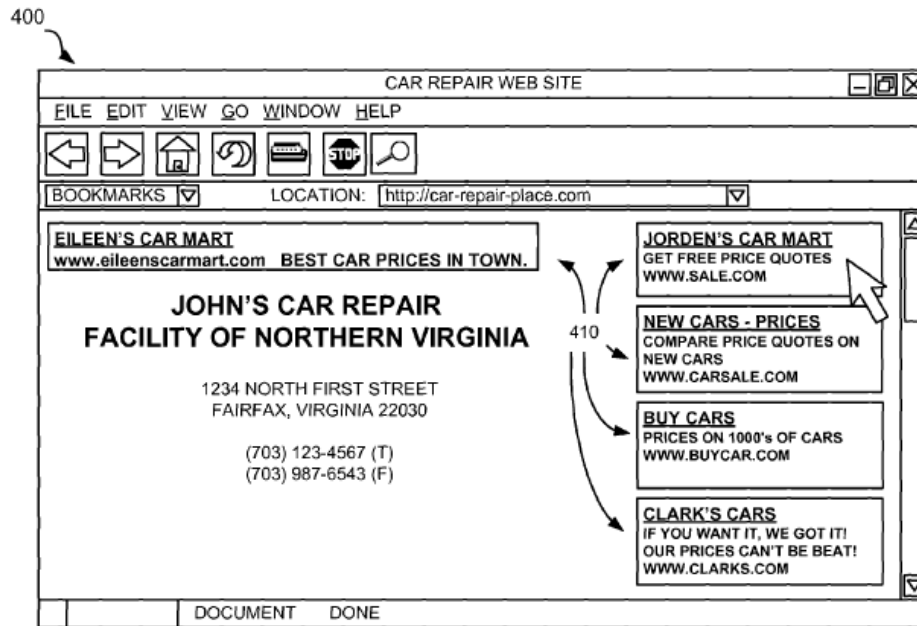


FIG. 4

(Ex. 1005, 8:13-33).

110. It would have been obvious to display an advertisement containing **an invitation to subscribe to a text messaging subscription service**. 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 of “any type of promotional information”, including “coupons or discount codes **delivered to the user either by response message or a later time**, instant-win game chances, **sweep stakes** entries,

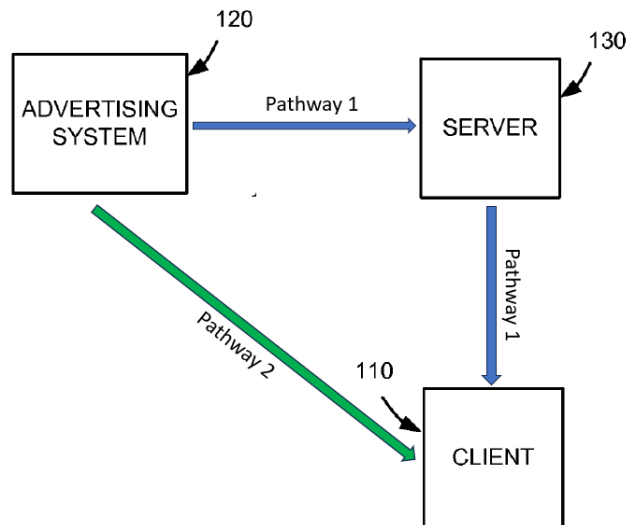
milestones in augmented reality games and/or **marketing promotions**, videos, digital **gifts**, instant connection to tech support and customer service, registration of a purchased product, etc.” (Ex. 1011, ¶0025)(Emphasis added)(see also Ex. 1011, Figs. 8-10, ¶¶0051-0054). Where such benefits are offered, to be provided in subsequent text messages, it would have been obvious to inform the user of this in the advertisement, in order to induce the user to sign up for a promotional subscription via text message. Promising the user a reward in an advertisement to induce the user to sign up for a text-message marketing service was a standard advertising technique in the relevant timeframe, as I explain above in ¶¶24-26.

111. Furthermore, the content of an advertisement is so-called “printed matter”. It bears no technical relationship to the underlying medium, inasmuch as the ad could say anything and the system would still technically function in the same way, although it would have a different influence on a human being that is interacting with it. Because there is no technical relationship between the meaning or content of the advertisement and the way in which the system functions, I understand that, as a matter of patent law, the content of the advertisement is not a limiting part of the claim, and does not need to be shown in the prior art.

(b) send, from at least one server and to a mobile device

112. Agrawal renders obvious that the means for displaying (code) is **sent from at least one server and to a mobile device**. It was obvious to perform the

sending over two different paths, namely (1) from the advertising system 120 to the server of the website publisher 130, from which it will be downloaded by the client device, and (2) from the advertising system 120 directly to the client device, as shown in the following diagram created by the Petitioner:



(b)(i) First pathway: advertising system 120 to mobile devices via web servers 130.

113. First, Agrawal renders obvious sending its code (means for displaying) to web servers that serve webpages to mobile devices, thus causing the code to be sent from the advertising system 120 to the web servers 130 and then to the mobile devices, when the mobile device downloads the page. The advertising system 120 is a **server**. (Ex. 1005, 4:16-18)(“Advertising system 120 may include a **server** entity that maintains and provides advertising links.”)(Emphasis added). Servers 130 are also servers.

114. Agrawal renders obvious having the advertising system 120 **send** the code (means for displaying) to the mobile devices via servers 130. First, Agrawal incorporates the '900 application by reference. (Ex. 1005, 5:58-63). In the '900 application, similar code to be executed by browsers to collect information is “provided to content servers for insertion into all their Web pages”, where the content servers of the '900 application are normal web servers akin to the servers 130 of Agrawal. (Ex. 1006, 026:29-32, 011:7-26, 022:15-023:5, Fig. 4, Fig. 2). To have content servers insert code into their websites, the '900 application explains that website publishers log on to the ad-network system and *download* a piece of code that enables display of an advertisement. (Ex. 1006, 023:1-6, *generally* 022:6-023:21). The downloading is done from the central ad-network server, which in the '900 application is called the “ad entry, maintenance and delivery system 120”. The teachings of the '900 application are incorporated by reference into Agrawal and disclose an ad-network system quite similar to Agrawal's—down to the reference numeral (120) of the central server. (Ex. 1005, 5:58-63, Fig. 1, 3:60-4:35)(Ex. 1006, Figs. 1-2, 010:26-29, 013:21-014:6). Thus, Agrawal suggests, via the incorporated '900 application, to have the central server of the ad system (i.e. advertising system 120) to provide the “code” for insertion into webpages that are eventually downloaded to the mobile device.

115. Furthermore, it was also obvious for servers 120 in Agrawal (client servers) to download (**send**) Agrawal's code (**means for displaying**) from the advertising system 120 (**server**). Having the advertising system 120 (**server**) **send** the **means for displaying** to the client server for incorporation into a website would have been the most obvious way to implement the system. As explained above in the "Overview of the Combination", Agrawal's "code" causes an advertisement query request and detailed user information to be sent to the advertising system 120. (Ex. 1006, 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. For example, in the '900 application that is incorporated-by-reference into Agrawal, there are examples of ad tags (or in the parlance of the '897 patent, integration tags or means for displaying) that send information back to the ad server, e.g.:

In one embodiment of the invention, Javascript is used in the context of a proxy. This Javascript extracts “interesting” document information such as meta tags, header, titles, etc., from any Web page it is available on. A target page could include the following Javascript as embedded instructions:

```
<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>
```

Setting the URL of the iframe instructs the browser to reload the frame. This Javascript could be provided to content servers for insertion into all their Web pages.

(Ex. 1006, 026:18-33). In this excerpt, the script denoted by the scrip tag <script> collects information, and then forms a URL, with the destination server pagead.google.com/ads and parameters such as “contents=word1+word2” encoding the collected information. The script then sets the target address of an iframe to this URL, causing the iframe to reload, and causing an ad from pagead.google.com to appear in the iframe. Because the correct place (pagead.google.com/ads in the example above) and correct form (encoded as certain parameters in a URL in the example above) are best-known to the administrators of the advertising system 120 (pagead.google.com/ads in the example above), it would have been obvious for the advertising system 120 to provide the code to client servers initially. Furthermore, in operation at scale, many thousands of Agrawal’s “pieces of code” may need to be defined in a short period of time. Each tag can have unique characteristics for the particular publisher site, such as a unique identifier assigned to the site and particular

business rules that are executed. (Ex. 1006, 023:1-5). In order to generate a unique identifier and business rules with each piece of code (means for displaying) during large-scale operations, it would have been most obvious to configure the advertising system 120 to define and generate the pieces of code and send them to publisher servers. 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 their web sites, such that the code will transmit information and request advertising when a consumer's browser downloads and interprets the web site. 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 elements according to known methods, yielding predictable results, and thus obvious.

(b)(ii) Second pathway: advertising system 120 directly to mobile devices.

116. Furthermore, Agrawal renders obvious sending code directly to mobile devices. Specifically, Agrawal, via the incorporated '900 application, teaches that code can be included using a JavaScript link within a webpage. When that occurs, HTML code within the webpage contains a <SCRIPT> tag having an "src" parameter. When this parameter is interpreted by a browser, the browser will request the JavaScript code directly from the server. For example, the '900 application teaches the following code in a webpage:

“One alternative to embedded Javascript is to provide a static Javascript link. More specifically, to avoid increasing the size of each Web page by 4KBytes or so by embedding Javascript, a link to a static Javascript page can be used instead. The following is an example of a link to a static javascript page:

<script src=pagead.google.com/ads/scrape.js> </script>

Since most browsers will cache this Javascript link, the actual Javascript will only be fetched infrequently.”

(Ex. 1006, 028:01-028:31). The script tag in the quote above (denoted with <script>) has an “src” parameter that designates the source (location) of the script, namely the file “scrape.js”, located again on the server at pagead.google.com/ads, which is the same central server that serves the advertisement. (Ex. 1006, 026:18-32). When the browser loads the webpage containing this <script> tag, it executes the tag, causing the mobile device to download the script from the central server. This causes the central server to **send, to the mobile device**, the relevant code, which in Agrawal’s case, is the **means for displaying**. This technique of having code downloaded directly from the central ad server by the mobile device when it loads a webpage by using a “src” parameter was a standard JavaScript / HTML technique and was well-known in the art (Ex. 1006, 028:01-028:31), and could have been implemented for Agrawal’s code without unpredictable results. The technique entailed advantages in certain situations because it avoided having to include the entire code within the

webpage itself, making the webpage smaller, allowed the central ad server to change the code for multiple client websites without requiring each client to perform an update. Agrawal also makes clear that the servers 130 and the Advertising System 120 and can be combined into a single server (Ex. 1005, 4:28-35). When that is the case, the mere act of downloading a webpage with embedded code from the publisher server 130 to the mobile device would also be sending the code from the advertising system 120 directly to the mobile device.

“[15c] send, from the at least one server and to the mobile device, means for causing (1) the mobile device to transition from the first application to a second application different from the first application in response to the mobile device detecting a first interaction with the invitation, the second application being a messaging application, and”

117. The combination renders this limitation obvious. In brief, when a user clicks on a advertising link in an ad, the functionality of Molinet sends, from a server, a properly-formatted URI (the **means for causing**) which causes the mobile device to transition from the browser (first application) to a messaging application (second application).

(a) in response to the mobile device detecting a first interaction with the invitation

118. As explained above under limitation [15b], Agrawal teaches displaying advertisements in webpages, where the advertisements have hyperlinks that can be clicked. (Ex. 1005, Fig. 4, 8:13-32). The clicking of a link in an ad is a

first interaction with the invitation. The click must and obviously is **detected by the mobile device**, because the mobile device responds to the click. It was obvious in the relevant timeframe for a mobile device to **detect** user input. 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.

(b) send, from the at least one server and to the mobile device, means for causing

119. As explained above, it would have been obvious to implement the business method of Oliver using mobile ads, by using the web-based deeplinking technology of Molinet.

120. Molinet teaches that a user can interact with a link (URL) in a webpage, causing the browser to send a request to the server. Molinet explains:

“In one embodiment, the contextual deep link is a URL and the contextual deep linking server 110 receives a request associated with the contextual deep link from a web browser 123 of a client device 120.”

(Ex. 1008, ¶0065).

121. The server of Molinet then checks whether it has contextual data for the mobile device, for example, whether the server already has a list of installed applications for the mobile device. (Ex. 1008, ¶¶0074-0082). If the server has

such information, a “contextual response module” at the server chooses an appropriate URL (a type of URI) to send back to the mobile device, which will cause a transition to another mobile app. (Ex. 1008, ¶¶0081-0082). Molinet explains:

“In one embodiment, for a client device 120 that supports an application ecosystem, the contextual response module 340 uses the matching contextual information to **determine whether the application 121** that the contextual deep link includes link data for **is installed** on the client device 120....[¶0082]. **If the application is installed**, the contextual response module 340 includes in the response a message that causes the client device 120 to initiate (e.g., open or bring into focus) the application 121 associated with the contextual deep link. For example, **the response may include a URL identifying the application 121 that causes the operating system on the client device 120 to initiate the application.**”

(Ex. 1008, ¶¶0081-0082). The “URL” is simply a more specific form of URI that also complies with RFC 3986. (Ex. 1033). The URI “causes the operating system on the client device 120 to initiate the application”, which in the case of the combination, is the messaging application. (Ex. 1011, ¶¶0051-0054, Fig. 10). Thus, the combination teaches sending, from the at least one server and to the mobile device, **means for causing the mobile device to transition from the first application to a second, application different from the first application**, where

the **second application is a messaging application.**

“[15d] (2) the second application to automatically populate a custom text message based on the user data;”

122. It would have been obvious, for reasons discussed above in ¶¶90-96, to implement Oliver’s business method together with Agrawal and Molinet.

(a) automatically populate a custom text message

123. Oliver’s business method includes using application deeplinks to **automatically populate a custom text message based on the user data.** (*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 custom text message (in a text-messaging application) having an address 210 and a message body 220:

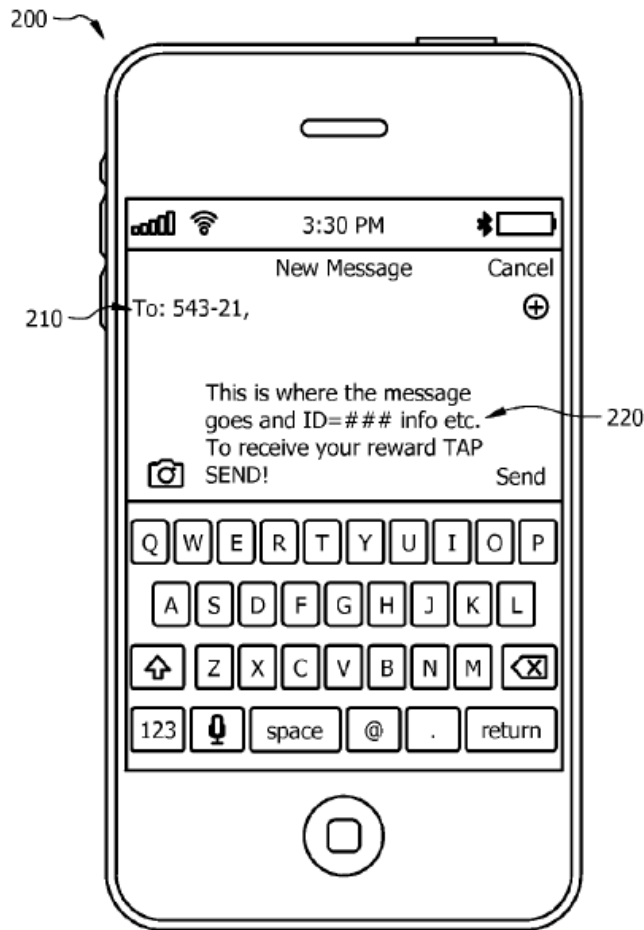


FIG. 2

(Ex. 1011, Fig. 2, ¶¶0027-0030, 0012).

124. 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 deeplink to an SMS (text messaging) application. (*citing* Ex. 1016, p. 584). 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). It would have been obvious to use smsto: or sms: links as the Molinet URLs, because such links were already available in the most-common mobile operating systems. (Ex. 1016, p. 557)(Ex. 1010, pp. 015, 021)

125. 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). 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 Molinet. (Ex. 1006, ¶¶0027, 0063). Using Molinet’s techniques to activate a prepopulated text message as taught by Oliver would result in **automatically populating a custom text message based on the user data.**

(b) a custom text message based on the user data

126. It would have been obvious to generate the custom text message based on user data. I note here that there is no antecedent basis for “the user data”.

127. The combination renders obvious **basing** the automatically-generated text message **on user data** in three ways. First, Oliver teaches including user identification information in the text message (Ex. 1011, ¶¶0030, 0037, 0008, Abstract). Oliver explains:

“The **prepopulated text message 220 can also contain** unique identifiers such as promotional identifiers and/or specific keywords that can be extracted by the network application to determine promotional information....the unique identifier, which has been previously associated with the item of value that is the subject of the promotion, may be included in any part of the message, and it may be visible to the user or not visible, and instead embedded within the prepopulated message. **This is also the case with any other information** being transmitted using the prepopulated message, such as the **user identification information**, the ‘To’ address of the message, or any other information.”

(Ex. 1011, ¶0030)(Emphasis added). The “user identification information” is **user data**, because it identifies a user. (Ex. 1011, ¶0008, cls. 13, 18).

128. Second, it would have been obvious to include Agrawal’s “unique identifier” in the text message. Agrawal explains:

“Each advertising link that is returned in response to the advertisement query request 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)(Emphasis added). That is, Agrawal teaches that the advertising system 120 responds to an ad query request by sending to the mobile device a unique

identifier that identifies the unique webpage, unique advertising link, and unique user involved in a transaction, and is thus **user data**. Agrawal teaches that this same identifier is, in turn, sent from the mobile device back to the advertising system 120 when the link is clicked. Agrawal states:

“An advertisement click request may be received in those situations when the user selects one of the advertising links....the advertisement click request may include the unique identifier that is associated with the selected advertising link.”

(Ex. 1005, 6:55-58).

129. It would have been obvious to configure the text-message URL to include Agrawal's unique identifier in the body of the prepopulated text message, based on the express teaching of Agrawal to send back the unique identifier to the advertising system 120 when the link is clicked (Ex. 1005, 6:55-58), and the express suggestion of Oliver to include ad and user identifiers in the text message (Ex. 1011, ¶0030). Furthermore, it would have been obvious to send Agrawal's unique identifier because it would have allowed the system to track the user, webpage and advertisement through the text-based enrollment process. (Ex. 1005, 1:27-33, 3:23-35). This was important, because the same ad and link could appear on different webpages, different ads can have deeplinks leading to an SMS signup with the same advertiser, and the same user can click multiple links. Thus, it was desirable for the system to identify which users were clicking through which ads on

which sites. This would have obviously provided information on which ads and which sites are effective, and also would have provided information necessary for payments between the advertisement system operator, advertisers and publishers, and to limit the number of offers provided to the same user. (Ex. 1005, 1:34-49)(Ex. 1011, ¶¶0045-0047).

130. Third, it would have been obvious to base the text message on “user data” collected by the Agrawal’s advertising system 120 (*see* Overview of Agrawal, above), in order to use the user data to authenticate the user. Specifically, Agrawal teaches a method of authentication that includes a comparison of user data before and after clicking an ad link, where a significant change in the user data indicates potential fraud. (Ex. 1005, 2:23-40, 7:45-8:12). Both Agrawal and Oliver emphasize the advantages of authenticating the user to prevent fraudulent payments to publishers for click-throughs, and to prevent the same users from redeeming offers too many times. (Ex. 1005, 1:33-58)(Ex. 1011, ¶¶0045, 0047). By including the user data or some subset thereof in the text message, the mobile device can provide confirmation that it is the same mobile device on which the advertisement was displayed, thus allowing Agrawal’s system to confirm that text message responses are not being generated to increase a given publisher’s payments. (Ex. 1005, 1:33-58)(Ex. 1011, ¶¶0045, 0047).

“[15e] receive, from the mobile device, the custom text message in

response to the mobile device detecting a second interaction with the messaging application; and”

131. The combination renders this limitation obvious. 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). A PHOSITA in the relevant timeframe would have understood and found obvious that the tap on the “send” button of a messaging application would have to be **detected**, and did, in fact, trigger “sending” of the message to the destination number (here, the advertising system 120). Pressing or tapping the ‘send’ button is a **second interaction**, because the first interaction was clicking the ad. (Ex. 1011, ¶0009)(“the disclosed principles streamline the response process by simply requiring only two actions by the user...”). The claim language “second interaction with the messaging application” does not require two interactions *with the messaging application*, because the “first interaction” is expressly recited as being with the “invitation” (the ad), not with the messaging application.

132. Oliver teaches that its text messages are directed to a web service. (Ex. 1011, ¶0038). As explained above under the Overview of the Combination and Rationale (Motivation) for the Combination, it would have been obvious to operate Oliver’s web service on Agrawal’s advertising system 120, such that the processor **receives** the text message.

“[15f] deliver an installment of the text messaging subscription service without receiving an opt-in communication after receiving the custom text message.”

133. The combination renders obvious delivering an installment of the text messaging subscription service without receiving an opt-in communication after receiving the custom text message. Specifically, Oliver teaches that when the custom text message is received, the user is registered and a responsive text message is sent to the user. This is shown, for example, beginning in Fig. 9 of Oliver, shown below with highlighting added:

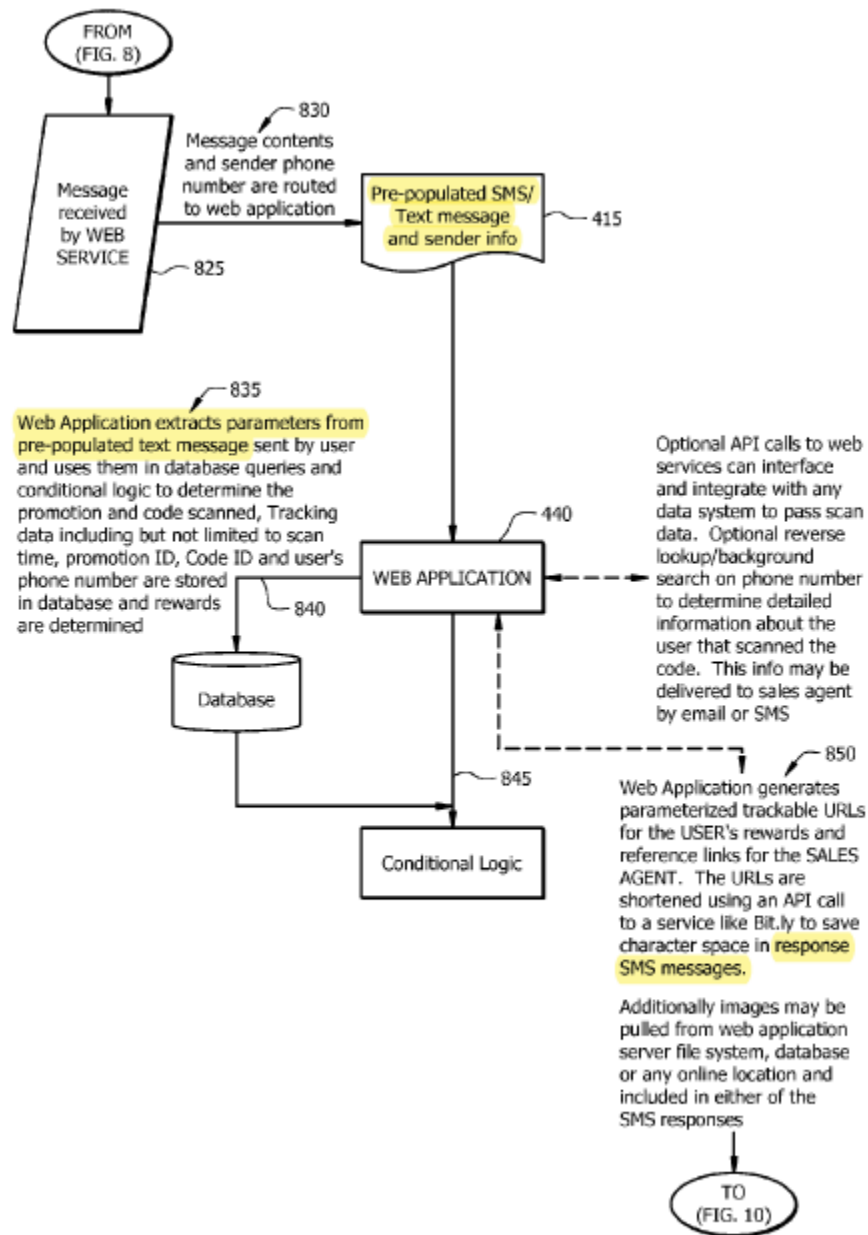
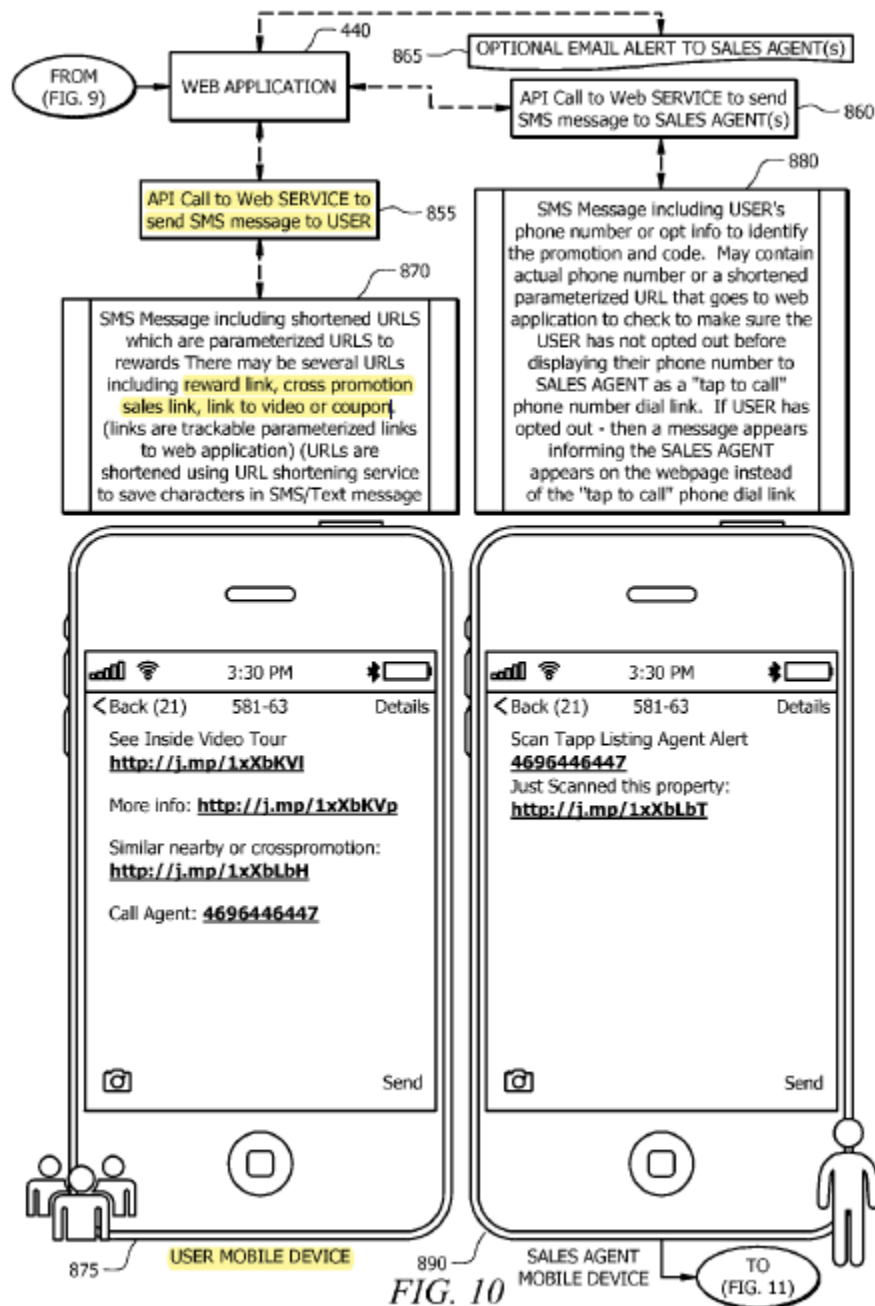


FIG. 9

(Ex. 1011, Fig. 9, ¶¶0051-0054). In Fig. 9, the server receives the pre-populated text message at step 825, extracts the relevant information at step 415, performs database checks with the information at step 835, and defines a response text message at step 850. (Ex. 1011, Fig. 9, ¶0053). The process continues with Fig. 10, reproduced below with highlighting added:



(Ex. 1011, Fig. 10, ¶¶0051-0054). In Fig. 10, Oliver shows (see numerals 855, 870, and 875) the generation of a response SMS to the user's device having reward or sales information in it. (Ex. 1011, Fig. 10, ¶0054). This is the **delivery of an installment of the text messaging subscription service**. As shown in Figs. 8-10 of

Oliver, this is performed **without receiving an opt-in communication after receiving the custom text message**. (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). I further understand that, in patent law, a reference need not state a feature’s absence in order to disclose a negative limitation.

CLAIM 16

“16. The non-transitory processor-readable medium of claim 15, wherein: the means for displaying comprises an integration tag; and the means for causing comprises a Uniform Resource Identifier (URI).”

134. In my opinion, the combination renders this claim obvious.

(a) means for displaying comprises an integration tag

135. See above, claim 15, limitation [15b]. Agrawal’s code (means for displaying) is an integration tag, because it is code (JavaScript) that performs the functions of an integration tag. As stated in the ’897 patent:

“In some embodiments, the memory 103 can store code representing integrator instructions 105 (i.e., integration tag) to be executed by the at least one processor 101 or the user mobile device 150. The execution of the integrator instructions 105 at the user mobile device 150 can cause the user mobile device 150 to send a request to the click-to-text server 100 and receive from the click-to-text server 100 promotional messages associated with the client.”

(Ex. 1001, 5:29-36)(*see also* discussion under claim 15, limitation [15b]).

(b) means for causing comprises a Uniform Resource Identifier (URI)

136. See above, claim 15, limitations [15c]-[15d]. As discussed there, the means for causing is Molinet's URL. (Ex. 1008, ¶0081-0082). The "URL" is a type of URI. .

CLAIM 17

"17. The non-transitory processor-readable medium of claim 15, wherein the means for displaying includes code to cause the mobile device to send user data to the at least one server, the code further comprising code to cause the processor to: define the means for causing based on the user data."

137. In my opinion, the combination renders claim 17 obvious.

(a) the means for displaying includes code to cause the mobile device to send user data to the at least one server

138. First, both Agrawal and Molinet teach collecting user data using JavaScript embedded in the webpage. Agrawal teaches that when a user's device loads a webpage, the code (**means for displaying**) causes an advertisement request query to be generated:

"[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). The advertisement query request is an HTTP request that **sends user data** to the advertising system 120 as a directed result of the execution of the “piece of code” (means for displaying). (Ex. 1005, 6:40-43, 5:29-39). The destination for the information items sent by the code (means for displaying) is the advertising system 120, which is the **at least one server**. (Ex. 1005, 4:16-22, 5:3-65, Fig. 3, 10:3-6).

139. Agrawal also teaches that code running on the browser can collect and **send user data to the** advertising system 120. Agrawal states:

“The information relating to the advertisement query request may be obtained in various ways. For example, in one implementation consistent with the principles of the invention, the information may be obtained via the browser used by client 110. **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, 6:30-40)(Emphasis added). A PHOSITA would have found obvious to implement the “piece of code” that collects further user data (Ex. 1005, 6:30-40) together with the “piece of code” that generates an advertisement query request (Ex. 1005, 5:29-39). Agrawal teaches that user data is “collected with the advertisement

query request”, from which a PHOSITA would have understood that the advertisement query request *includes* the user data sent to the advertising system 120. (Ex. 1005, 3:26-42). Because the advertising query request is generated by a “piece of code” (integration tag) operating on the user’s browser (Ex. 1005, 5:29-39), and because the user data is also collected by a “piece of code” operating on the user’s browser (Ex. 1005, 6:30-40), a PHOSITA would have understood (and found obvious) that Agrawal’s piece of code (means for displaying) also collects and transmits the user data described in Agrawal at 5:64-6:29. Having the same body of code collect user data and transmit the advertisement query request would have been a trivial matter for a PHOSITA, and would have allowed the query request to transmit the user data, fulfilling Agrawal’s recommendation that user data be “collected with the advertisement query request”. (Ex. 1005, 3:26-42). Agrawal furthermore expresses flexibility in how user data is collected. (Ex. 1005, 6:30-31, 6:43-46). Even if separate “pieces of code” were used, however, they would still be **a means for displaying**, and the means for displaying would still **cause the mobile device to send user data to the at least one server** per the claim.

140. Molinet provides a very similar teaching, namely that JavaScript operating in a browser collects user data and sends it to the central server, which in the combination is included within Agrawal’s advertising system 120, **at least one server**. (Ex. 1009, ¶¶0037, 0065).

(c) user data

141. The information collected by Agrawal's system includes types of information specifically identified in the '897 patent as “**user data**” (*compare* Ex. 1001, 5:12-17), including user cookie information, “other user information”, user agent data (browser type, client (mobile device) type, operating system information), the user device's IP address, the page referrer URL, and specific configuration settings for the user's device and browser. (Ex. 1005, 5:45-6:46, 8:20-33). Molinet teaches collecting similar information. (Ex. 1009, ¶0037, 0065).

(c) the code further comprising code to cause the processor to: define the means for causing based on the user data

142. The combination renders obvious causing the processor to define the means for causing based on the user data in two different ways.

143. First, as explained above under claim 16, it would have been obvious to use user data collected by Agrawal's code (means for displaying) in the formulation of the text message. To do so, it would have been obvious to include user data items in the URL used to transition the mobile device from the browser of the mobile device to the text-messaging application—a technique expressly taught by Oliver. Oliver shows, for example, that information to be included into the body of a text message can be encoded into a link:

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

(Ex. 1011, ¶0028). Furthermore, Molinet teaches using known URI schemes that work with typical mobile operating systems. (Ex. 1009, ¶0082). Additionally, Molinet's incorporated provisional application teaches "If the user has the app, the system uses a standard URI scheme to take them to the appropriate content" and "'direct deeplink' is existed functionality provided by the iOS and Android operating systems wherein an app can register itself on the user's phone as a handler for a certain URI scheme (a link starting with "myapp://"), and when the user is directed to a link starting with the URI scheme (*e.g.*, "myapp://open"), the app which registered the URI scheme is opened, and the app is informed of any data associated with the link that send the user to it.". (Ex. 1010, pp. 015 and 021).

144. In this manner, it would have been obvious to include user data intended for the body of the text message, as taught by Oliver, in the text of the URL provided for contextual deeplinking as taught by Molinet. Thus, **the processor defines the means for causing based on the user data.**

145. Second, Molinet teaches that **user data** collected from a mobile device is used to determine the proper context in which the deeplink is to operate. (Ex. 1009, ¶¶0079-0082). Using the user data, the system defines a URL that is likely to work on the mobile device in question (Ex. 1009, ¶¶0079-0082). In this

way, the combination also renders obvious **having the processor define the means for causing based on the user data.**

CLAIM 18

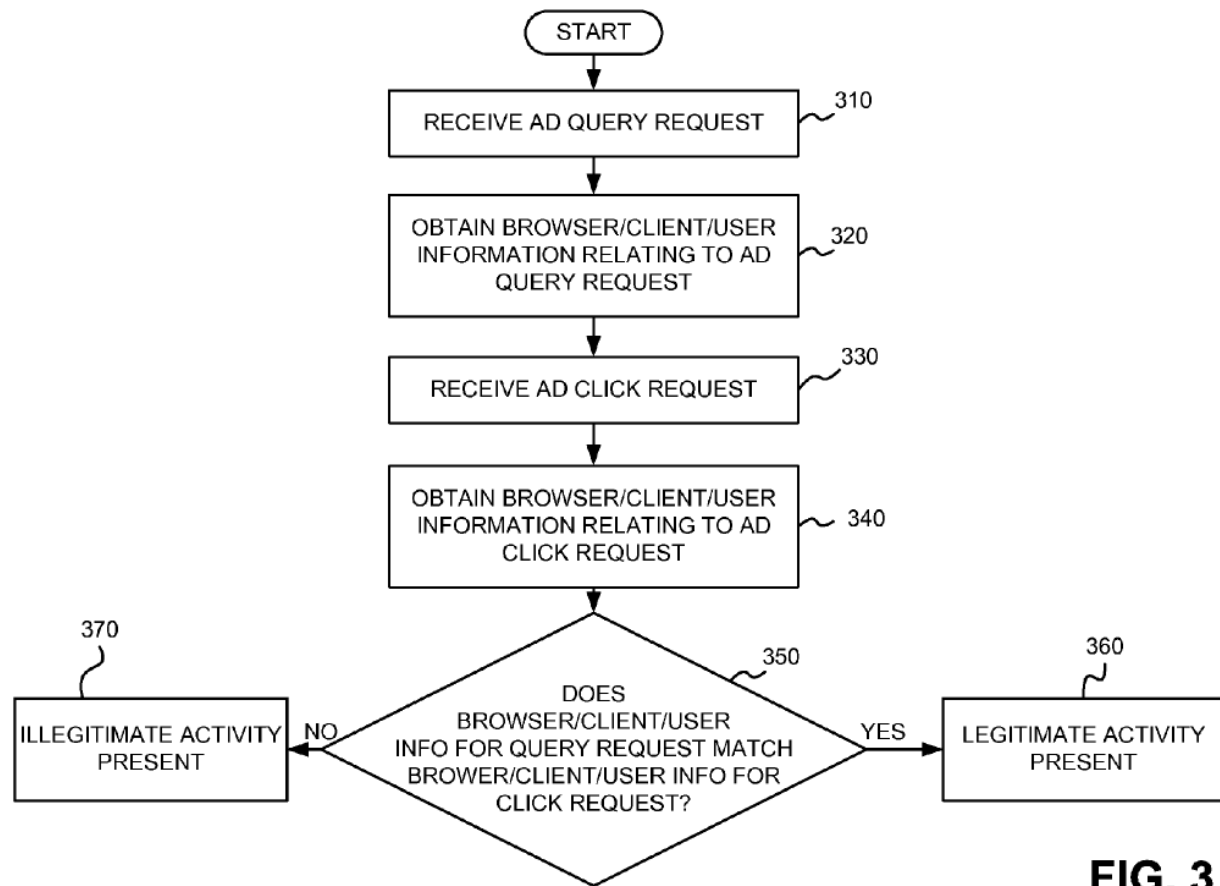
“18. The non-transitory processor-readable medium of claim 15, the code further comprising code to cause the processor to receive a cookie identifier in response to at least one of the mobile device requesting and/or loading the webpage or the mobile device executing the means for displaying.”

146. In my opinion, the combination renders this claim obvious. Both Agrawal and Molinet teach receiving cookie information when the mobile device loads the webpage (and thereby also executes the means for displaying). (Ex. 1005, 6:28-29, 5:64-6:29, claims 8, 13, 24)(Ex. 1009, ¶¶0037, 0065). The cookie information is specifically an “identifier” because, for both Agrawal and Molinet, the cookie information is used as a method for identifying the mobile device. (Ex. 1005, 7:45-8:12)(Ex. 1009, ¶¶0074-0082).

CLAIM 19

“19. The non-transitory processor-readable medium of claim 15, wherein: the custom text message includes a second user identifier, the code further comprising code to cause the processor to: receive a first user identifier in response to at least one of the mobile device requesting and/or loading the webpage or the mobile device executing the means for displaying; and authenticate the mobile device based on matching the second user identifier matching the first user identifier, the installment of the text messaging subscription service delivered in response to authenticating the mobile device.”

147. In my opinion, the combination renders this claim obvious. First, Agrawal teaches a method of collecting user data a first time and a second time, comparing the two sets of collected user data, and authenticating the user if the two sets of user data match. (Ex. 1005, 7:64-8:12, *see also* 5:25-8:12, Fig. 3). This is shown in Fig. 3 of Agrawal, reproduced here:



(Ex. 1005, Fig. 3, 5:25-8:12).

148. As discussed above under claim 17, both Agrawal and Molinet teach collecting information that can be used to identify the user. (Ex. 1005, 5:45-6:46, 8:20-33)(Ex. 1009, ¶¶0037, 0065). This information is collected and sent to the

server when the browser executes code in the webpage, *i.e.* when the browser **executes the means for displaying** while loading the webpage. Thus, the combination teaches **receiving a first user identifier in response to at least one of the mobile device requesting and/or loading the webpage or the mobile device executing the means for displaying.**

149. It would have been obvious to place user information in the text message generated per Oliver's teachings, such that the user information can be compared with information collected earlier to authenticate the user per Agrawal. As discussed above under claim 15, limitation [15d], Oliver expressly teaches including "user identification information" in the pre-populated text message. (Ex. 1011, ¶030). Oliver also expressly teaches that it is advisable to authenticate users, to prevent the user from entering a sweepstakes or giveaways too many times, stating:

"The disclosed principles can further provide a mechanism to authenticate a user and optionally limit the number of entries into sweepstakes and giveaways, or simply the number of times the user responds to a specific promotion."

(Ex. 1011, ¶0047).

150. Because Agrawal teaches the basic method of comparing sets of information to authenticate a user, and teaches collecting user-identifying information to do so, while Oliver also teaches the advisability of authenticating

users prior to entry into sweepstakes and giveaways, and expressly teaches to place user-identifying information in the pre-populated text message, it would have been obvious to have the **custom text message include a second user identifier**, and to **authenticate the mobile device based on matching the second user identifier matching the first user identifier**, the installment of the text messaging subscription service (as discussed above under claim 15, limitation [15f]) delivered in response to authenticating the mobile device.

CLAIM 21

“21. The non-transitory processor-readable medium of claim 15, the code further comprising code to cause the processor to: determine whether the mobile device has previously subscribed to the text messaging subscription service, the installment of the text messaging subscription service delivered based on determining that the mobile device has not previously subscribed to the text messaging subscription service.”

151. In my opinion, the combination renders this claim obvious. Oliver teaches checking the user phone number to determine whether the device has been used to gain access to a reward more than a certain number of times, optionally with a time limit between allowed uses. (Ex. 1011, ¶¶0045, 0047). Oliver states:

“With certainty, a system or method in accordance with the disclosed principles can register and **limit the specific number of scans by the same user** using the same communications device scanning the same QR code because the communications device’s phone number has been acquired during each scan and saved into the database...”

(Ex. 1011, ¶0045)(Emphasis added). It would have been obvious that, regardless of the limit on the “number of scans” (*i.e.* the number of activations of an ad link), a user that previously had zero activations (a new user) would not be blocked by a limit on the number of subscription attempts. It would furthermore have been obvious based on Oliver’s teachings to deliver the installment of the text messaging subscription service **based on determining that the mobile device** has zero prior attempts to subscribe, *i.e.* that the user has **not previously subscribed to the text messaging subscription service**.

CLAIM 22

“22. The non-transitory processor-readable medium of claim 15, the code further comprising code to cause the processor to: determine whether the mobile device has previously subscribed to the text messaging subscription service; and send a message to the mobile device indicating that the mobile device is already subscribed to the text messaging subscription service in response to determining that the mobile device has previously subscribed to the text messaging subscription service.”

152. In my opinion, the combination renders this claim obvious. As discussed above under claim 21, Oliver teaches checking for a prior number of attempts by a user to obtain some benefit (such as a reward or entry into a sweepstakes), and that the user should be blocked after a certain number of times. Because Oliver teaches sending a response SMS (see above, claim 15, limitation [15f], Ex. 1011, Figs. 9-10, ¶¶0051-0054)), it would have been obvious to inform that user that the user had previously responded and was already registered.

Furthermore, the content of the responsive text message (*i.e.*, a message **indicating that the mobile device is already subscribed to the text messaging subscription service**) is “printed matter” that is unconnected to the technical functioning of the system, and is therefore not limiting. *See Distefano*, 808 F.3d at 851.

CLAIM 23

“23. The non-transitory processor-readable medium of claim 15, the code further comprising code configured to cause the processor to: subscribe the mobile device to the text messaging subscription service, the first interaction and the second interaction being the only user interactions with the mobile device between sending the means for receiving and receiving the custom text message.”

153. In my opinion, the combination renders this claim obvious. *See* above, claim 15, limitation [15f]. Oliver expressly teaches that:

“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). The two actions correspond to the first interaction (clicking the ad link) and the second interaction (pressing or tapping the ‘send’ button). Again, a prior art reference need not state a feature’s absence in order to disclose a negative limitation.

CLAIM 24

“24[a]. A method, comprising: receiving, at at least one server, an indication of a web browser application executing on a mobile device requesting and/or loading a webpage containing a promotional message offering a subscription to a text message subscription service;”

154. In my opinion, the combination renders this claim obvious. As explained above under claim 15, limitation [15b], Agrawal teaches that when a user loads a webpage on a web browser, JavaScript instructions will cause the mobile device to send an ad query request to the advertising system 120. (Ex. 1005, 5:29-39). The ad request is **an indication of a web browser application executing on a mobile device requesting and/or loading a webpage**. The webpage is the webpage originally requested by the user, which will, when it receives a response from the advertising system 120 and finishes loading, contain the promotional message.

155. The *content* of the promotional message is printed matter unrelated to the technical functioning of the system, and thus not limiting. *See Distefano*, 808 F.3d at 851. However, in the combination, the promotional message would obviously **offer a subscription to a text message subscription service**, because Oliver teaches that the distribution of marketing messages containing discounts and coupons is a possible use of advertisement actions (*see above*, claim 15, limitation [15b] and [15e]), and it was obvious to inform the reader of a

promotional message how to obtain such messages, in case they were of interest to the user. This is basic marketing, and would have been obvious to a PHOSITA in the relevant timeframe.

“[24b] defining, at the at least one server, a uniform resource identifier (URI) that deeplinks to a text messaging application that is different from the web browser application;”

156. See above, claim 15, limitation [15c]. The URI described there **deeplinks** to the text messaging application because it causes the mobile device to generate a message, rather than simply opening the top screen of the text-messaging application.

“[24c] sending the URI from the at least one server to the mobile device,”

157. See above, claim 15, limitation [15c].

“[24d] the URI configured to, in response to a first user interaction with the promotional message via the web browser application, cause the mobile device to (1) automatically render a user interface of the text messaging application and”

158. See above, claim 15, limitation [15c]. When the mobile device transitions to the text messaging application, it would have been obvious to draw the text messaging app on screen automatically (*i.e.* **render a user interface of the**

text messaging application), because otherwise the user would not see the text message that was generated.

“[24e] (2) automatically populate a custom text message with a destination address for a recipient of the custom text message, the text messaging application configured to send the custom text message upon detection of a second user interaction with the text messaging application; and”

159. See above, claim 15, limitations [15c] and [15d].

“[24f] delivering an installment of the text message subscription service based on the first user interaction and the second user interaction and without receiving any further indications of user interactions with the mobile device between receiving the custom text message and delivering the installment of the text message subscription service.”

160. See above, claim 15, limitations [15d]. The combination does not disclose receiving any further indications of user interactions with the mobile device between receiving the custom text message and delivering the installment of the text message subscription service, and thus discloses **without receiving** such indications. Again, a prior art reference need not state a feature's absence in order to disclose a negative limitation.

CLAIM 25

“25. The method of claim 24, further comprising: receiving user data in response to the web browser application requesting and/or loading the webpage; ascertaining, based on the user data, at least one of an identity of at the mobile device or an identity of the user

associated with the mobile device, the URI defined based on at least one of the identity of the first mobile device or the identity of the user associated with the mobile device.”

161. In my opinion, the combination renders this claim obvious. *See* above, claim 17. As described there, both Agrawal and Molinet teach collecting user data, and that the user data is used to define the URI in the combination.

162. The present claim 25 further requires ascertaining, based on the user data, at least one of an identity of at the mobile device or an identity of the user associated with the mobile device, where the URI is defined based on at least one of the identity of the first mobile device or the identity of the user associated with the mobile device. This was obvious in two ways. First, Molinet expressly teaches that the collected user data is used to form a device fingerprint that identifies the user device, and that the identity of the user device is used to define the URI such that the URI will work on the user device. (Ex. 1009, ¶¶0065, 0078-0082). Second, as discussed above under claim 17, it would have been obvious to include user data (and specifically a user identifier that indicates the identity of the user as discussed under claim 19) in the body of the message itself, and thus in the URI that generates the message, as taught by Oliver. (Ex. 1011, ¶¶0027-0029).

CLAIM 26

“26. The method of claim 24, further comprising: sending, from the at least one server and to the mobile device, code to cause the mobile device to store a cookie in response to the web browser application executing on the mobile device requesting and/or loading the

promotional message; and receiving user data associated with the cookie in response to the first user interaction with the promotional message, the custom text message including customized content that is associated with the cookie.”

163. In my opinion, the combination renders this claim obvious.

Specifically, Molinet teaches collecting user data and forming it into a “fingerprint” to identify the device, which the server then causes to be stored in a cookie on the mobile device. (Ex. 1009, ¶0065-0067). This is done using **code sent from the at least one server and to the mobile device**. Cookies were usually set in one of two ways: using instructions sent as part of an HTTP message from the server to the mobile device, or using JavaScript (which is sent from the server to the mobile device as discussed above under claim 15, limitation [15b]). Agrawal expressly states that its user data can be cookie data (Ex. 1005, 7:25-28) as does Molinet (Ex. 1009, ¶¶0065-0067), and it would have been obvious to receive user data **associated with a cookie in response to the first user interaction with the promotional message** and to include the user data in the promotional message. Specifically, Molinet teaches to look up cookie data in the form of a “locally-stored fingerprint” after a contextual deeplink is clicked, and to formulate a response with a link that works for the particular device based on the user data, as shown in Fig. 5, reproduced below:

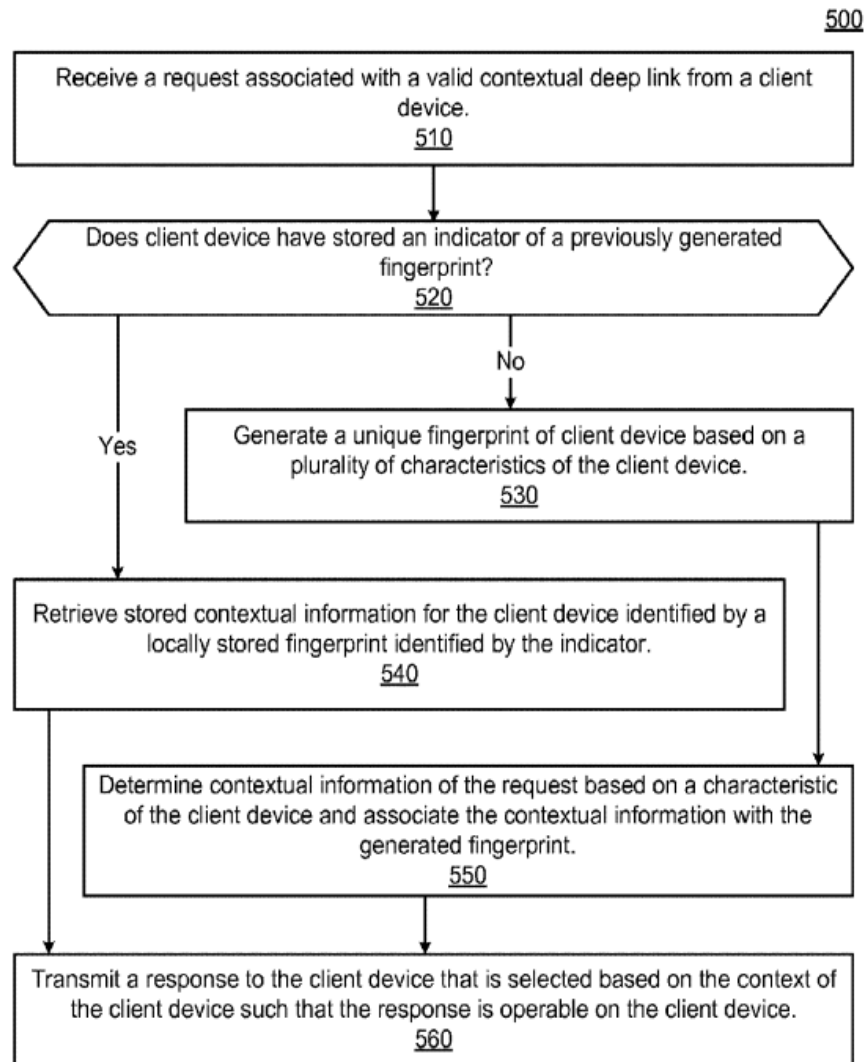


FIG. 5

(Ex. 1009, Fig. 5, ¶¶0102-0107). In Fig. 5, step 510 includes a user clicking on a link, which in the combination would be an ad link. (Ex. 1009, ¶0102). The system then checks whether there is stored data matching the user's fingerprint, which Molinet's server can cause to be stored in a cookie on the mobile device. (Ex. 1009,

¶¶0103, 0067). The system then takes the fingerprint data, uses it to look up context data about the user's device (steps 540 and 550), and sends a properly formatted link (URI) back to the device (step 560). (Ex. 1019, ¶¶0103-0107).

164. Furthermore, as explained above under claims 17 and 19, it would have been obvious to use user data, and in particular, a user identifier within the custom text message to carry out the method of Oliver. . Because Agrawal teaches collecting user data including cookie data after a link is clicked (Ex. 1005, 6:46-7:28), and Molinet teaches a specific method of using user data in cookie data to formulate a link to send to the device to effectuate mobile app deeplinking, and because it was obvious to use this user data in a pre-populated SMS as taught by Oliver, this claim was further obvious.

CLAIM 27

“27. The method of claim 24, further comprising: receiving user data in response to the web browser application requesting and/or loading the webpage; and identifying at least one of the mobile device or the user of the mobile device based on the user data, the URI defined based on an identity of the at least one of the mobile device or the user of the mobile device.”

165. In my opinion, the combination renders this claim obvious. *See* above, claim 25, which has the same substantive content worded slightly differently. The differences in claim language are not material to the claim scope, and the discussion under claim 25 applies to claim 27 with equal force. .

CLAIM 28

“28. The method of claim 24, wherein the custom text message complies with opt-in requirements for enrollment in the text message subscription service such that the mobile device can be enrolled in the text message subscription service without sending or receiving an additional authorization from the mobile device.”

(a) enrolled in the text message subscription service without sending or receiving an additional authorization from the mobile device

166. In my opinion, the combination renders this claim obvious. See the discussion above under claim 15, limitation [15f], explaining Oliver’s response messages. As shown in Figs. 8-10 of Oliver, enrollment in a promotion takes place without sending or receiving an additional authorization 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, it would be obvious that enrollment in and participation in the subscription is completed **without sending or receiving an additional authorization from the mobile device**. This is because no further communications are needed in order enroll and participate in the subscription. Again, a prior art reference need not state a feature’s absence in order to disclose a negative limitation.

(b) the custom text message complies with opt-in requirements for enrollment in the text message subscription service

167. Regarding the limitation “complies with opt-in requirements for enrollment in text message subscription services”, the specification of the ’897 patent teaches that use of the “two-tap” method (such as Oliver’s) complied with opt-in requirements for enrollment in text message subscription services. (Ex. 1001, 7:19-23)(*See also* Ex. 1015, pp. 122-123) .

168. It further 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 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, 7:19-23). 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 as admitted by the ’897 patent, the mobile device **can be automatically subscribed to the subscription list without sending or receiving an additional authorization from the mobile device.**

CLAIM 29

“29. The method of claim 24, further comprising: determining that the mobile device has not previously subscribed to the text

messaging subscription service, the installment of the text messaging subscription delivered based on determining that the mobile device has not previously requested delivery of the text messaging subscription.”

169. In my opinion, the combination renders this claim obvious. *See* above, claim 21.

CLAIM 30

“30. The method of claim 24, further comprising: determining that the mobile device has previously subscribed to the text messaging subscription service; and sending a message to the mobile device indicating that the mobile device is already subscribed to the text messaging subscription service.”

170. In my opinion, the combination renders this claim obvious. *See* above, claim 22.

Ground 2. Claims 15-30 were obvious as in Ground 1, in further view of *Ad Serving Technology*

In my opinion, claims 15-30 were obvious as in Ground 1, in further view of the textbook Cristal, G. “*Ad Serving Technology, Understand the marketing revelation that commercialized the internet*” (“**Ad Serving Technology**”)(Ex. 1012).

171. *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.

A. Overview of the Ground

172. This ground challenges the claims at-issue under ground 1, and additionally, claim 20. Claim 20 requires the “means for displaying” to be configured to execute certain business rules to decide whether to display the ad. Agrawal (via the incorporated ’900 application) teaches applying business rules to determine whether promotional messages should be displayed, but does not specifically teach that the business rules are executed in means for displaying (Agrawal’s code). *Ad Serving Technology*, however, teaches that ad tags were well-known to contain JavaScript that could determine whether or not to display the ad, and thus motivates the combination. (Ex. 1012, p. 054 (referring to added page numbers)).

173. The combination will be further explained in the claim mapping section below.

B. Analogous Art

174. *Ad Serving Technology* is analogous art because it is in the same field as the ’897 patent and is reasonably related to the problems facing the named inventors. (Ex. 1012, pp. 027-028).

C. Claim Mapping

CLAIM 20

“20. The non-transitory processor-readable medium of claim 15, wherein the means for displaying includes code configured to be executed the mobile device to: determine that the mobile device satisfies a client business rule; and cause the mobile device to display the invitation to subscribe to a text messaging subscription service via the first application in response to determining that the mobile device satisfies the client business rule.”

175. Agrawal (via the incorporated '900 application) already teaches applying business rules to determine whether promotional messages should be displayed. (Ex. 1006, 015:28-016:24, 037:9-32, 039:19-27). The business rules can include, for example, whether the user has been over-exposed to an ad (by seeing it too many times within a recent time period) (Ex. 1006, 039:19-27), using a blacklist of advertiser URLs or keywords (037:9-32), the content of a page, the general content of a publisher, user characteristics (e.g., language or location), the appearance, location or size of an ad on a page (015:28-016:24).

176. It would have been obvious to enforce these rules by using **code** in the **means for displaying** of Agrawal **to cause the first application to display the promotional message in response to a business rule being met**. As discussed above under Ground 1, claim 15, element [15b], Agrawal's integration tag is implemented using JavaScript. JavaScript, as the textbook *Ad Serving Technology* states, was a powerful scripting language that was commonly used in ad tags (e.g.,

integration tags). (Ex. 1012, p. 053 including Fig. 1.5 and caption). In an ad tag, JavaScript was used to make decisions about which ads to display:

“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.**”

(Ex. 1012, p. 054, *see also* pp. 051-054)(Emphasis added).

177. Based on the fact that Agrawal teaches business rules that, when met, allow promotional messages (ads) to be displayed, that Agrawal teaches using means for displaying comprising JavaScript, and that it was well-known in the relevant time to use JavaScript to “make decisions about what ad to display in real time”, it would have been obvious to implement Agrawal’s integration tags to implement business rules such as those described in the incorporated ’900 application. A PHOSITA would have had a reasonable expectation of success in programming the well-known and widely used JavaScript language to implement Agrawal’s business rules in this manner. .

XI. SECONDARY CONSIDERATIONS

178. 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, ¶¶0029, 0034, 0048, 0007)(Ex. 1007, 10:1-5, 9:48-55, 13:12-18)(Ex. 1013, ¶¶0039-0042)(Ex. 1017, ¶¶0026, 0069, 0088).

179. I understand that in a co-pending district court litigation (where the Patent Owner was responding to an argument that the ’897 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.

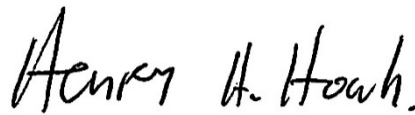
180. Overall, it is my opinion that there is not significant evidence of secondary considerations, much less evidence that would change my opinion about the strong evidence of obviousness over the prior art as discussed above.

I. OATH

181. 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 19, 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