

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. 11,416,887

DECLARATION OF DR. HENRY HOUH

TABLE OF CONTENTS

TABLE OF CONTENTS.....	2
I. ENGAGEMENT AND COMPENSATION	5
II. SUMMARY OF OPINIONS	7
III. QUALIFICATIONS	8
IV. MY UNDERSTANDING OF THE RELEVANT LAW	9
A. Claim Construction.....	9
B. Anticipation	10
C. Obviousness.....	11
V. TECHNICAL INTRODUCTION	14
Two-tap example: U.S. Pat. No. 8,532,283 to Haggerty (Ex. 1007).....	20
Two-tap example: U.S. Pat. App. Pub. 2015/0178784 to Oliver (Ex. 1011).....	20
Two-tap example: U.S. Pat. App. Pub. 2015/0142568 To Hsu (Ex. 1013).....	22
Two-tap example: U.S. Pat. App. Pub. 2009/0247140 to Gupta (Ex. 1017).....	23
Deeplinking technologies were well-established	25
Basic Ad-Network Technology	29
Basic Web Site Technology	30
VI. CLAIMS OF the '887 PATENT	41
VII. RELEVANT TIMEFRAME FOR DETERMINING OBVIOUSNESS	43
VIII. LEVEL OF ORDINARY SKILL IN the ART	43

IX.	DETAILED EXPLANATION OF THE REASONS FOR UNPATENTABILITY	44
Ground 1.	Claims 1-2, and 13-14 were obvious over Haggerty in view of Demsey and Khanna.....	44
A.	Overview of the Ground.....	44
1.	Overview of Haggerty (two-tap method).....	45
2.	Overview of Demsey (ad network).....	49
3.	Overview of Khanna (mobile app deeplinking)	51
B.	Rationale (Motivation) Supporting Obviousness.....	53
C.	<i>Graham</i> Factors	57
D.	Reasonable Expectation of Success	57
E.	Analogous Art	58
F.	Claim Mapping	59
Ground 2.	Claims 1-4, 6-10, 13-14, and 19 would have been obvious as in Ground 1, in further view of Rakavy.	82
A.	Overview of the Ground and Rationale for the Combination.....	83
B.	Reasonable Expectation of Success	84
C.	Analogous Art	84
D.	Claim Mapping	84
Ground 3.	Claims 1-2, 5, 13-18, and 20 were obvious as in Ground 1, in further view of <i>Ad Serving Technology</i>	90
A.	Overview of the Ground.....	90
B.	Reasonable Expectation of Success	91
C.	Analogous Art	91
D.	Claim Mapping	91

Ground 4.	Claims 1-2, 11-14, and 20 were obvious as in Ground 1, in further view of Molinet.	95
A.	Overview of the Ground.....	96
1.	Overview of Molinet.....	96
B.	Reasonable Expectation of Success	97
C.	Analogous Art	97
D.	Claim Mapping.....	97
X.	SECONDARY CONSIDERATIONS	102
XI.	OATH	103

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 11,416,887 (the ’887 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. 11,416,887 (“ the ’887 patent ”).
1004	File History of U.S. App. Ser. No. 17/669,114 (issued as the ’887 patent).
1005	U.S. Pat. App. Pub. 2014/0379482 (“ Demsey ”).
1006	U.S. Pat. App. Pub. 2013/0111328 (“ Khanna ”).
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 ”).
1014	Plaintiff’s Answering Brief in Opposition to Defendant’s Motion to Dismiss Complaint Pursuant to Fed. R. Civ. P. 12(B)(6) in <i>Attentive Mobile Inc. v. Stodge Inc. d/b/a Postscript</i> , Case, No. 1-23-cv-00087 (D. Del., Filed March. 15, 2023).
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 ”).
1018	U.S. Pat. No. 5,913,040 (“ Rakavy ”).
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
1029	U.S. Pat. No. 7,136,875 (" Anderson ").

II. SUMMARY OF OPINIONS

4. It is my opinion that:

- Claims 1-2, and 13-14 were obvious over Haggerty in view of Demsey and Khanna.
- Claims 1-4, 6-10, 13-14, and 19 were obvious as in Ground 1, in further view of Rakavy.
- Claims 1-2, 5, 13-18, and 20 were obvious as in Ground 1, in further view of Ad Serving Technology.
- Claims 1-2, 11-14, and 20 were obvious as in Ground 1, in further view of Molinet.

III. QUALIFICATIONS

5. I hold a Ph.D. in electrical engineering and computer science from the Massachusetts Institute of Technology (“MIT”), awarded in 1998. I also hold a Master’s Degree in electrical engineering and computer science from MIT, awarded in 1991, and two Bachelor’s degrees (the first in electrical engineering 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 term 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 ¶67, 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 ¶68 below.

17. The relevant time frame for obviousness, the “time the invention was made”, is discussed in ¶67, 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 ’887 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 ’887 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 ’887 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, 7:19-29).

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 →

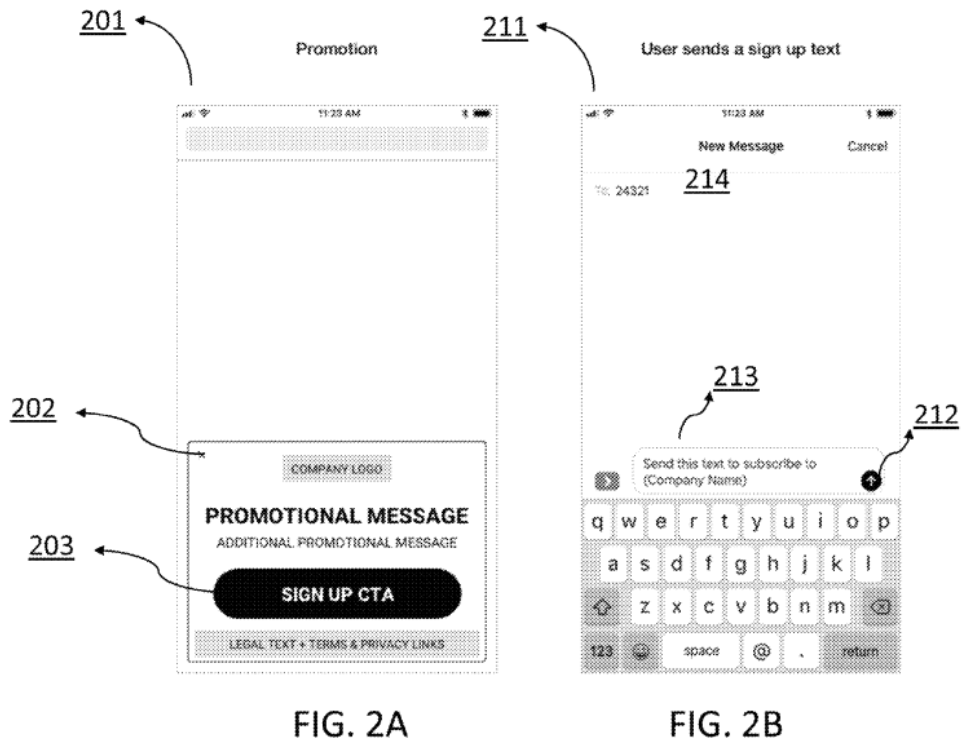
The following is advertised:

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 '887 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 '887 patent.

27. The '887 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, 6:66-7:8). Clicking the link from a mobile device causes the device’s text-messaging app to open. (Ex. 1001, 6:66-7:2). Upon opening, the text-messaging app presents the user with a pre-filled text message. (Ex. 1001, 7:13-17). 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, 7:17-29). 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 '887 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, 7:30-65). 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, 6:66-8:5). Because only two user actions are required, the '887 patent refers to this approach as a “two-tap” approach. (Ex. 1001, 3:3-7, 7:25-30).

29. I understand that the Patent Owner considered this “two-tap” method to be the point of novelty for the ’887 patent. (Ex. 1014, p. 2). The ’887 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, 1:36-47). 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. No. 8,532,283 to Haggerty (Ex. 1007)

31. Haggerty, which is the subject of Ground 1, below, teaches a two-tap method for SMS marketing. I provide a more detailed overview of Haggerty in ¶¶72-76, below. But briefly, in Haggerty, a user can view an ad online with an embedded link. (Ex. 1007, Abstract, 2:28-31). Clicking the link will cause the user's mobile device to present to the user "a pre-populated text message addressed to the destination address with the contents indicated therein" together with "a prompt for confirmation for sending the message." (Ex. 1007, 10:1-5, 11:19-21, 9:48-55, 13:12-18). 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. 2015/0178784 to Oliver (Ex. 1011)

32. Oliver 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).

33. 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).

34. 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).

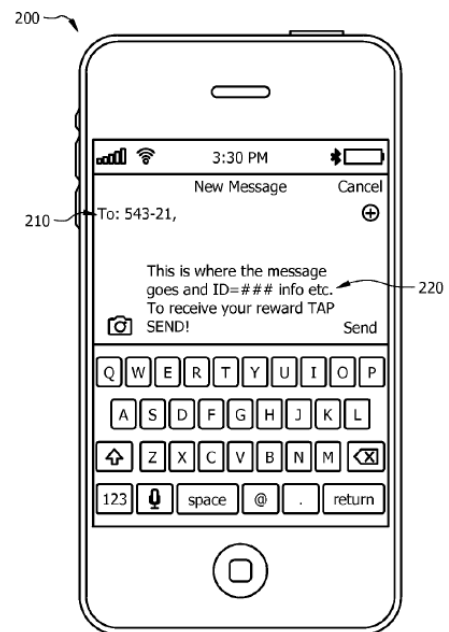


FIG. 2

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

35. 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)

36. 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).

37. 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).

38. 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

39. 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. 1006, ¶0030).

40. 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).

41. 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.”).

42. 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).

43. *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.

44. 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

45. The claims of the '887 patent also pair the known two-tap business method to standard Internet advertising technology. The standard advertising technology used by the '887 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).

46. *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.

47. 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.

48. 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

49. 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.

50. 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.

51. 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:

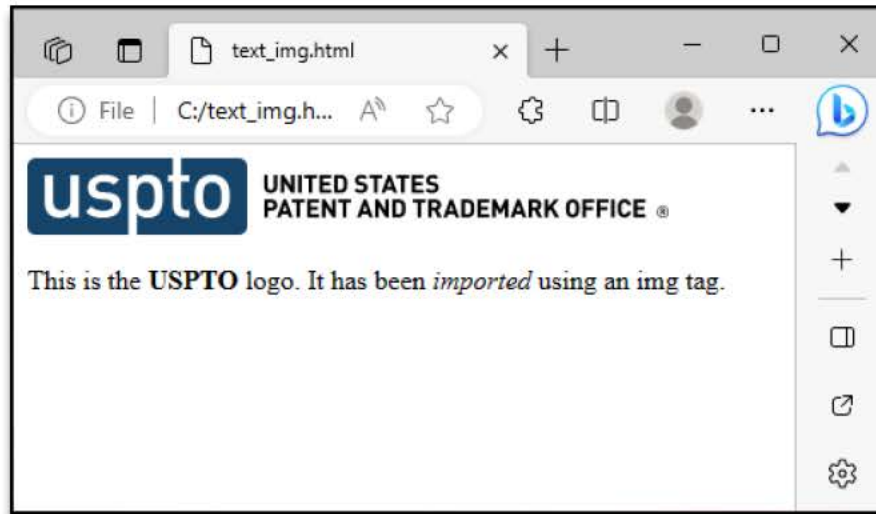
```

```

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:



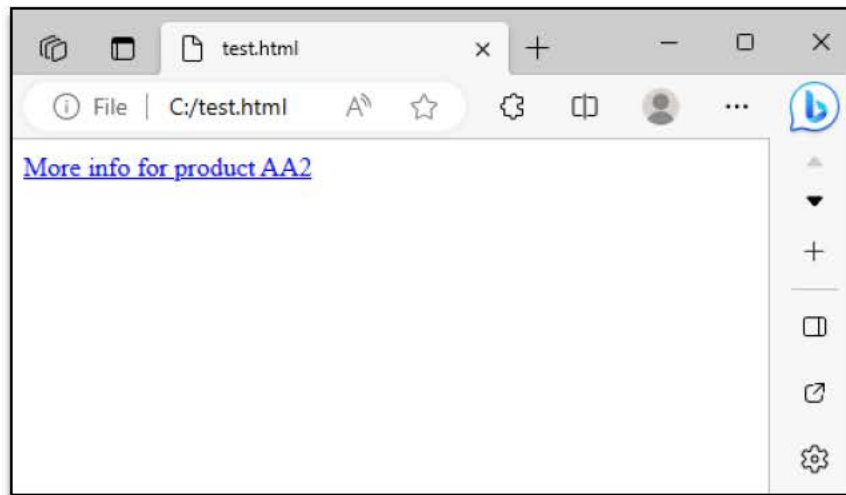
52. In the code above, the img tag causes the browser to issue a request to a USPTO Web server to retrieve the USPTO logo. The `<br>` tag inserts a line break. The `<b>` and `</b>` tags indicate the start and end of bold text. The `<i>` and `</i>` tags the start and end of italic text. To view all the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

53. 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.

54. 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:



55. 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.

56. 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.

57. 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).

58. 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).

59. 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 prior-art U.S. Pat. No. 7,136,875 (“Anderson”):

```
<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. 1029, 13:37-46).

60. 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`.

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

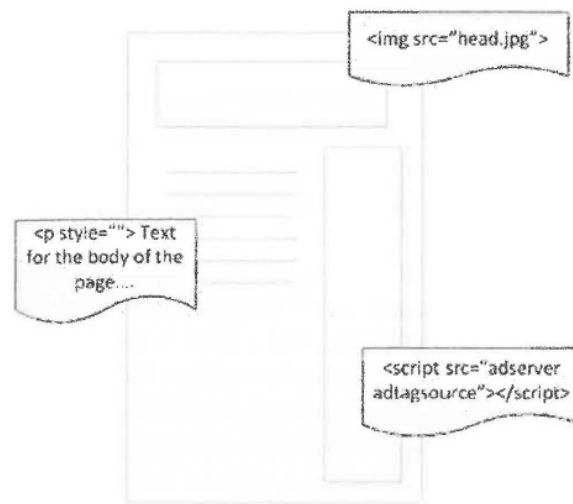


Figure 1.5: Behind the scenes a standard web page is made up of a series of HTML tags. An Ad tag is no different, which can sit on any web page to display an ad to the user. In this instance (bottom right) the HTML tag contains a reference to JavaScript.

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.

62. 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.

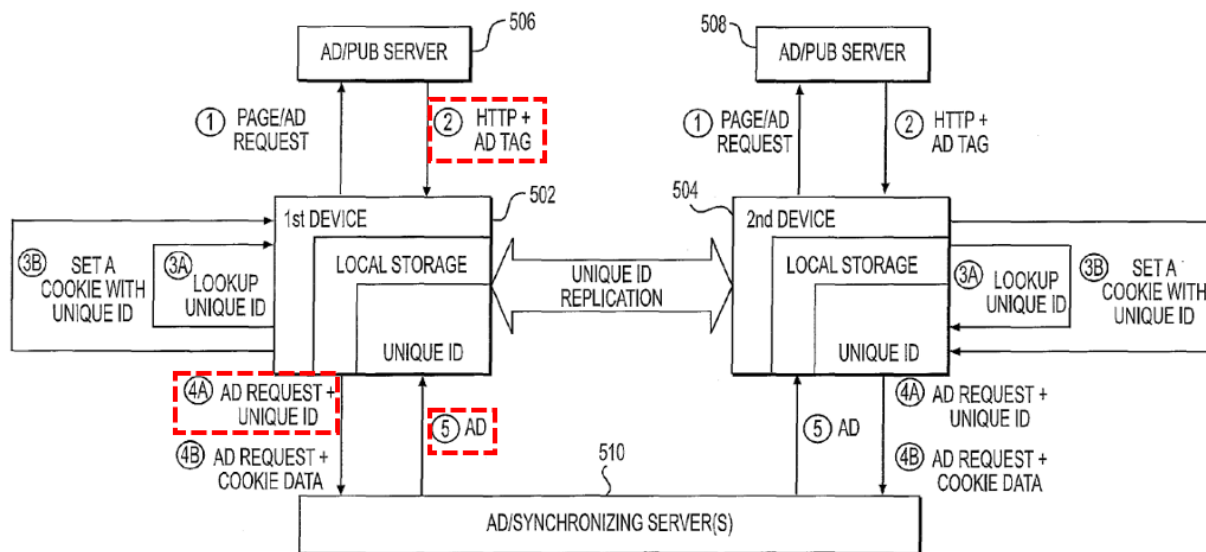


FIG. 5

(Ex. 1005, 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. 1005, ¶¶0045-0048).

63. 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.

64. The '887 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 '887 PATENT

65. The '887 patent has 20 claims with a single independent claim (claim 1). Claim 1 can be divided into roughly three parts, corresponding to (1) *standard ad-network technology* that uses a script-based ad tag to gather user data to send to a server, (2) *standard deeplinking technology* that provides a way to link to a text-messaging application, and (3) the two-tap business method. Claim 1 is shown in the table below, with added numbers (1[a] [1b], etc.):

Claim element	Category
“1[a]. At least one non-transitory processor-readable medium storing code configured to be executed by a processor of a click-to-text server, the code including instructions configured to cause the click-to-text server to:”	Preamble
“[1b] send, to a client server, an integration tag configured to be served with a webpage hosted by the client server,”	Ad-network
“[1c] the integration tag configured to cause a mobile device loading the webpage via a first application to send user data to at least one of the client server or the click-to-text server;”	Ad-network
“[1d] define and send a uniform resource identifier (URI) to the mobile device in response to the mobile device executing the integration tag, the URI deeplinking to a messaging application different from the first application and”	Deeplinking

“[1e] configured to cause the mobile device to (1) automatically transition from the first application to the messaging application in response to the mobile device detecting a user interaction with a promotional message associated with the webpage, the messaging application being different from the first application, and”	Deeplinking / Business method
“[1f] (2) automatically populate a custom message in the messaging application that includes an address associated with the click-to-text server and a message body that includes an identifier associated with at least one of the webpage or the user data”	Business method
“[1g] such that, upon detecting a single user interaction with a send button of the messaging application, the mobile device sends the custom message to the click-to-text server;”	Business method
“[1h] receive the custom message at the click-to-text server; and”	Business method
“[1i] enroll the mobile device in a promotion associated with the promotional message based on receiving the custom message and without receiving any additional information from the mobile device.”	Business method

66. In claim 1, a “click-to-text server” (the central server of an ad network) sends an “integration tag” (which can be, *e.g.*, an ad tag) to a “client server” (a publisher) for inclusion in a website. The “integration tag” is configured to send user data to the central server, and gets a response. This is standard ad-network technology. The response contains a deeplinking URI, which is simply a link in an ad that deeplinks to a mobile app. This is standard deeplinking technology. Clicking on the deeplink causes a pre-filled text-message to appear that provides consent for further marketing activities, as was well-known in the art.

VII. RELEVANT TIMEFRAME FOR DETERMINING OBVIOUSNESS

67. I understand that obviousness must be evaluated at the time the alleged invention was made. From the second page of the '887 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

68. 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. DETAILED EXPLANATION OF THE REASONS FOR UNPATENTABILITY

Ground 1. Claims 1-2, and 13-14 were obvious over Haggerty in view of Demsey and Khanna.

69. It is my opinion that claims 1-2, and 13-14 were obvious over **Haggerty** (Ex. 1007), in view of U.S. Pat. App. Pub. 2014/0379482 (“**Demsey**”)(Ex. 1005) and U.S. Pat. App. Pub. 2013/0111328 (“**Khanna**”)(Ex. 1006).

A. Overview of the Ground

70. Haggerty teaches a “two-tap” method for enrolling users in promotions, where a single user action opens a pre-filled text message, which then only needs to be sent. (Ex. 1007, 10:1-5, 9:48-55, 13:12-18). In my opinion, it would have been obvious to implement Haggerty’s business method using well-known ad-serving technology, as described in Demsey (an ad-network patent) and Khanna (a mobile app deeplinking patent).

71. Thus, in the combination, Demsey provides the standard ad-network functionality, Khanna provides the standard deeplinking functionality, and Haggerty teaches the known “two-tap” business method. Haggerty also provides a click-to-text server that can be easily extended and modified to encompass the

functions of Demsey and Khanna, although in principle, the combination could be carried out on any of the central servers of the three references.

1. Overview of Haggerty (two-tap method)

72. Haggerty teaches a “two-tap” business method similar to that of the ’887 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).

73. 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:

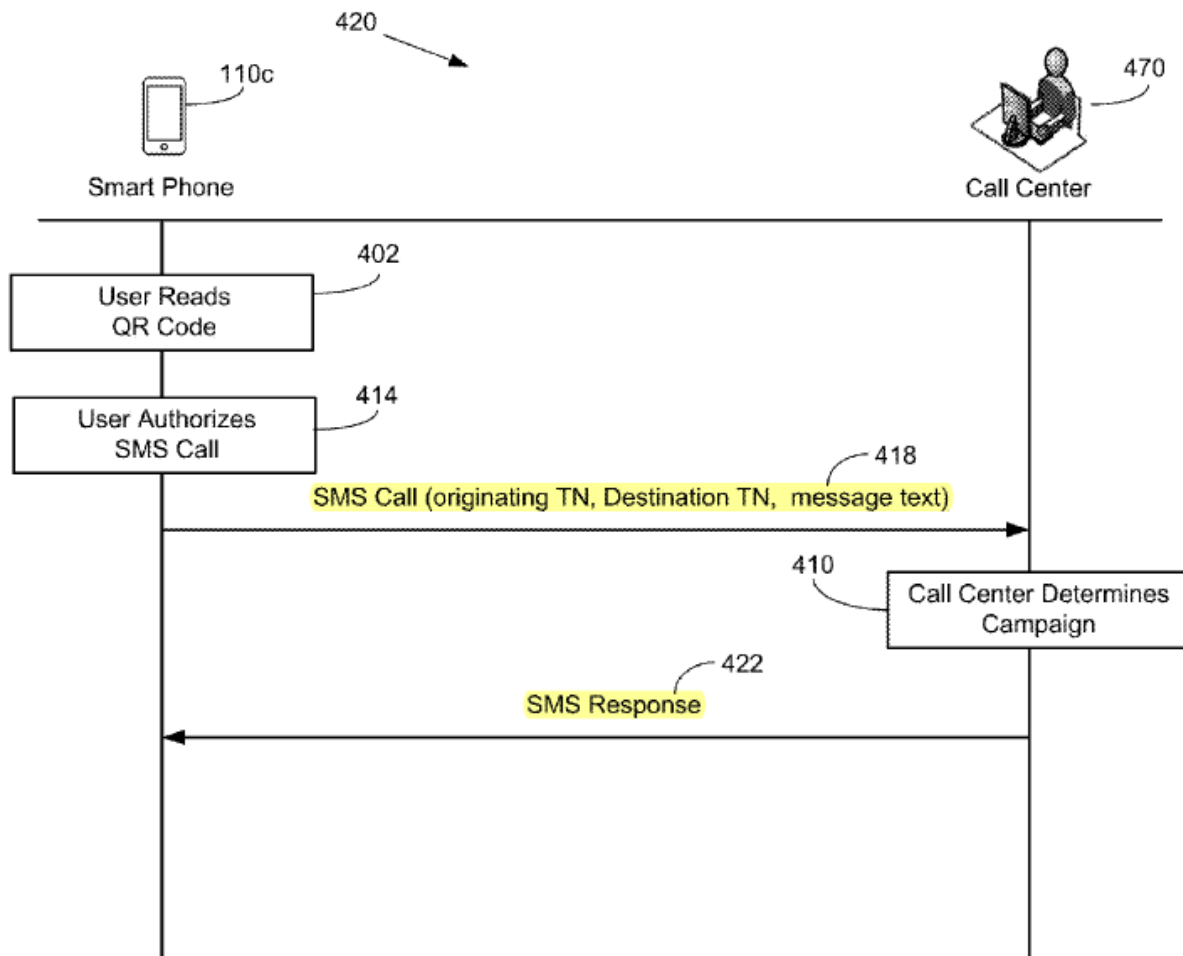
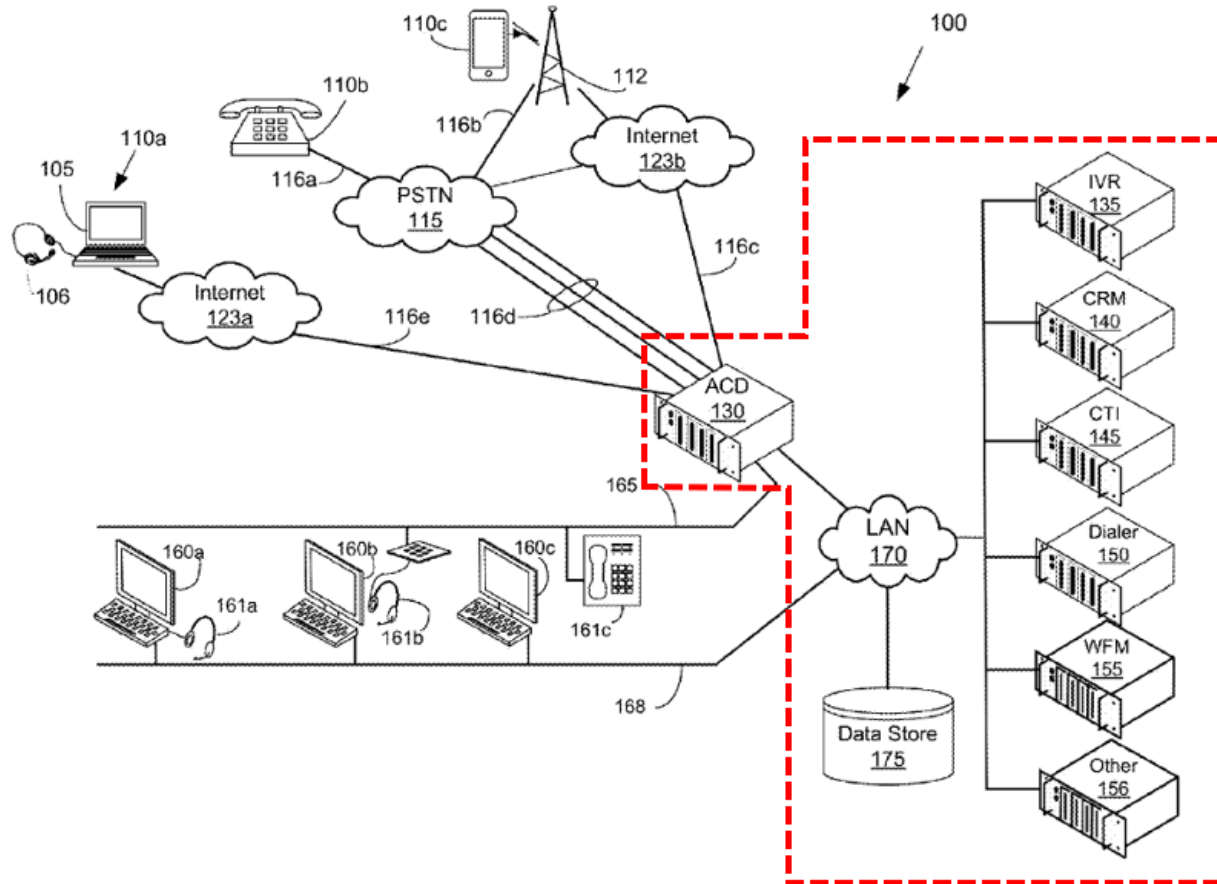


FIG. 4B

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

74. Just like the '887 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).

75. Haggerty teaches that its call center can be implemented by server technology such as shown in Fig. 1, reproduced here with a red-dashed box around the collection of individual servers that make up an overall server function:



(Ex. 1007, Fig. 1, 3:44-63, 5:45-7:62). This server function of Haggerty's call center can be modified and extended to encompass additional functions, as Haggerty teaches. (Ex. 1007, 7:34-36, 7:42-62, 27:5-42, Figs. 10, 14).

76. While Haggerty assumes advertisements with clickable links (Ex. 1007, 2:28-31), it does not describe the technology used to serve those ads. This technology, however, was standard in the relevant timeframe, as described in **Demsey**.

2. Overview of Demsey (ad network)

77. Demsey is an ad-network system that has a central server, advertisers, publishers, and users who download webpages with ads, as shown in Fig. 2, reproduced here:

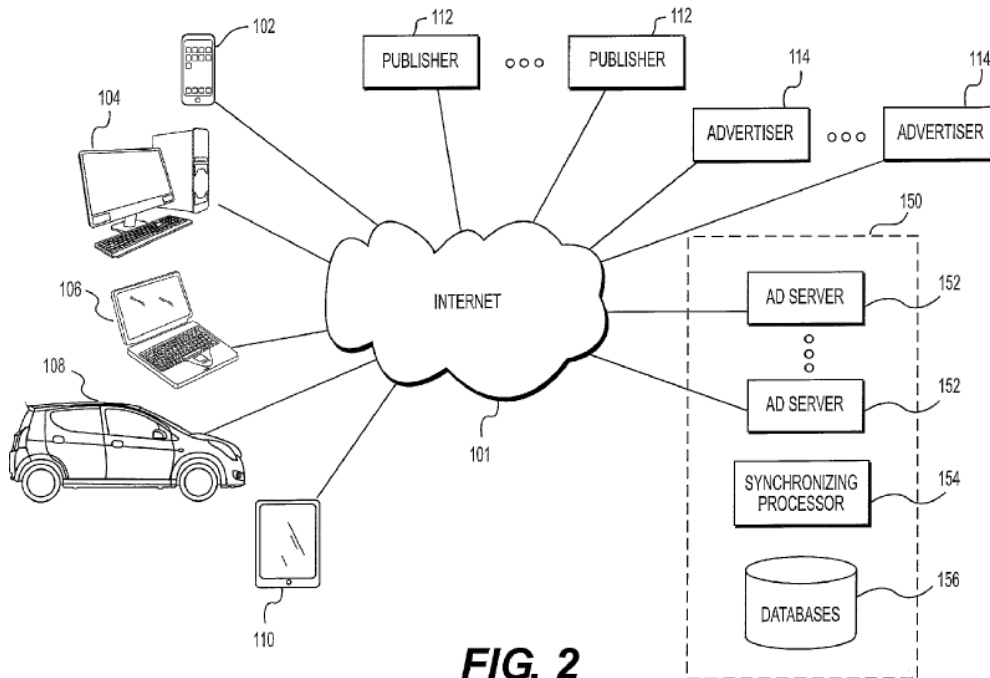


FIG. 2

(Ex. 1005, Fig. 2, ¶¶0024-0025). In Fig. 2, the central server is the synchronizing server system 150,¹ advertisers are represented by servers 114, publishers (the owners of Web sites that have content) are represented by servers 112, and user

¹ Demsey sometimes also uses the term “synchronization server” (e.g. Ex. 1005, ¶0028). I do not believe there is a difference between “synchronizing” and “synchronization”.

devices are shown by devices 102 through 110. (Ex. 1005, ¶¶0024-0025).

78. Demsey sends “ad tags” to a publisher for inclusion in webpages. (Ex. 1005, ¶0005). When a webpage is downloaded to a user device, the ad tag (which uses JavaScript, discussed above) executes and requests that an ad be sent back to the user’s device for display. (Ex. 1005, ¶¶0045-0048, Fig. 5). This is shown, for example, in Fig. 5 of Demsey, reproduced below with three added red-dashed boxes:

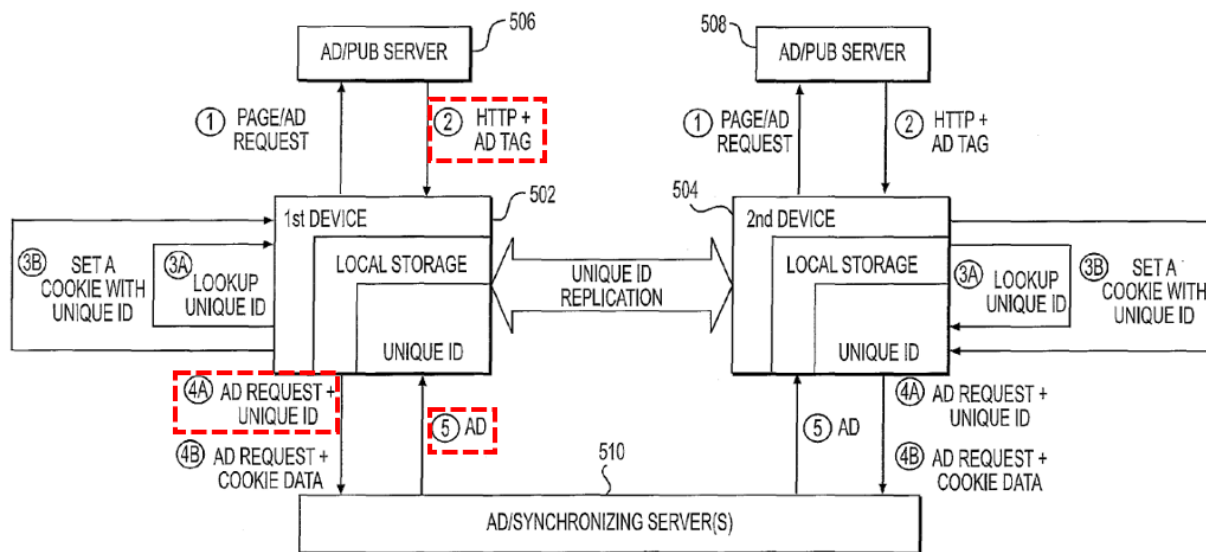


FIG. 5

(Ex. 1005, Fig. 5). As shown in Fig. 5, a user device 502 (left side, middle) downloads a webpage at action (2) (shown by a red-dashed box at the top). The webpage contains an “AD TAG”. (Ex. 1005, ¶0045). The “AD TAG”, when executed upon loading of the Web page by a user browser, causes the user’s device 502 to issue an “AD REQUEST” at action (4A) (red-dashed box, lower left) to the

AD/SYNCHRONIZING SERVER(S) 510. (Ex. 1005, ¶0045). Along with the ad request, the user's device sends a unique ID. (Ex. 1005, ¶0045). When the server 510 receives the ad request, the server chooses an ad, and sends the ad back to the user device for incorporation into the webpage at action (5) (red-dashed box). (Ex. 1005, ¶¶0045, 0041).

79. Demsey's particular method involved assigning each user a unique identifier (like a number or code), and having the user device store the identifier in a cookie or other local storage. (Ex. 1005, ¶¶0009, 0039, 0045). This unique identifier was retrieved by the ad tag, and sent back to a central ad server, so that the central ad server could ascertain which user was requesting the ad. (Ex. 1005, ¶0045). Demsey also teaches providing "instance identifiers", which are identifiers for a *particular device* of a given user, understanding that a user could view webpages from multiple devices. (Ex. 1005, ¶¶0022, 0036).

3. Overview of Khanna (mobile app deeplinking)

80. A natural extension of Demsey's ad-network is provided by **Khanna**. Khanna teaches a way to allow mobile app deeplinking using links in ads that are served by an ad network (like Demsey's). (Ex. 1006, ¶¶0031-0044). Like Demsey's system, Khanna's system also has a central server, called the Advertisement Management System 110 ("AMS 110"), shown in Fig. 1, reproduced here:

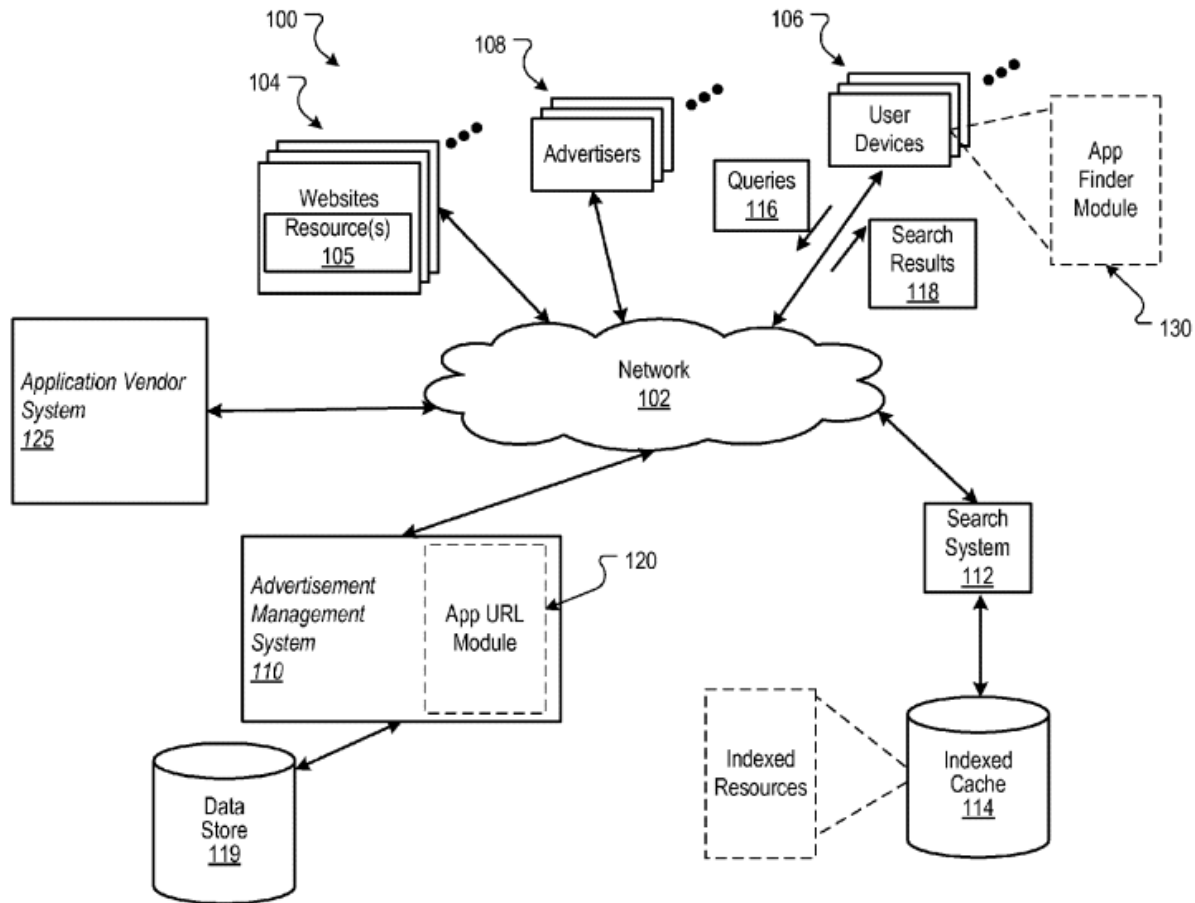


Fig. 1

(Ex. 1006, Fig. 1, ¶0031). The system also has user devices 106 (*e.g.* smartphones), which can download “resources” 105 (*e.g.* websites). (Ex. 1006, ¶¶0033-0035, 0103). The user devices correspond to the user devices 102-110 of Demsey. (Ex. 1005, ¶0023). The websites have advertisement slots that are intended to be filled with ads. (Ex. 1006, ¶0035). This corresponds to Demsey’s ad tags. (Ex. 1005, ¶¶0005, 0045, 0047). When a user views a webpage on a browser, the webpage—just as in Demsey—makes a request to the AMS 110 for advertisements to fill the

ad slots. (Ex. 1006, ¶0038). The AMS 110 corresponds to the synchronizing server system 150 of Demsey. (Ex. 1005, ¶0024). And just as in Demsey, the AMS 110 responds to an ad request by sending back a response with the ads. (Ex. 1006, ¶0040).

81. To this basic ad-network system, Khanna adds mobile app deeplinking capability. (Ex. 1006, Abstract, ¶0025). Specifically, Khanna allows the ads to have links that specify a target mobile app, and a target product. (Ex. 1006, ¶0025). The links are generated when the AMS 110 selects an ad. (Ex. 1006, ¶0044). The AMS 110 uses an application URL module 120 (shown above in Fig. 1) to generate or modify the linked URL in the ad such that it will open a target mobile app to a desired starting point relating to the desired product (note that “product” is understood broadly, and could be, for example, information about a song and artist, ¶0051). (Ex. 1006, ¶0044). Khanna also provides code on the client device to check whether the target application is installed (¶¶0028, 0067), and if so, to carry out the deeplinking function (¶¶0046-0047).

B. Rationale (Motivation) Supporting Obviousness

82. In my opinion, it would have been obvious use Khanna’s techniques to enhance Demsey’s system, and using that system, to implement the method of Haggerty for Internet advertising. I incorporate my opinions above, particularly relating to the Background of the technology prior to May 2017.

83. **First**, it was obvious to use Demsey and Khanna together. Both references are intended for use with web-based advertising. (Ex. 1005, Abstract)(Ex. 1006, ¶¶0003, 0038-0039). Demsey describes an ad-network server that provides ads to webpages that can be read on mobile devices. (Ex. 1005, ¶¶0005, 0023). Khanna picks up at this point, operating on ads, to be incorporated into a webpage for reading on a mobile device, that are served from an ad-network server based on an ad request. (Ex. 1006, ¶¶0031, 0038-0039). Given the natural technological relationship between the two references, it would have been obvious to a PHOSITA to extend Demsey's capability with Khanna's.

84. Additionally, a PHOSITA with Demsey's system would have desired the mobile-app-deeplinking technology of Khanna, because Khanna states that "[a]llowing publishers or advertisers to create content items that deep-link directly to the content within their applications may lead to better user experience as well as create opportunities for the publisher to drive more business." (Ex. 1006, ¶0025). Similarly, a PHOSITA considering Khanna's system would have desired to use the system of Demsey, in order to be able to serve advertisements and to track users and user interactions with advertisements (Ex. 1005, ¶0021), as expressly desired by Khanna (Ex. 1006, ¶0043).

85. **Second**, it was obvious to implement Haggerty's method using the system of Demsey and Khanna. Haggerty's method would have appealed to

businesses that sell goods and services and that seek new potential customers. (Ex. 1007, 1:19-26, 1:53-62). 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 from people interested in the business's offerings. (Ex. 1007, 1:19-26, 1:53-62, 17:4-18, 18:46-59).

86. A PHOSITA would have found the combination of Haggerty with the system of Demsey/Khanna advantageous. Haggerty teaches a two-step sequence for acquiring new leads, in which a potential customer first interacts with a hyperlink, and second, clicks "send" on a pre-filled text message. (Ex. 1007, 10:1-5, 1:53-62, 14:41-49, 1:36-62, 2:1-5, 8:51-56, 10:59-11:21, 16:26-31, Figs. 3C, 4G). Haggerty further teaches that its links can be incorporated into electronic advertising. (Ex. 1007, 14:41-49). A PHOSITA would have found obvious that the hyperlinks of Haggerty that open a text-messaging application to a particular (pre-populated) state are exactly the kind of function that Khanna's deeplinks provide, and would have been aware of the standard URI protocols available to deeplink to SMS applications, as discussed in the Introduction. A PHOSITA would have found it advantageous to use Internet ad systems like Demsey/Khanna in order to increase the reach of Haggerty's marketing efforts, to better target advertisements to particular users (as expressly taught by Demsey), and to account

for conversions of deeplinks (as expressly taught by Khanna). (Ex. 1005, ¶¶0002, 0021)(Ex. 1006, ¶¶0043, 0051, 0066).

87. In particular, it would have been obvious to carry out the server-based software techniques of Demsey and Khanna at Haggerty's call center server. Haggerty teaches that its call center functionality can be expanded with other server-based functions. (Ex. 1007, 7:34-36). Each of Demsey and Khanna, moreover, provides functionality with a central server (like Haggerty's call-center), such as Demsey's Synchronizing Server 150 and Khanna's Search System 100 (Ex. 1005, ¶¶0021-0028, 0045-0048)(Ex. 1006, ¶¶0031, 0038-0039). Both Demsey and Khanna also express flexibility in the arrangement of the software, user and server devices that make up their systems. (Ex. 1005, ¶¶0026-0032)(Ex. 1006, ¶¶0108, 0120-0122). It would have been obvious to combine the functionalities of these references into a single "server" as that term is understood in the '887 patent (Ex. 1001, 5:4-6:4) 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 Demsey, Khanna and Haggerty.

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

C. *Graham* Factors

89. 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 ¶68.

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

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

92. **Secondary considerations** are discussed in ¶¶175-177, below.

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

D. Reasonable Expectation of Success

94. 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 '887 patent) are implemented in software, are compatible, and were predictable in the relevant timeframe (May 2017). A PHOSITA would have been able to make any necessary modifications to implement the Ground, and in particular would have been able to implement Khanna's software techniques in Demsey's system, and then to carry out Haggerty'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 AMS 110 in Khanna, the synchronizing server 150 in Demsey, and the call center server function in Haggerty, 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.* Haggerty's call-center server function.

E. Analogous Art

95. Haggerty, Demsey, and Khanna are analogous art because they are in the same field as the '887 patent (Internet and mobile device technology). (Ex. 1001, 1:30-35)(Ex. 1007, Abstract)(Ex. 1005, Abstract)(Ex. 1006, Abstract).

Furthermore, the methods of Haggerty, Demsey, and Khanna 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, 1:48-52).

F. Claim Mapping

96. 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 dependent claim combination also renders obvious the claims from which the dependent claim depends.

CLAIM 1

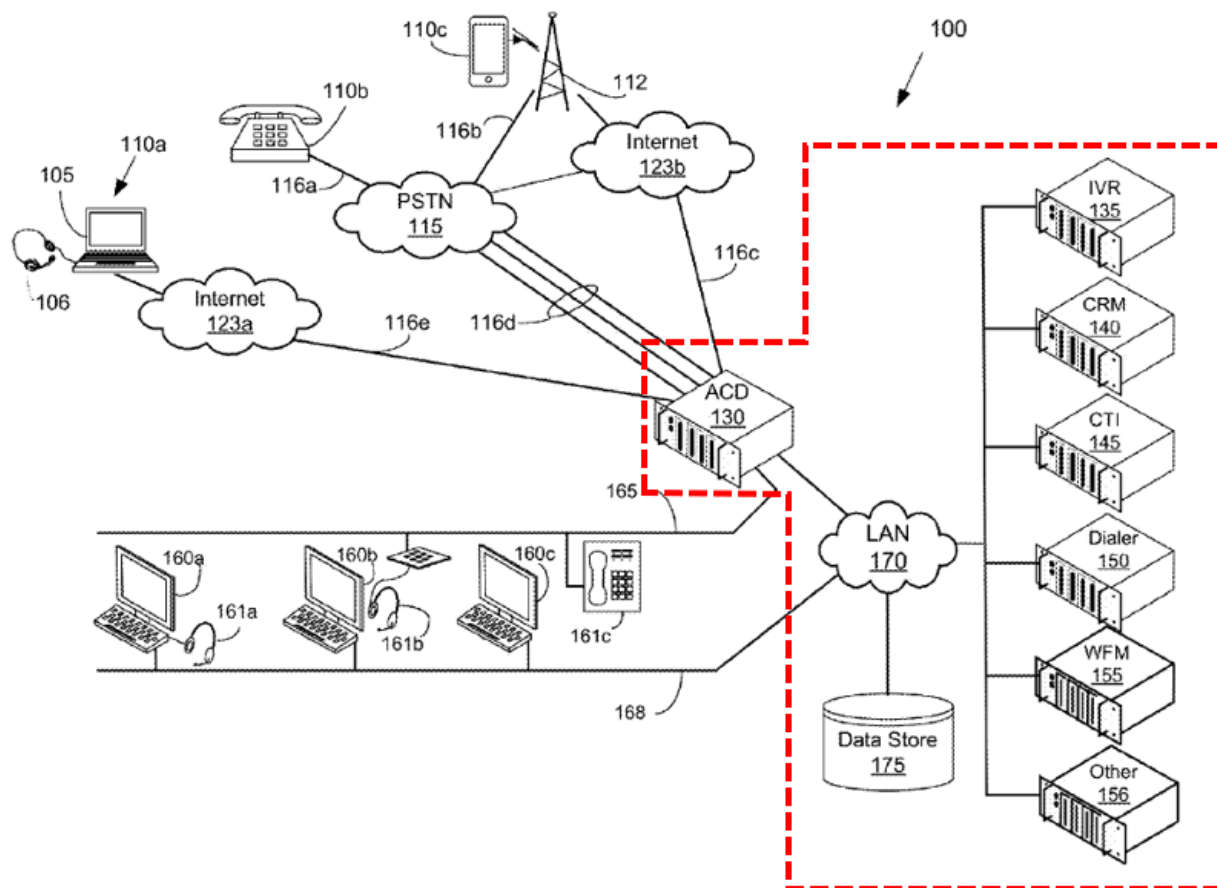
“1[a]. At least one non-transitory processor-readable medium storing code configured to be executed by a processor of a click-to-text server, the code including instructions configured to cause the click-to-text server to:”

97. It is my opinion that the combination discloses and renders obvious claim 1 of the '887 patent.

(a) non-transitory processor-readable medium storing code configured to be executed by a processor of a click-to-text server

98. Haggerty teaches a **non-transitory processor-readable medium storing code configured to be executed by a processor of a click-to-text server.**

Specifically, Haggerty teaches a “call center” for carrying out its methods. (Ex. 1007, 3:44-7:62, 27:5-28:42). The call center has several computing functions 130-156 and a data store 170, each of which can be implemented in software operating on a server. (Ex. 1007, Fig. 1, 3:44-63, 5:45-7:62). These computing functions are shown in Fig. 1 of Haggerty, reproduced below, with an added red-dashed line around the **click-to-text server**:



(Ex. 1007, Fig. 1, 3:44-63, 5:45-7:62). The server is a “**click-to-text**” server because, in the combination, it performs the click-to-text functionality required by this claim in the same manner as the click-to-text server of the ’887 patent, as further

described under elements [1b]-[1i], below. Furthermore, in Haggerty, the server receives the pre-populated text message and responds to it, as discussed below under limitations [1g]-[1i].

99. Haggerty teaches that the various computing functions can be implemented in software running on a server (Ex. 1007, 7:42-62), which can be, for example, a single hardware platform (Ex. 1007, 7:52-56, 27:5-42, Figs. 10, 14). Haggerty further teaches that the functions of the **click-to-text server** (as identified above) may be carried out using a **non-transitory processor-readable medium storing code configured to be executed by a processor**. (Ex. 1007, 28:16-42). I note the '887 patent's teachings regarding the hardware and software arrangement of the click-to-text server. (Ex. 1001, 5:4-6:4).

100. In general, a PHOSITA would have understood that the methods of the prior art were software-based methods that have instructions to be carried out by a processor, and that the instructions needed to be stored in a non-transitory processor-readable medium.

(b) code including instructions configured to cause the click-to-text server to

101. For the reasons described above, I am of the opinion that it would have been obvious to combine certain functions of Demsey and Khanna with Haggerty's call center, and it thus would have been obvious, as taught by Haggerty, to arrange the **code** of the click-to-text server to **include instructions**

configured to cause the click-to-text server to perform the functions of the click-to-text server recited in each of the claim limitations below per Haggerty’s express suggestion that its functions can be carried out using “computer program products comprising memory storing instructions causing a processor to perform the operations associated with the above technologies.” (Ex. 1007, 28:16-42). In particular, it would have been obvious to implement, in a single server function such as the one in Haggerty’s call center, an ad-network technology described in Demsey, that is, to distribute ad tags in a known fashion to publishers, to configure the ad tags to collect user data as described in Demsey, to have the ad server respond to an ad request with an ad as described in Demsey, to allow mobile device deeplinking from the ads as described in Khanna, and to use such deeplinking for click-to-text activities as described in Haggerty.

“[1b] send, to a client server, an integration tag configured to be served with a webpage hosted by the client server,”

102. Demsey discloses **sending, to a client server, an integration tag configured to be served with a webpage hosted by the client server.** (Ex. 1005, ¶0005). The integration tags used in Demsey are called “ad tags”. (Ex. 1005, ¶0005). Demsey states:

“By way of background, ad networks typically have hundreds or thousands of partners (often referred to as ‘**publishers**’) that **publish online content along with ‘ad tags’ provided by their partner ad**

network. For example, each of a news website (www.news.com), a blog (www.blog.com), and an informational website (www.info.com), among others, may engage an ad network to efficiently and profitably advertise to those websites' viewers. The **ad network provides each of those sites with an ad tag** having the ad network's domain, e.g., 'adnetwork.com'"

(Ex. 1005, ¶0005)(Emphasis added)(see also ¶¶0030, 0045, 0047). A PHOSITA reading this disclosure would understand and find obvious that Demsey is describing sending ad tags from its synchronizing server to publisher servers, such that the publisher servers incorporate the ad tags into Web sites that get served to users in response to user requests.

103. Ad tags are “**integration tags**” because they are code (JavaScript) (Ex. 1005, ¶¶0045, 0047) representing integrator instructions, meaning that they are a script that caused the user mobile device to send a request to the central server and receive from the central server promotional messages associated with the publisher client. (*Compare*, e.g. Ex. 1001, 5:35-42). In the quote above, the “publishers” are publishers of websites from publisher servers 112 (as shown in Fig. 2, e.g.). (Ex. 1005, ¶¶0025-0026, 0005, 0045). Because the publishers partner with ad networks for ads, they are “clients”, and their servers are **client servers**. (Ex. 1005, ¶¶0005-0006).

104. As discussed above, the ad tags are **configured to be served with a webpage hosted by the client server**. (Ex. 1005, ¶¶0005, 0045, 0047). This is shown, *e.g.*, in Fig. 5, with an added red-dashed box to show the service of a web page with an ad tag from a publisher server 506:

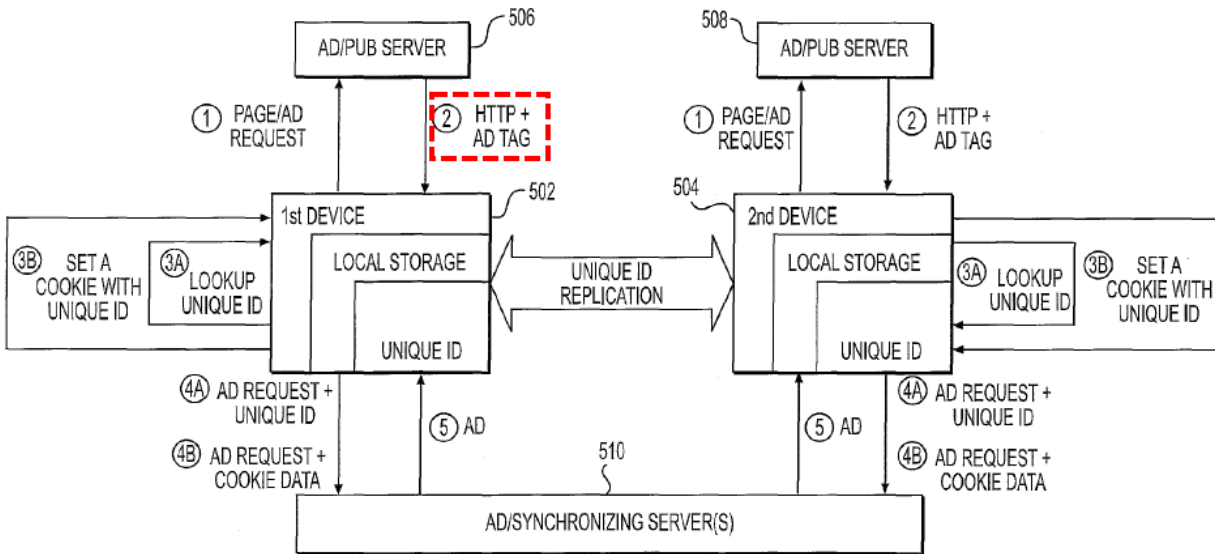


FIG. 5

(Ex. 1005, Fig. 5, ¶¶0045, 0047). The Web page at (2) is served in response to a page request (1) issued by a user device. Note that Fig. 5 combines potential third-party requests issued by the user device, and also shows possible ad requests to third parties at (1), although the primary ad request with which Demsey is concerned is issued to the synchronizing servers at (4A).

105. Demsey explains that its **integration tags** can be made of up of scripts, like JavaScript. (Ex. 1005, ¶¶0045, 0047)(“The user’s first device 502 may

execute the ad tag, such as by executing any type of client side script (e.g., JavaScript)"). This was typical for ad tags in the relevant timeframe.

106. I am further of the opinion that it would have been obvious, based on Demsey's disclosures, to implement the **sending** of the integration tag from a central server (in the combination, the **click-to-text server** as defined above under limitation [1a]) **to the client server**. Demsey already teaches that "[t]he ad network provides each of those sites with an ad tag" (Ex. 1005, ¶0005). It would have been apparent and obvious to a PHOSITA that providing ad tags to a site means sending the ad tag **to the client server** that serves the webpage, because the ad tag will be integrated *into* the webpage, and thus served with the webpage from that same server. (Ex. 1005, ¶¶0045, 0047, 0005, Fig. 5). It was further apparent and obvious to **send the integration tag** (e.g. ad tag) **from the click-to-text server**. Demsey discloses that publisher servers communicate with the ad network via the server of the ad network (the "synchronizing server" in Demsey)(Ex. 1005, ¶0026). The functionality of the ad network is implemented at its server (Ex. 1005, ¶0028), and thus it would have been obvious to provide ad tags from that server. Furthermore, it would have been obvious to a PHOSITA to arrange to have a central server perform the sending of the ad tags at scale. As Demsey explains, an ad network may have hundreds or thousands of partner publishers (Ex. 1005, ¶0005), and these publishers may change their websites frequently. Allowing the

publisher servers to download ad tags from the ad-network server would have facilitated the quick deployment of ad tags.

107. The ad tag must also send data and an ad request to the correct address at the ad network server (Ex. 1005, ¶¶0045, 0047, Fig. 5) and in the correct form for the ad network to use the data. For example, the ad tag I discussed above (see ¶¶59-60) from Anderson (Ex. 1029) needed to send information in a specific form (a list of key-value pairs of parameters) to a specific ad server at Google , having a specific URL (<http://pagead.google.com/ads>). Similarly, in the present combination, the central server in the combination (the click-to-text server) would have a specific URL that it needs to be addressed at, and would have had specific requirements on the form of the data to be received. As these things are not a priori known to publishers and are subject to change by the central server, the central server would have been best able to define the ad tag to send data in the *correct form* to the *correct place*.

108. Furthermore, the technique of sending over the Internet (*i.e.* downloading) a piece of code, from one server to another, was well-known in the relevant timeframe (early 2017), would have been predictable, and was one of only a few possible choices available to a PHOSITA seeking to implement Demsey's teaching that "ad network provides each of those sites with an ad tag." (Ex. 1005, ¶0005).

“[1c] the integration tag configured to cause a mobile device loading the webpage via a first application to send user data to at least one of the client server or the click-to-text server;”

(a) mobile device loading the webpage via a first application

109. In my opinion, Demsey discloses this limitation. First, Demsey discloses **mobile devices** (Ex. 1005, ¶¶0023, Figs. 1 and 2, rn 102, 110) that **load webpages** (Ex. 1005, ¶¶0005, 0045, 0047) for viewing **via a first application** (e.g. a browser) (Ex. 1005, ¶¶0005, 0045, 0047). The webpages contain integration tags (as discussed above under limitation [1b]) comprising, e.g., JavaScript. (Ex. 1005, ¶¶0005, 0045, 0047, Fig. 5). Demsey describes and renders obvious that mobile device browsers loading webpages that contain JavaScript ad tags will execute when the webpage is **loading**. (Ex. 1005, ¶¶0005, 0045, 0047). As I explain above (¶58), <script> tags containing JavaScript are executed when a page loads. (Ex. 1012, p. 054)(Ex. 1021, pp. 332, 021, 060).

(b) integration tag configured to cause...to send user data to at least one of the client server or the click-to-text server

110. Demsey renders obvious that the ad tags (a kind of **integration tag**) are **configured to cause a mobile device loading the webpage via a first application to send user data to at least one of the client server or the click-to-text server**. Specifically, the ad tags cause a unique ID to be passed to the click-to-text server (which is the central server or synchronizing server in Demsey).

Demsey states:

“As shown in FIG. 5, in one embodiment, the user’s first device 502 may (1) request a web page or ad from an ad or publisher server 506; and (2) **receive HTTP and an ad tag** from the ad or publisher server 506. The user’s first device 502 may **execute the ad tag**, such as by executing any type of client side script (e.g., **JavaScript**) to (3A) **look-up a unique ID** stored on the device. As described above, the unique ID may be stored within local storage associated with a browser of the device 502. The user’s first device 502 may then (4A) **issue an ad request to ad or synchronizing servers 510, where the request includes the looked-up unique ID.**”

(Ex. 1005, ¶0045)(Emphasis added). A PHOSITA would have understood this to mean and found obvious that Demsey’s JavaScript ad tag is causing an ad request containing a unique ID to be sent to the ad network server, which in the combination would be the **click-to-text server**. Demsey teaches that the unique ID can be an ID identifying a user, or an instance ID identifying a user in combination with a particular device. (Ex. 1005, ¶¶0035-0036). Because the unique ID identifies a user or a user and device, it is **user data**. (*Compare* Ex. 1001, 5:18-23).

111. Demsey further discloses that **integration tags can cause the mobile device to send** cookie data to at least one of the client server or the click-to-text server. (Ex. 1005, Fig. 5, rn 4B ¶¶0045, 0047). 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. Cookie data is also **user data**. (*Compare* Ex. 1001, 5:18-23).

“[1d] define and send a uniform resource identifier (URI) to the mobile device in response to the mobile device executing the integration tag, the URI deeplinking to a messaging application different from the first application and”

(a) define and send a uniform resource identifier (URI) to the mobile device in response to the mobile device executing the integration tag

112. In my opinion, Khanna teaches mobile app deeplinking from ads served from an ad server (as I discuss above in ¶¶80-81). Khanna would have been obvious to use for the reasons I discuss above in ¶¶82-88.

113. Specifically, Khanna’s Advertisement Management Server 110 (a central server like Demsey’s synchronizing server 150), receives ad requests when users browse websites (just as in Demsey). (Ex. 1006, ¶¶0035-0039). It would have been obvious, as I explain above, to implement the functionality of the central servers of Khanna and Demsey at Haggerty’s servers, as a **click-to-text server**. In response to an ad request, Khanna teaches returning an advertisement (called a “content item”). (Ex. 1006, ¶¶0031, 0035-0039). In addition, Khanna teaches that the content item (advertisement) can be augmented by **defining and sending a uniform resource identifier (URI) to the mobile device**, where the URI

deeplinks to an application different from the first application. (Ex. 1006, ¶¶0043-0044, *see also* ¶¶0025-0030). Khanna explains:

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

(Ex. 1006, ¶0044)(Emphasis added). A URL (as used in Khanna) is simply a more-specific form of a **URI**. After the formatted URI of Khanna is “generate[d] or modify[ed]” at the server, it is sent with the content item to the mobile device for display. (Ex. 1006, ¶¶0044, 0051--52, 0038-0041, Fig. 2). Specifically, Khanna’s AMS 110 receives an ad (a “content item”) request from the mobile device and selects an appropriate ad (¶¶0035-0041). AMS 110 then generates or modifies a deeplinking URI for inclusion in the ad (¶¶0044, 0051-0052), and **sends** the ad and deeplinking URI to the mobile device for display (¶¶0035-0040, 0044, 0052-0054, Fig. 2). Khanna’s URIs **deeplink to an application different from the first application**, because the first application is a browser (*see* limitation [1b]), while the second application is a specific application associated with the content item, and the browser first checks to see if the application is installed before launching it. (Ex.

1006, ¶¶0067, 0005). To transition from one application to another, Khanna teaches using “**deep-linking**” technology. (Ex. 1006, ¶¶0015, 0025-0030, 0044).

(b) the URI deeplinking to a messaging application different from the first application

114. In the combination, the URI **deeplinks to a messaging application different from the first application**. As discussed above, it would have been obvious to use the technology of Demsey and Khanna to implement the business method of Haggerty, wherein the deeplinking technology of Khanna is used to open a pre-populated text message in a text messaging application. (Ex. 1007, 9:39-44, 9:56-10:6)(Ex. 1008, p. 030, ¶¶0066-0067)². This use of URIs was well-known in the relevant timeframe, as I discuss above in ¶¶39-44. The messaging application is **different from the first application**, because the first application is a browser.

“[1e] [the URI...] configured to cause the mobile device to (1) automatically transition from the first application to the messaging application in response to the mobile device detecting a user interaction with a promotional message associated with the webpage, the messaging application being different from the first

² Ex. 1008 refers to U.S. Pat. App. Ser. No. 13/461,541 incorporated-by-reference into Haggerty at Ex. 1007, 1:7-15. Citations to Ex. 1008 are in the format {page}, {¶}, with {page} indicating the three-digit numbers added at the bottom.

application, and”

*(a) automatically transition...in response to the mobile device
detecting a user interaction with a promotional message
associated with the webpage*

115. Khanna teaches that when a user clicks on a portion of the content item (advertisement) that activates the URI (*i.e.* when the mobile device **detects a user interaction with a promotional message associated with the webpage**) the URI **causes** the mobile device **to automatically transition** from a first mobile application to a different mobile application. Khanna states:

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

(Ex. 1006, ¶0027). The “content item” is the **promotional message**. (Ex. 1006, ¶0003). It is **associated with the website** because it is within a content slot of the website. (Ex. 1006, ¶¶0003, 0035). Furthermore, Khanna teaches that the transition referred to above is **automatic** after detecting user input. (Ex. 1006, ¶¶0006, 0008, 0010, 0013-0014, 0046, 0062-0063). Khanna states:

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

(Ex. 1006, ¶0063)(Emphasis added). In my opinion, it was obvious in the relevant timeframe that a device **detects** 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.

116. It would further have been obvious to cause a user click on an advertising link in Demsey (*i.e.* a **promotional message**) to launch a text **messaging application**, as taught by Haggerty, for the reasons provided above under the section entitled “Rationale (Motivation) for the Combination”. In the combination, the app that is automatically opened is a **messaging application** per Haggerty. (Ex. 1007, 9:39-44). The messaging application is **different from the first application**, because the first application is a browser.

“[1f] (2) automatically populate a custom message in the messaging application that includes an address associated with the click-to-text server and a message body that includes an identifier associated with at least one of the webpage or the user data”

(a) automatically populate a custom message in the messaging application that includes an address associated with the click-to-text server

117. In my opinion, the combination renders this limitation obvious by carrying out Haggerty's business method using the technology of Demsey and Khanna. As explained above under elements [1d] and [1e], Khanna describes deeplink URIs for inclusion into an advertisement, where the deeplinks (when

clicked by a user) cause the mobile device to transition to a second application. (Ex. 1006, ¶¶0044-0046). Specifically, it would have been obvious to arrange a link using Khanna that would (per Haggerty) initiate “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, 10:2-5, 11:19-21, *see also* 14:41-49, 1:36-62, 2:1-5, 8:51-56, 10:59-11:21, 16:26-31, Fig. 3C, 4G). Here, the “destination address” as described in Haggerty (Ex. 1007, 10:2-5) was **an address associated with the click-to-text server**. (Ex. 1007, Fig. 4B, 12:39-67).

118. The “pre-populated text message” would be an **automatically-populated custom text message**, as taught by Haggerty (Ex. 1007, 10:2-5, 11:19-21, 14:41-49), which is enabled by the automatic transition to an application and passing of data to the application as taught by Khanna, as explained above presented to the user [1d]. (Ex. 1006, ¶¶0006, 0008, 0010, 0013-0014, 0027, 0046, 0062-0063).

(b) and a message body that includes an identifier associated with...the user data

119. Haggerty expressly teaches incorporating information about the user into the pre-populated message body stating:

“[T]he user may be prompted for authorization for the publisher 490 to use **information about the user**. Accordingly, **this information**

may be sent along in the body of the message to the publisher 490. Further, in particular embodiments, the advertisement with the embedded link may also request **information from the reader** that may be **included in the body of the message**. For example, the advertisement may request a **telephone number** at which the reader may be reached. **Those of ordinary skill in the art can envision several pieces of information that may be included along with the message** 482 sent to the publisher 490 in light of this disclosure.”

(Ex. 1007, 14:60-15:4)(Emphasis added)(*see also* 15:5-15). Information about the user is an **identifier** associated with the user, and therefore **associated with the user data**. (Ex. 1007, 14:60-15:15). For example, a telephone number of the user is a specific **identifier** of a user that is **associated with user data**, because it also serves to identify the user. Thus, the identifier is associated with the user data, because both are associated with the user.

120. Furthermore, Haggerty (in the quote above) expressly suggests including other information associated with the user in the body of the message. (Ex. 1007, 14:60-15:15)(“...Those of ordinary skill in the art can envision several pieces of information that may be included along with the message...”)(Ex. 1008, p. 031, ¶0069). It would have been obvious based on Haggerty’s express suggestion to include user-identifying information (Ex. 1007, 14:60-15:15) to incorporate information about the user that is already available, such as the unique identifier or unique instance (device) identifier from Demsey (*see* limitation [1c],

above) in the body of the text message, in order to allow a server receiving a text message to identify the user or to identify the user's device or device type for further marking activities and to be able to track the conversion of an advertisement through the sending of a text message. (Ex. 1006, ¶0042). A user ID or instance ID is **associated with the user data** because it *is* user data.

121. Khanna teaches a method to pass such data to a target application, such that the application can use the data—in this case by placing it in the body of a text message. Specifically, Khanna states that its URLs can encode information that is provided by the mobile device to the target mobile app. (Ex. 1006, ¶0061). For example, Khanna states:

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

(Ex. 1006, ¶0061)(Emphasis added). A PHOSITA would have understood this to mean that the data elements are incorporated into the deeplink URI (for example as parameters, *see* ¶60 above), such that they are configured to guide the target

application to a specific starting point, with the correct information. (Ex. 1006, ¶0044)(Ex. 1012, p. 583, 588)(Ex. 1019, p. 010).

(c) and a message body that includes an identifier associated with...the webpage

122. Haggerty also teaches including **identifiers** that are **associated with the webpage**. For example, Haggerty teaches including a product or service name (“Widget”) in the body of the text message (Ex. 1007, 9:64-10:6, Fig. 3A). The product or service name is an **identifier associated with the webpage** because it is the name of the product or service that the ad on the webpage is promoting. Where the promotion occurs on a webpage published by the seller of the product or service, the product or service name is also associated with the proprietor of the webpage of the product or service. Similarly, Haggerty teaches including an alphanumeric “campaign identifier” in the body of the text message. (Ex. 1007, 12:55-67, 15:5-15, 19:55-63, 20:39-40)(Ex. 1008, p. 033, ¶0078, p. 037, ¶0092). The campaign identifier identifies the advertising campaign, and is **associated with the webpage** because the advertisement is located on the webpage.

“[1g] such that, upon detecting a single user interaction with a send button of the messaging application, the mobile device sends the custom message to the click-to-text server;”

123. Haggerty teaches that:

“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. In other
embodiments, the message can be immediately sent.”**

(Ex. 1007, 10:1-6)(Emphasis added)(*see also* Ex. 1007, 9:39-55). In other words, there are two cases, one in which a message is sent immediately (without user input), and a second case in which “a prompt for confirmation for sending the message” is displayed first. (*Id.*). This latter case discloses and renders obvious that only a **single interaction** (*i.e.* the user’s confirmation action) need be **detected** by the mobile device before sending the message. Furthermore, it would have been obvious to have the **single interaction** be **with a send button**, because it was well-known in the relevant timeframe that mobile SMS applications had send buttons that already performed the function of confirming that a message should be sent. (*e.g.* Ex. 1011, Fig. 1, ¶0007). This would have been obvious as the most natural and user-intuitive implementation of a confirmation (because users were already used to pressing “send”) and, independently, would have been obvious as a known method for implementing a send confirmation as taught by Haggerty that could have been implemented with no unpredictable results.

124. Haggerty further teaches sending the SMS to the **click-to-text server** (as discussed under limitation 1[a]). (Ex. 1007, 19:33-40, 12:39-47, 13:9-35, Figs. 4B-4C, 18:38-42). This is shown, for example, in Fig. 7A, reproduced below, with

highlighting added to show the sending of the SMS sent to the click-to-text server:

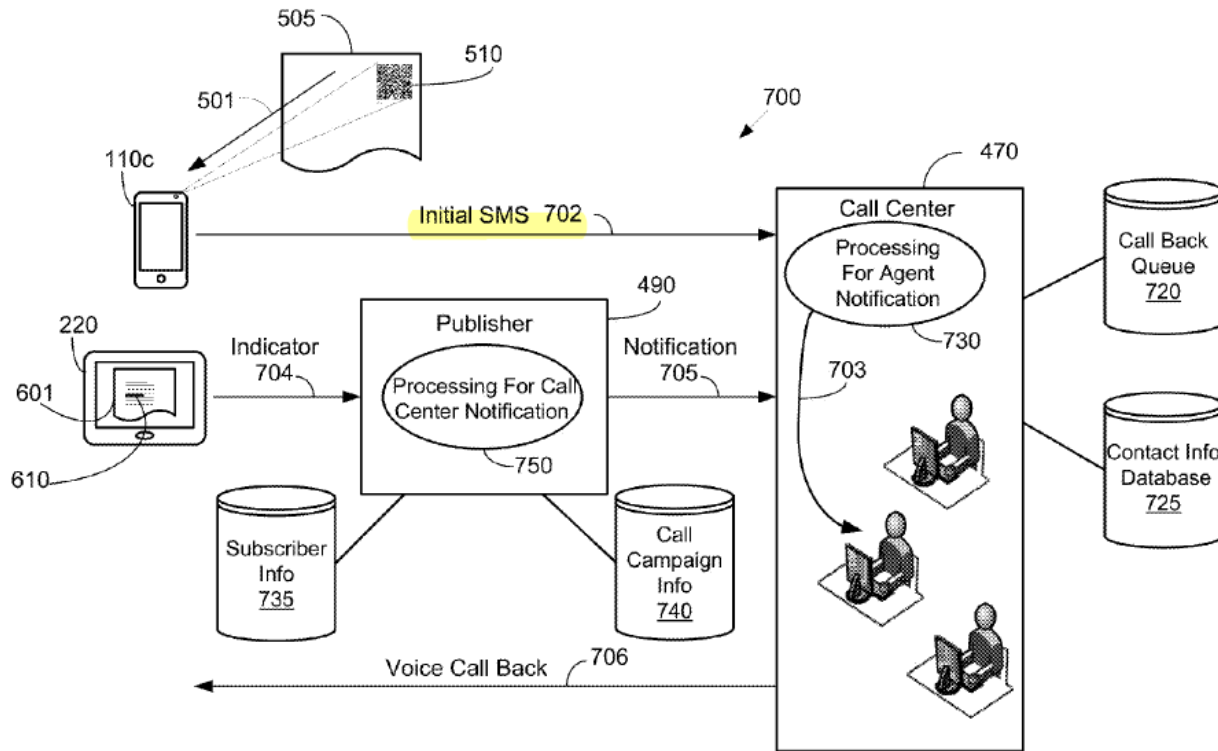


FIG. 7A

(Ex. 1007, Fig. 7A, 19:33-42, 12:39-47, Figs. 4B-4C, 18:38-42, 27:5-28:42). Specifically, the servers of the call center, including the ACD 130 (Ex. 1007, 8:33-9:8) receive and process the text message. (Ex. 1007, 20:31-40).

“[1h] receive the custom message at the click-to-text server; and”

125. Haggerty teaches that the click-to-text server **receives the custom text message**. (Ex. 1007, 23:1-12, Fig. 9, rn 905, Fig. 7A, 19:33-40, 12:39-47, 13:9-35, Figs. 4B-4C, 18:38-42).

“[1i] enroll the mobile device in a promotion associated with the promotional message based on receiving the custom message and without receiving any additional information from the mobile device.”

126. Haggerty teaches that, when the message is received at the click-to-text server, it **enrolls the mobile device in a promotion** by recording user information in a database, recording the user’s consent to receive further communications in a consent database, and placing the user in a queue for a response (a call, text or email back). (Ex. 1007, Fig. 9, 22:47-23:21, Fig. 4B, 12:39-13:8).

127. The communication to the user back from the click-to-text server can involve any type of conventionally-known **promotion**, such as the provision of a coupon, discount, or further information about a product or service. (Ex. 1007, 10:27-30, 13:1-8, Fig. 9, 23:8-21)(Ex. 1008, p. 030, ¶0068). The promotion is **associated with the promotional message**, because the click-to-text server first identifies the campaign associated with the promotional message prior to enrolling the user in the promotion. (Ex. 1007, Fig. 9, 22:47-23:21, Fig. 4B, 12:39-13:8). The enrollment in the promotion happens **without receiving any additional information from the mobile device**. This is shown, for example, in Fig. 4B, reproduced below, where only the initial SMS message (418) is received from the mobile device prior to the response SMS (422):

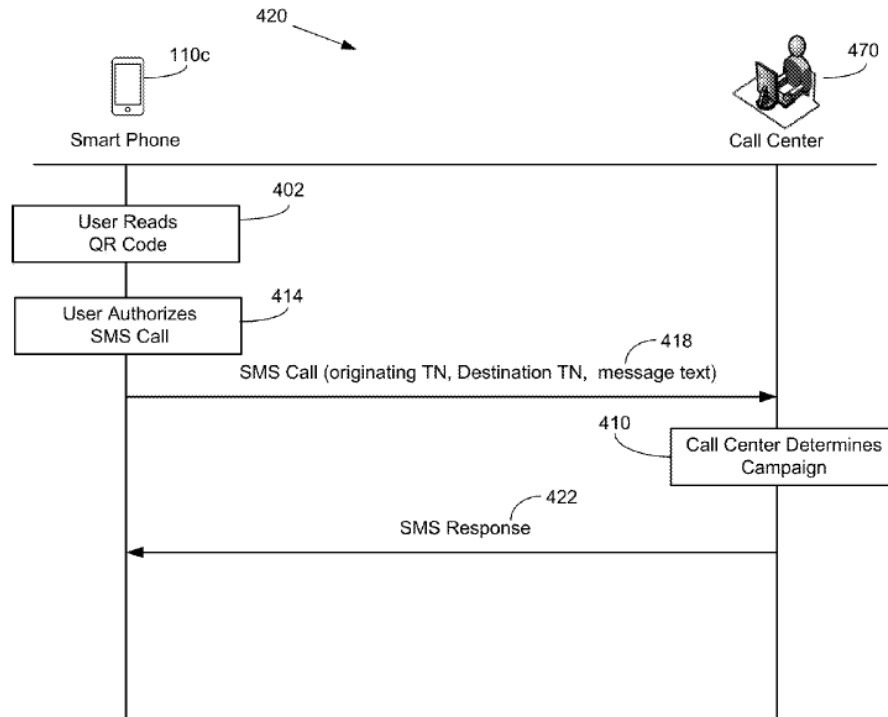


FIG. 4B

(Ex. 1007, Fig. 4B, 12:39-13:8)(see also Fig. 9, 22:47-23:21).

CLAIM 2

“2. The non-transitory processor-readable medium of claim 1, wherein the mobile device is enrolled in the promotion without the click-to-text server charging a fee to a user of the mobile device.”

128. Haggerty’s method does not charge a fee to the user for enrollment in the promotion. (Ex. 1007, Fig. 9, 22:47-23:21, Fig. 4B, 12:39-13:8).

CLAIM 13

“13. The non-transitory processor-readable medium of claim 1, wherein the user data includes at least one of a mobile device identifier, a user viewing history, user tap history, cookie information, an identification of a referring webpage, or a user agent of the first application.”

129. Demsey teaches that the user data can include an instance identifier, which is a **mobile device identifier**. (Ex. 1005, ¶¶0039, 0036, 0048). Demsey further discloses that the user data can be **cookie information** (Ex. 1005, Fig. 5, rn 4B, ¶¶0045, 0047).

CLAIM 14

“14. The non-transitory processor-readable medium of claim 1, the code further comprising code to cause the click-to-text server to: send a signal to cause the promotional message to be displayed at the mobile device via the first application.”

130. Both Demsey and Khanna teach having the central server (the **click-to-text server** in the combination) send an ad to the mobile device browser in response to an ad request. (Ex. 1005, Fig. 5 rn 4A-5, ¶¶0045-0048, 0030)(Ex. 1006, ¶¶0039-0040, Fig. 1). This is a **signal to cause the promotional message to be displayed at the mobile device via the first application**. Specifically, the signal is an HTTP message that provides an ad back to a user device in response to an ad request. (Ex. 1005, ¶0045, Fig. 5).

Ground 2. Claims 1-4, 6-10, 13-14, and 19 would have been obvious as in Ground 1, in further view of Rakavy.

131. In my opinion, claims 1-4, 6-10, 13-14, and 19 would have been obvious as in Ground 1, in further view of U.S. Pat. No. 5,913,040 (“**Rakavy**”)(Ex. 1018).

A. Overview of the Ground and Rationale for the Combination

132. In my opinion, claims 1-2 and 13-14 are obvious in the same way as in Ground 1. Additionally, claims 3-4, 6-10 and 19 all claim various forms of authentication, and would have been obvious over the combination of ground 1 in further view of motivation provided by Rakavy as I discuss below.

133. Rakavy is an early web-advertising patent. Like Demsey, Rakavy teaches an ad network with a central server (Advertising System Server 600) that can provide ads to user devices. (Ex. 1018, Abstract, Fig. 1, 4:39-45, 5:31-47). Like Demsey, Rakavy teaches assigning each user a user ID. (Ex. 1018, 5:31-47). Like Demsey, the user ID is used to identify a particular user to be better able to track user activities and target ads to the user. (Ex. 1018, 5:38-47).

134. As relevant to the present ground, Rakavy teaches authenticating the user. Specifically, Rakavy states that advertisements might offer incentives such as contests and rewards. (Ex. 1018, 3:34-39, 6:16-26, 15:14-21). To avoid fraudulent behavior related to such ads, Rakavy teaches that it is advisable to authenticate the user. (Ex. 1018, 15:14-21, 6:21-26). As one example, Rakavy further specifically teaches to use the user ID to authenticate the user. (Ex. 1018, 6:16-26).

135. I will provide more specific reasoning concerning the rationale for my obviousness opinions in the claim mapping section, below.

B. Reasonable Expectation of Success

136. A PHOSITA would have had a reasonable expectation of success in implementing ideas from Rakavy in the manner described above (*see* ¶94) and in the claim mapping, below. The systems involved are all software-based and would have been predictable in the relevant timeframe. The concepts from Rakavy (including the concept of authentication and checking a user ID) would have been simple to accomplish in a software system.

C. Analogous Art

137. Rakavy is in the same field as the '887 patent and would have been reasonably pertinent to the problems facing the named inventors. (Ex. 1018, 5:32-7:11, 15:14-21).

D. Claim Mapping

CLAIM 3

“3. The non-transitory processor-readable medium of claim 1, the code further comprising instructions to cause the click-to-text server to authenticate the mobile device based on the user data received in response to the mobile device loading the webpage and the identifier received with the custom message.”

(a) authenticate the mobile device based on the user data received in response to the mobile device loading the webpage

138. Demsey teaches that the **user data** can be a unique ID (*see* Ground 1, claim 1, limitation [1c]). Rakavy teaches to use a user ID to **authenticate** the user to avoid fraud. (Ex. 1018, 6:16-20, 15:14-21). It would have been obvious in the

combination to use Demsey's unique ID or instance ID (**user data received in response to the mobile device loading the webpage**) to **authenticate the mobile device** in order to mitigate the potential fraud mentioned by Rakavy. (Ex. 1018, 15:14-21, 6:16-20). Authentication as suggested by Rakavy would have been most obvious to perform by checking provided information (*e.g.* a unique ID) against a database of information. Rakavy expressly teaches a database for authenticating, *e.g.*, using a user ID. (Ex. 1018, 6:16-20).

(b) authenticate the mobile device...based on the identifier received with the custom message

139. A PHOSITA would have found it obvious to use a separate **identifier received with the custom message** (as described above in ¶¶119-123, *e.g.* a telephone number) to assist in the authentication of the user. For example, Haggerty (via its prior application, U.S. Pat. App. Ser. No. 13/461,541 (Ex. 1008), which is incorporated by reference in Haggerty at Ex. 1007, 1:11-15) teaches to use the user's telephone number to check against national wireless number databases, which is a form of authentication. (Ex. 1008, p. 031, ¶0069). Haggerty's incorporated '541 application further teaches that the SMS message can contain an additional security code associated with the user (**identifier received with the custom message**) to "aid the recipient in identifying and verifying the sender", *i.e.* to **authenticate the user**. (Ex. 1008, p. 031, ¶0069). Each of these methods would have provided additional and complementary

protection against the potential for fraud described by Rakavy, and thus would have been obvious to use. (Ex. 1018, 15:14-21, 6:16-20).

140. Furthermore, the methods of authentication discussed under this claim 3 were well-known in the relevant timeframe, as taught by Rakavy and Haggerty, and could have been used to improve Haggerty without unpredictable results.

CLAIM 4

“4. The non-transitory processor-readable medium of claim 1, the code further comprising instructions to cause the click-to-text server to: authenticate the mobile device based on the user data received in response to the mobile device loading the webpage; and send the promotional message to the mobile device in response to authenticating the mobile device.”

141. See above, claim 3 regarding authenticating based on user data. It would have been obvious, based on Rakavy, to send the promotional message to the mobile device only in response to authenticating the mobile device in order to avoid the fraud warned against by Rakavy, and to save resources associated with transmission to potentially fraudulent or spurious users. (Ex. 1018, 15:14-21, 6:16-20).

CLAIM 6

“6. The non-transitory processor-readable medium of claim 1, the code further comprising instructions to cause the click-to-text server to authenticate a user of the mobile device based on the custom message satisfying a predefined criteria, the mobile device enrolled in the promotion based on the user being authenticated.”

142. See above, claim 3. In the analysis of claim 3, the **predefined criteria** [sic] required by this claim (6) is that the user data and/or identifier matches corresponding stored data, such that the **user** can be **authenticated**. It would have been obvious to enroll the mobile device in the promotion (e.g. contest) based on the **mobile device being authenticated** to avoid the fraud described by Rakavy. (Ex. 1018, 15:14-21, 6:16-20).

CLAIM 7

“7. The non-transitory processor-readable medium of claim 1, the code further comprising instructions to cause the click-to-text server to authenticate a user of the mobile device based on information in the custom message appearing on a white list or being absent from a black list, the mobile device enrolled in the promotion based on the user being authenticated.”

143. See above, claim 3. Authentication using any of the methods discussed above under claim 3 would most obviously have required checking provided information against a database of information. (Ex. 1018, 6:16-20). Such checking constitutes **checking whether information in the custom message appears on a white list or is absent from a black list**. It would have been obvious to enroll the user in the promotion based on the user being authenticated for the reasons described above under claim 6. (Ex. 1018, 15:14-21, 6:16-20).

144. Furthermore, Haggerty, via its incorporated '541 application, teaches that the user's telephone number (obtained in the SMS message) can be checked against various lists, including “state or federal do-not-call (‘DNC’) databases or

wireless number lists.” (Ex. 1008, p. 025, ¶0051). It would have been obvious to check the telephone number against telephone numbers that have been registered to not receive marketing communications (a **blacklist**) or wireless mobile numbers (a **whitelist**) prior to enrolling the user in the promotion to determine whether the user has been spoofed or whether the user has otherwise indicated it should not be contacted, and **enrolling the user in the promotion** only on the **basis** of such **authentication**, in order to avoid the potential fraud described by Rakavy. (Ex. 1018, 15:14-21, 6:16-20).

CLAIM 8

“8. The non-transitory processor-readable medium of claim 1, the code further comprising instructions to cause the click-to-text server to determine that the mobile device is a new subscriber, the mobile device enrolled in the promotion based on the mobile device being a new subscriber.”

145. See above, claim 3. It further would have been obvious to determine that the mobile device is a new subscriber, the mobile device enrolled in the promotion based on the mobile device being a new subscriber. Rakavy teaches that an advertisement can use an award or contest (Ex. 1018, 3:37-49) and that the system “monitors the user’s interaction with the advertisements and produces raw data on how many times a particular advertisement was accessed as well as the user’s response to advertisements” (Ex. 1018, 3:47-51) and “check the user’s participation and any awards from contests”. (Ex. 1018, 6:16-26). Rakavy further

teaches to prevent fraud in promotions. (Ex. 1018, 15:14-21). Based on this, it would have been obvious to check whether the user is already participating in a promotion where some incentive (a contest, discount, etc.) is offered, and to permit new enrollment upon the **determination that the mobile device is a new subscriber**, and further to ensure that the user is not enrolling in the promotion multiple times.

CLAIM 9

“9. The non-transitory processor-readable medium of claim 1, the code further comprising instructions to cause the click-to-text server to authenticate the mobile device based on the user data sent in response to the mobile device loading the webpage.”

146. See above, claim 3, under which authentication based on user data is explained.

CLAIM 10

“10. The non-transitory processor-readable medium of claim 1, the code further comprising instructions to cause the click-to-text server to authenticate the mobile device based on the identifier received with the custom message.”

147. See above, claim 3, under which authentication based on the identifier received with the custom message is explained.

CLAIM 19

“19. The non-transitory processor-readable medium of claim 1, wherein the mobile device is enrolled in the promotion based on a determination, by the click-to-text server that the mobile device is

not an existing subscriber to the promotion based on the identifier.”

148. See above, claim 8.

Ground 3. Claims 1-2, 5, and 13-18 were obvious as in Ground 1, in further view of *Ad Serving Technology*.

149. I am of the opinion that claims 1-2, 5, 13-18, and 20 were obvious as in Ground 1, in further view of *Ad Serving Technology* (Ex. 1012).

A. Overview of the Ground

150. This ground relates to the claims at-issue under ground 1, and additionally, claims 5, 15-18 based on *Ad Serving Technology*, which is a general textbook related to Internet ad technology. (Ex. 1012, pp. 005-028).

151. Claim 15 requires that the integration tag include code to cause the first application to display the promotional message in an **iframe**. An iframe was a well-known HTML tag in the relevant timeframe frequently used to display ads, as documented in *Ad Serving Technology*. (Ex. 1012, p. 191, 442). It thus would have been obvious to use an iframe for its well-known purpose described in *Ad Serving Technology*.

152. Each of claims 5 and 16-18, in turn, requires the integration tag be configured to execute in a certain way: by distinguishing between mobile and desktop devices (claim 5) or by using business rules to decide whether to display

the ad (claims 16-18). *Ad Serving Technology* 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, as explained above in Ground 3. (Ex. 1012, p. 054).

153. I will provide more specific reasoning concerning the rationale for my obviousness opinions in the claim mapping section, below.

B. Reasonable Expectation of Success

154. A PHOSITA would have had a reasonable expectation of success in using the teachings of *Ad Serving Technology* in the manner described herein.

C. Analogous Art

155. *Ad Serving Technology*, as a general textbook describing Internet techniques for Advertising was in the same field as the '887 patent and would have been reasonably pertinent to the problems facing the named inventors. (Ex. 1012, pp. 005-028).

D. Claim Mapping

CLAIM 5

“5. The non-transitory processor-readable medium of claim 1, the code further comprising instructions to cause the click-to-text server to define the integration tag, the integration tag configured to be executed by mobile devices and not desktop computers or laptop computers.”

156. In my opinion, the combination renders claim 5 obvious. Regarding the **click-to-text server defining the integration tag**, it would have been obvious to do so for the reasons provided under Ground 1, claim 1, limitation [1b], ¶¶106-108, specifically that it would have been easiest for the server to define ad tags because there might be thousands of publisher clients, ad tags may need to be changed periodically, and the click-to-text server would best be able to define the ad tags such that the data they send is compatible with the needs of the click-to-text server. It further would have been obvious to have the integration tag **configured to be executed by mobile devices and not desktop computers or laptop computers** given the purpose of Haggerty to present ads with links that will pre-populate SMS messages (*see* above, Ground 4, claim 1, limitation [1f]), and the fact that mobile devices, not desktop computers or laptop computers, typically have SMS apps that can respond to such links. *Ad Serving Technology* adds to this by teaching that different ads are appropriate for mobile devices versus other devices (Ex. 1012, p. 282), and that JavaScript can “easily” distinguish between mobile and non-mobile users. (Ex. 1012, pp. 282-283). Because Demsey’s **integration tags** include JavaScript, it would have been obvious to use this JavaScript to determine whether the user is using a mobile device, and to only present ads intended for mobile devices when mobile devices are being used.

CLAIM 15

“15. The non-transitory processor-readable medium of claim 1, wherein the integration tag includes code to cause the first application to display the promotional message in an iframe.”

157. *Ad Serving Technology* teaches that iframes were a known technology for implementing ad tags (a form of **integration tag**) in the relevant timeframe. (Ex. 1012, p. 191)(“The ‘ad tags’ themselves are short pieces of code that simply reference the location of the ad on the third party ad server. This code is most commonly generated as JavaScript although it **can also be generated as an iframe (more HTML friendly script)** or for added flexibility, standard HTML.”)(Emphasis added). These ad tag iframes fetch ads for display within the ad tag iframe. (Ex. 1012, p. 442)(“Some Publishers (particularly in the exchange space - See Chapter 8) will **serve ads from within an HTML iframe.**”)(Emphasis added). A PHOSITA would have understood how to use iframes predictably, and would have found it obvious to use an iframe to display a promotional message (ad) because this was a known technique for displaying ads that would have been implementable without unpredictable results. I note here that JavaScript can be used *within* an iframe. (e.g. Ex. 1029, 16:38-45, Fig. 9B).

CLAIM 16

“16. The non-transitory processor-readable medium of claim 1, wherein the integration tag includes code to cause the first application to display the promotional message in response to a

business rule being met.”

158. See above, claim 5, where the **business rule** executed in the **integration tag** to determine if the **promotional message is displayed** is that the ad should be displayed when the user is using a mobile device.

159. Furthermore, it would have been obvious based on *Ad Serving Technology* to cause the integration tag to contain code to execute a business rule that only shows a promotional message when a user has not seen the promotional message within a certain prior time period, because *Ad Serving Technology* teaches that the system should limit the frequency of ad display to users (Ex. 1012, pp. 131-132)(“[f]requency caps are usually set to limit the user’s exposure to just a campaign to a handful of ‘impressions’ over a given time period of twenty four hours or less.”). *Ad Serving Technology* likewise teaches using JavaScript in ad tags to control when ads should be displayed. (Ex. 1012, p. 054)(“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.”)(see also pp. 051-054). It would have been obvious from these disclosures to **include code** in the integration tag **to cause the first application to display the promotional message in response to a business rule being met**, e.g. limiting views based on recent prior viewing by the same user.

CLAIM 17

“17. The non-transitory processor-readable medium of claim 16, wherein the business rule is at least one of a whitelist, a blacklist or a timeout.”

160. See above, claim 16, where preventing re-serving of an ad within a given timeframe is the **timeout**. *Ad Serving Technology* teaches that “[f]requency caps are usually set to limit the user’s exposure to just a campaign to a handful of ‘impressions’ over a given time period of twenty four hours or less.” (Ex. 1012, p. 132).

“18. The non-transitory processor-readable medium of claim 16, wherein the business rule is that a predefined event occurred within a predefined period of time before the integration tag was executed.”

161. See above, claim 17. In the case of claim 18, the business rule is a rule that an ad was not served more than a specified number of times (the “frequency cap”) within a prior, predefined period of time (e.g. 24 hours). (Ex. 1012, p. 132).

Ground 4. Claims 1-2, 11-14, and 20 were obvious as in Ground 1, in further view of Molinet.

162. In my opinion, claims 1-2, 11-14, and 20 were obvious as in Ground 1, in further view of U.S. Pat. App. Pub. 2016/0142858 A1 (“**Molinet**”)(Ex. 1009).

A. Overview of the Ground

163. This ground relates to the claims at-issue under ground 1 in the same way as explained under ground 1, and additionally, deals with claims 11-12 and 20. Each of claims 11-12 and 20 requires that the URI that is sent from the click-to-text server to the mobile device be customized for the operating system of the mobile device (or simply for the mobile device as in claim 20). In my opinion, Molinet teaches this feature.

1. Overview of Molinet

164. Molinet teaches a system that can use mobile app deeplinking with “contextual deeplinks”. (Ex. 1009, ¶¶0016-0018). “Contextual deeplinks” are app deeplinks that are customized for the particular operating environment of a user’s mobile device. (Ex. 1009, ¶0037). This customization is desirable, according to Molinet, because different operating environments may not use or allow all deeplinking protocols. (Ex. 1009, ¶0005). Molinet further incorporates (Ex. 1009, ¶0001) its provisional application, U.S. Prov. App. 62/079,512 (Ex. 1010), which is relevant to certain aspects of claims 11-12. For example, the ’512 provisional teaches the use of standard URIs for mobile app deeplinking used by the iOS and Android operating systems. (Ex. 1010, pp. 021, 015).

165. I will provide more specific reasoning concerning the rationale for my obviousness opinions in the claim mapping section, below.

B. Reasonable Expectation of Success

166. A PHOSITA would have had a reasonable expectation of success in using the teachings of Molinet in the manner described herein.

C. Analogous Art

167. Molinet is analogous art because it is in the same field as the '887 patent and is reasonably related to the problems facing the named inventors. (Ex. 1009, ¶0003).

D. Claim Mapping

CLAIM 11

“11. The non-transitory processor-readable medium of claim 1, wherein: the user data includes an indication of an operating system of the mobile device; the messaging application is a default messaging application for the operating system; and the URI is customized for the operating system such that the URI deeplinks to the default messaging application.”

168. The combination renders claim 11 obvious. First, Molinet, like Demsey, teaches collecting user data (as discussed above under claim 1, element [1c]) using JavaScript on a webpage. (Ex. 1009, ¶0065). The user data includes **an indication of an operating system of the mobile device.** (Ex. 1009, ¶0065)(“These characteristics may include network address (e.g., IP address), operating system version, operating system type, device model information, and screen size.”)(*see also* Ex. 1009, ¶¶0017, 0025, 0037, 0054, 0069, 0085, 0105).

169. Molinet further teaches that its system **customizes the URI** (as discussed above under claim 1, element [1d]) based on the user data, including the operating system, such that the URI is operable on the particular mobile device.

Molinet teaches:

“The server link module 115 also responds to requests associated with contextual deep links....If no fingerprint exists, the server link module 115 generates one based on characteristics of the client device 120 (e.g., network address, **operating system version**, and so on). The server link module 115 also determines contextual information for the client device 120 based on the characteristics of the client device 120. **Based on the contextual information**, the server link module 115 transmits a response back to the client device 120 **such that the response is operable on the client device 120.**”

(Ex. 1009, ¶0037)(Emphasis added)(*see also* Ex. 1009, Abstract, Fig. 5 rn 550-560, ¶¶0018, 0037, 0065, 0080, 0107). Note that it is clear from Molinet that the “operating system” type and version are part of the contextual information used to ensure that an operable response (including a URI) is sent to the client device. (Ex. 1009, ¶¶0065, 0037, 0017, 0025, 0054, 0069, 0085, 0105).

170. Molinet’s incorporated provisional application (the ’512 provisional) furthermore teaches:

“‘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, p. 021). It would have been obvious to use such direct deeplinking technology built into the operating systems of mobile devices to implement the deeplinking of Khanna, where the particular direct deeplink is chosen according to the operating system of the mobile device, because these deeplinking functions were already provided by common mobile operating systems. (Ex. 1010, pp. 021-022, 015). As discussed in ¶¶39-44, above, the deeplinking URI formats were also widely known for SMS deeplinking in the relevant timeframe.

171. Furthermore, it would have been obvious to configure the URI to deeplink to the default messaging application, such that the **messaging application is a default messaging application for the operating system**. Specifically, Haggerty teaches using a text-messaging (SMS) application, and in the relevant timeframe, most smartphones came with a default text messaging (SMS) app, making that the most obvious choice (the app that comes with the phone is likely to be installed, whereas with a non-default app there is a risk that the app is not installed (Ex. 1006, ¶0046)). Furthermore, there were already standard URI

formats available to deeplink to default SMS apps (including sms: and smsto:), making the default app the most obvious choice. (Ex. 1010, pp. 015, 021).

CLAIM 12

“12[a]. The non-transitory processor-readable medium of claim 1, wherein: the user data includes a user agent associated with the first application; and the messaging application is a default messaging application for an operating system of the mobile device, the code further comprising instructions to cause the click-to-text server to: identify the operating system of the mobile device based on the user agent, the URI customized for the operating system such that the URI deeplinks to the default messaging application.”

172. See above, claim 11. Claim 12 differs from claim 11 in that the user data includes a user agent associated with the first application, and the software identifies the operating system of the mobile device based on the user agent. The term “user agent” refers to a variety of possible information about the mobile device, operating system and browser that can be sent along with an HTTP request in what is known as the User-Agent field of the HTTP header. (Ex. 1013, ¶0048)(Ex. 1010, p. 022)(Ex. 1016, pp. 328). The user agent information is also accessible by JavaScript operating on a Web page. (Ex. 1021, pp. 364-365, 992).

173. Molinet teaches that **user agent** information can be collected as part of the user data. (Ex. 1009, ¶0085)(“Examples of contextual information that may be used by the contextual response module 340 to determine the platform type of the client device 120 includes the operating system type and version, **user agent information**, device model, and so on.”)(Emphasis added)(Ex. 1010, p. 022). The

“user agent information” in Molinet means the same thing as “user agent” in the claim. As Molinet’s incorporated provisional application explains, the user agent information sent by a browser refers to the “User-Agent” field of an HTTP request, which included an identification of the operating system, operating system version, and name and version of the browser itself. (Ex. 1010, p. 022)(Ex. 1009, ¶0065). It would have been obvious to collect user agent information as “user data” sent together with the ad request of Demsey (*see* above, ground 1, claim 1, element [1c]), both because the user agent data is sent with the HTTP header in the ad request (Ex. 1010, p. 022)(Ex. 1005, Fig. 5, ¶0045), and because the user agent data could be accessed using JavaScript executed by a browser by using, for example, the Navigator object’s userAgent field. (Ex. 1021, pp. 364-365, 992). In this way, it would have been obvious to **identify the operating system of the mobile device based on the user agent**, because the user agent states the identity of the operating system directly. (*Id.*). Claim 12 is otherwise obvious for the same reasons as claim 11.

CLAIM 20

“20. The non-transitory processor-readable medium of claim 1, wherein the URI is customized for the mobile device based on the user data.”

174. *See* above, claim 11. If a URI is customized for the operating system of the mobile device, it is **customized for the mobile device**.

X. SECONDARY CONSIDERATIONS

175. 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 ¶¶30-36, 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).

176. I understand that in a co-pending district court litigation (where the Patent Owner was responding to an argument that the '887 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.

177. 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

178. 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.

August 16, 2023
Date

Henry Houh
Dr. Henry Houh