

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

NETFLIX, INC.,
Petitioner,

v.

GoTV Streaming, LLC,
Patent Owner

Case: IPR2023-00759
U.S. Patent No. 8,103,865

PETITION FOR *INTER PARTES* REVIEW

Petition Filing Date: April 20, 2023

TABLE OF CONTENTS

I.	INTRODUCTION	1
II.	GROUND FOR STANDING (37 C.F.R. §42.104(A)).....	1
III.	STATEMENT OF PRECISE RELIEF REQUESTED FOR EACH CLAIM CHALLENGED, INCLUDING CLAIMS AND STUTORY GROUND (37 C.F.R. §42.204(B))	2
IV.	OVERVIEW OF THE '865 PATENT	3
A.	Background of Serving Remotely Executed Applications to Client Devices	3
B.	The Specification.....	4
C.	The Prosecution History	8
V.	LEVEL OF ORDINARY SKILL IN THE ART	9
VI.	CLAIM CONSTRUCTION	10
VII.	THE PRIMARY PRIOR ART REFERENCES	10
A.	Overview of Hariki.....	10
B.	Overview of Harris	12
C.	Overview of Dees	16
D.	Other Evidence Regarding the State of the Art.....	18
VIII.	DETAILED EXPLANATION OF THE GROUND.....	18
A.	Ground 1: Hariki in view of Harris renders obvious Claims 1-20, 31, 34.....	18
1.	Claim 1	18
2.	Claim 2	43
3.	Claim 3	44

4.	Claim 4.....	45
5.	Claim 5.....	46
6.	Claim 6.....	48
7.	Claim 7.....	49
8.	Claim 8.....	50
9.	Claim 9.....	51
10.	Claim 10.....	52
11.	Claim 11.....	56
12.	Claim 31.....	57
13.	Claim 12.....	58
14.	Claim 13.....	64
15.	Claim 14.....	64
16.	Claim 15.....	64
17.	Claim 16.....	64
18.	Claim 17.....	64
19.	Claim 18.....	65
20.	Claim 19.....	65
21.	Claim 20.....	65
22.	Claim 34.....	66
B.	Ground 2: Hariki in view of Harris and Dees renders obvious Claims 1-20, 31, 34.....	66
1.	Claim 1.....	66
2.	Claim 2.....	66

3.	Claim 3	66
4.	Claim 4	67
5.	Claim 5	67
6.	Claim 6	71
7.	Claim 7	73
8.	Claim 8	75
9.	Claim 9	75
10.	Claim 10	75
11.	Claim 11	75
12.	Claim 31	75
13.	Claim 12	75
14.	Claim 13	76
15.	Claim 14	76
16.	Claim 15	76
17.	Claim 16	76
18.	Claim 17	76
19.	Claim 18	76
20.	Claim 19	77
21.	Claim 20	77
22.	Claim 34	77
IX.	SECONDARY CONSIDERATIONS	77
X.	DISCRETIONARY DENIAL IS NOT WARRANTED UNDER EITHER § 314 OR § 325	77

A.	Fintiv Factors.....	77
B.	Becton Dickinson/Advanced Bionics/General Plastics Factors	81
XI.	MANDATORY NOTICES	81
A.	Real Parties-in-Interest.....	81
B.	Related Matters.....	81
C.	Lead and Back-Up Counsel, and Service Information Under 37 C.F.R. § 42.8(b)(3).....	82
D.	Service Information Under 37 C.F.R. § 42.8(b)(4).....	82
E.	Fees.....	82
XII.	CONCLUSION.....	83

EXHIBIT LIST

No.	Description
1001	U.S. Patent No. 8,103,865 (“’865Pat”)
1002	Expert Declaration of Ben Bederson (“ <i>Bederson</i> ”)
1003	Curriculum Vitae of Benjamin B. Bederson, Ph.D.
1004	File History of U.S. Patent No. 8,103,865 (“’865FH”)
1005	U.S. Patent Application Publication No. 2007/0150617 (“ <i>Hariki</i> ”)
1006	U.S. Patent Application Publication No. 2003/0023755 (“ <i>Harris</i> ”)
1007	U.S. Patent Application Publication No. 2003/0137539 (“ <i>Dees</i> ”)
1008	RESERVED
1009	U.S. Patent No. 7,447,486 (“ <i>Tamura</i> ”)
1010	U.S. Patent No. 6,996,627 (“ <i>Carden</i> ”)
1011	U.S. Patent No. 6,669,564 (“ <i>Young</i> ”)
1012	U.S. Patent No. 6,732,183 (“ <i>Graham</i> ”)
1013	U.S. Patent No. 6,507,727 (“ <i>Henrick</i> ”)
1014	Webpage excerpt from Computer History Museum at https://www.computerhistory.org/tdih/april/6/ (last accessed March 29, 2023)
1015	https://www.webdesignerdepot.com/2009/05/the-evolution-of-cell-phone-design-between-1983-2009/
1016	Excerpts from Dan R. Olsen, Jr., Developing user interfaces (1998)
1017	Alok Sinha, Client-server computing, in Communications of the ACM, 35, 7 (1992)
1018	Tim Berners-Lee, Robert Cailliau, Ari Luotonen, Henrik Frystyk Nielsen, and Arthur Secret, The World-Wide Web, Communications of the ACM 37, 8 76-82 (August 1994), available at http://dx.doi.org/10.1145/179606.179671

1019	Eric Kasten, HTML: A Gentle Introduction, Linux Journal (July 1995), available at https://www.linuxjournal.com/article/1081
1020	U.S. Patent Application Publication No. 2013/0124961 (“ <i>Linburn</i> ”)
1021	U.S. Patent No. 8,111,326 (“ <i>Talwar</i> ”)
1022	Docket Control Order, <i>GovTV Streaming, LLC v. Netflix, Inc.</i> , No. 2:22-cv-07556-RGK-SHK, Dkt. 61 (C.D. Cal. Feb. 13, 2023)
1023	Complaint in <i>GovTV Streaming, LLC v. Netflix, Inc.</i> , No. 2:22-cv-07556-RGK-SHK (C.D. Cal. October 17, 2022)
1024	https://web.archive.org/web/20070406171650/http://www.zenzui.com/products.html
1025	Bert Boss, <i>A brief history of CSS until 2016</i> , W3 Blog (Dec. 17, 2016), https://www.w3.org/Style/CSS20/history.html
1026	Eric A. Meyer, <i>Cascading Style Sheets – The Definitive Guide</i> (O’Reilly, 2d ed. 2004)
1027	<i>JavaScript</i> , Wikipedia (Aug. 12, 2004), https://web.archive.org/web/20040812085853/http://en.wikipedia.org/wiki/JavaScript
1028	<i>Document Object Model (DOM) Level 1 Specification Version 1.0</i> , W3C Recommendation (Oct. 1, 1998), https://www.w3.org/TR/1998/REC-DOM-Level-1-19981001/DOM.pdf
1029	<i>ECMAScript: A general purpose, cross-platform programming language</i> , Standard ECMA-262 (June, 1997), https://www.ecma-international.org/wp-content/uploads/ECMA-262_1st_edition_june_1997.pdf
1030	Peter-Paul Koch, <i>The Document Object Model: an Introduction</i> , Digital Web Magazine (May 14, 2001), https://www.digital-web.com/articles/the_document_object_model/

LISTING OF CHALLENGED CLAIMS

Claim 1	
[1pre]	A server implemented method for processing data for a wireless device, comprising:
1[a]	receiving from the wireless device a request for an application program, said request including an indication of a type of the wireless device;
1[b]	executing, in response to receiving said request, said application program to generate a wireless device generic template including a plurality of content items;
[1c]	sending a custom configuration to the wireless device, said custom configuration being specific to said application program;
[1d]	generating a page description based on said wireless device generic template and a capability of the wireless device,
[1e]	said page description having at least one discrete low level rendering command that is within said rendering capability of said wireless device but that is of a syntax that is wireless device generic; and
[1f]	sending said page description to the wireless device such that the wireless device is capable of presenting at least one content item from said plurality of content items using both said page description and said custom configuration.
Claim 2	
[2]	The method of claim 1, wherein said wireless device generic template includes relative screen location and display object size information and is independent of screen dimensions of the wireless device.
Claim 3	
[3]	The method of claim 1, wherein said wireless device generic template is substantially compliant with extensible markup language (XML) syntax.

Claim 4	
[4]	The method of claim 1, wherein said at least one discrete low level rendering command of said page description is operable to be executed by a plurality of rendering blocks of a graphical user interface of the wireless device.
Claim 5	
[5]	The method of claim 4, wherein said at least one discrete low level rendering command includes a physical screen position of a display element of said at least one content item.
Claim 6	
[6]	The method of claim 4, wherein said at least one discrete low level rendering command includes: a specification of a location and a size of a screen area; and an identification of a rendering block from said plurality of rendering blocks of said graphical user interface to operate within said screen area.
Claim 7	
[7]	The method of claim 6, wherein said at least one discrete low level rendering command includes parameters associated with said rendering block from said plurality of rendering blocks.
Claim 8	
[8]	The method of claim 1, wherein said page description includes at least one display rendering command and at least one audio rendering command.
Claim 9	
[9]	The method of claim 1, wherein said application program is wireless device generic and said wireless device generic template is a wireless device generic page description template.
Claim 10	

[10]	The method of claim 1, wherein the wireless device is operable to configure a plurality of rendering blocks of a graphical user interface based on said custom configuration.
Claim 11	
[11]	The method of claim 1, wherein said server stores a plurality of device profiles, said method further comprises: identifying a device profile from said plurality of device profiles associated with the wireless device based on said type of said wireless device.
Claim 12	
[12pre]	A non-transitory processor-readable medium storing code representing instructions to cause a processor to:
[12a]	receive a request for an application program from a wireless device, said request including an indication of a type of the wireless device;
[12b]	execute said application program to generate a wireless device generic template including a plurality of content items;
[12c]	request, using said indication of said type of the wireless device, a device profile from a library of device profiles,
[12d]	said device profile providing an indication of a rendering capability of the wireless device;
[12e]	eliminate, using said device profile, at least one content item from said plurality of content items from said wireless device generic template to define a device specific template associated with the wireless device,
[12f]	said at least one content item being unsupported by the wireless device;
[12g]	translate said device specific template into a page description having at least one discrete low level rendering command that is within said rendering capability of the wireless device but that is of a syntax that is wireless device generic; and

[12h]	send said page description to the wireless device.
Claim 13	
[13]	The non-transitory processor-readable medium of claim 12, wherein said wireless device generic template contains relative screen location and display object size information independent of screen dimensions of the wireless device.
Claim 14	
[14]	The non-transitory processor-readable medium of claim 12, wherein said wireless device generic template is substantially compliant with extensible markup language (XML) syntax.
Claim 15	
[15]	The non-transitory processor-readable medium of claim 12, wherein said at least one discrete low level rendering command of said page description is operable to be executed by a plurality of rendering blocks of a graphical user interface of the wireless device.
Claim 16	
[16]	The non-transitory processor-readable medium of claim 15, wherein said at least one discrete low level rendering command includes a physical screen position of a display element of a content item from the plurality of content items.
Claim 17	
[17]	The non-transitory processor-readable medium of claim 15, wherein said at least one discrete low level rendering command includes: a specification of a location and a size of a screen area; and an identification of a rendering block from said plurality of rendering blocks of said graphical user interface.
Claim 18	

[18]	The non-transitory processor-readable medium of claim 17, wherein said at least one discrete low level rendering command includes at least one parameter associated with said rendering block from said plurality of rendering blocks.
Claim 19	
[19]	The non-transitory processor-readable medium of claim 12, wherein said page description includes at least one display rendering command and at least one audio rendering command.
Claim 20	
[20a]	The non-transitory processor-readable medium of claim 12, further comprising code representing instructions to cause said processor to: send, in response to said request, a custom configuration to said wireless device, the wireless device is operable to configure a plurality of rendering blocks of graphical user interface based on said custom configuration.
[20b]	the wireless device is operable to configure a plurality of rendering blocks of graphical user interface based on said custom configuration.
Claim 31	
[31]	The method of claim 1, wherein the sending the custom configuration to the wireless device is in response to receiving an indication from the wireless device that the custom configuration is not stored at the wireless device.
Claim 34	
[34a]	The non-transitory processor-readable medium of claim 12, further comprising code representing instructions to cause the processor to: send a custom configuration to the wireless device, said custom configuration being specific to said application program,
[34b]	the wireless device operable to present a content item from said plurality of content items using said page description and said custom configuration.

I. INTRODUCTION

Netflix, Inc. (“Petitioner”) respectfully requests *inter partes* review of claims 1-20, 31, and 34 of U.S. Patent No. 8,103,865 (“the ’865 patent” or “’865Pat”), filed on August 1, 2007, and assigned to GoTV Streaming, LLC (“PO” or “GoTV”).

As demonstrated below, there is a high likelihood that Petitioner will prevail with respect to each of the challenged claims and, therefore, Petitioner respectfully requests that the Board institute trial.

II. GROUNDS FOR STANDING (37 C.F.R. §42.104(A))

Petitioner certifies that the ’865 patent is available for *inter partes* review (“IPR”) and Petitioner is not barred or estopped from requesting IPR to challenge the claims on the grounds herein. Petitioner files this Petition within one year of service of Patent Owner’s complaint against Petitioner. *See GoTV Streaming, LLC v. Netflix, Inc.*, No. 2:22-cv-07556-RGK-SHK (C.D. Cal. Oct. 17, 2022) (Netflix, Inc., served 10/25/2022).

III. STATEMENT OF PRECISE RELIEF REQUESTED FOR EACH CLAIM CHALLENGED, INCLUDING CLAIMS AND STATUTORY GROUNDS (37 C.F.R. §42.204(B))

Petitioner respectfully requests review and cancellation under 35 U.S.C. §311 of the challenged claims in view of:¹

Prior Art
U.S. Patent Application Publication No. 2007/0150617 (“ <i>Hariki</i> ”), filed July 25, 2006, published January 28, 2007, priority December 22, 2005; prior art under 35 U.S.C. §102(a) (Exhibit 1005).
U.S. Patent Application Publication No. 2003/0023755 (“ <i>Harris</i> ”), published January, 30, 2003; prior art under 35 U.S.C. §102(b) (Exhibit 1006)
U.S. Patent Application Publication No. 2003/0137539 (“ <i>Dees</i> ”), published July, 24, 2003; prior art under 35 U.S.C. §102(b) (Exhibit 1007)

Ground	Claims	Basis
1	1-20, 31, 34	35 U.S.C. §103 based on <i>Hariki</i> in view of <i>Harris</i>
2	1-20, 31, 34	35 U.S.C. §103 based on <i>Hariki</i> in view of <i>Harris</i> and <i>Dees</i>

¹ Petitioner does not concede that any challenged claim satisfies other requirements for patentability that cannot be raised in IPR, including Sections 101 and 112.

IV. OVERVIEW OF THE '865 PATENT²

A. *Background of Serving Remotely Executed Applications to Client Devices*

Downloading content from a remote device through a network was well-known before 2007. By 2000, following the dot-com boom of the late 1990s, the popularity of the Internet increased, and users were turning from traditional media sources, such as television and newspaper, to the Internet to receive media content. EX1009, 1:12-16.³ A user may go to media servers containing information from various content providers. *Id.*, 1:16-19. For example, techniques for

² This background, and the Grounds of Unpatentability herein, are further supported by the Declaration of Petitioner's expert, Ben Bederson, who has over 30 years of experience related to remote application execution. Declaration of Ben Bederson ("*Bederson*") (EX1002).

³ To the extent that any of the references in this section are cited in this Petition, it merely demonstrates POSITAs' knowledge, the state of the art, and/or as evidence that POSITAs would have been motivated to make the combinations in the manner discussed in this Petition. *KSR Int'l Co. v. Teleflex Inc.*, 550 U.S. 398, 420 (2007) ("[A]ny need or problem known in the field of endeavor at the time of invention and addressed by the patent can provide a reason for combining the elements in the manner claimed."). These exhibits are not part of the unpatentability grounds.

transmitting/receiving data via a cellular phone were developed well before 2003, so that users can experience content with a large amount of data, such as video and music, on their cellular phones. EX1010, 1:17-20.

Those in the art recognized that, by 2007, servers could provide to a remote client automatic delivery of multimedia content, such as entertainment content. EX1009, 1:29-34; EX1011, Abstract; *see also* EX1012, Abstract. Prior to the '865 patent, it was a known problem that transmission of content was limited by bandwidth and connection rates of mobile devices. EX1013, 1:42-57. A common goal was to improve network efficiency from a transmitter to a terminal. *Id.*

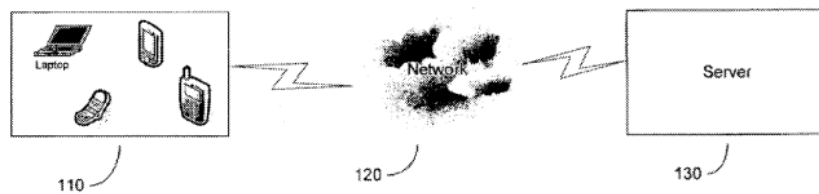
Client-server architectures, rendering technologies, and wireless devices were well-known to those of ordinary skill in the art at the time of the alleged inventions claimed in the '865 Patent. *Bederson*, ¶¶55-65 (citing exhibits 1014-1018, 1025-1030).

B. *The Specification*

The '865 patent is directed to “[a] server implemented method for processing data for a wireless device.” ’865*Pat*, Abstract. The specification explains that “a wireless device... is operable to communicate with a server and receive a series of low level basic commands that layout content, position, etc. for rendering application content on the wireless device.” *Id.*, 2:46-50. Figure 1A is described as “an exemplary communication system 100A in accordance with one

embodiment of the present invention including wireless clients and a remote server.” *Id.*, 5:65-6:1. The system includes “one or more wireless devices 110 [that] are coupled to a server 130 through a network 120.” *Id.*, 6:1-3.

100A



'865Pat, FIG. 1A.

Figures 2A and 2B show “communication sequence[s] 200A and 200B between a wireless device 210 and a remote server 230.” *Id.*, 11:9-12. “At step 212 the client 210 sends a message to the server 230 identifying the wireless device type and its capabilities along with a request to access an application on the server.” *Id.*, 11:12-15. The specification explains that “the application is generic and independent of the wireless device type” and “the wireless device type and its capabilities may include” a multitude of things, such as “the brand, the model and the version number” or device capabilities. *Id.*, 11:15-16, 22-31. The enumerated device types, however, “should not be construed as limiting the scope of the present invention.” *Id.*, 11:31-33. “In response to the message 212, the server 230

identifies the requested application and its corresponding custom configuration to be used by the client 210” by “send[ing] a message to the client 210 identifying the corresponding custom configuration.” *Id.*, 11:59-63. If the wireless device does not have the identified custom configuration, it may request “to download the identified information (e.g., custom configuration) from the server 230” at step 216. *Id.*, 12:17-21. After receiving the request, the server sends the custom configuration to the wireless device. *Id.*, 12:62-64.

200A

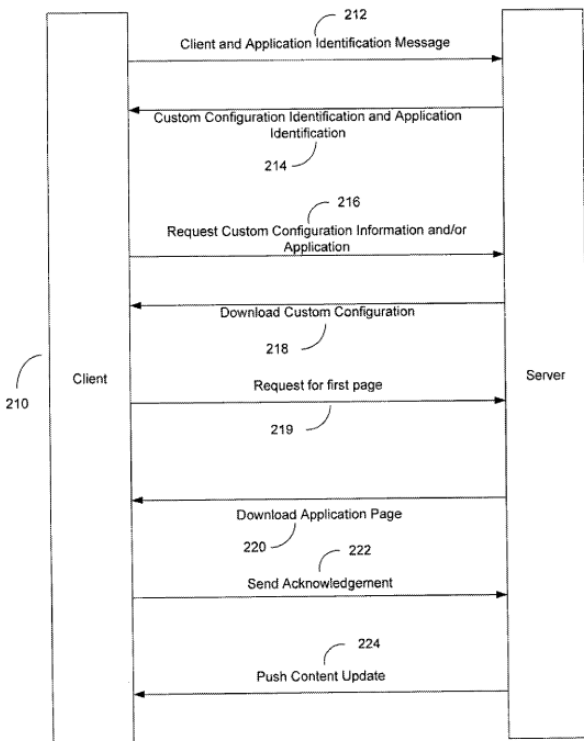


FIGURE 2A

200B

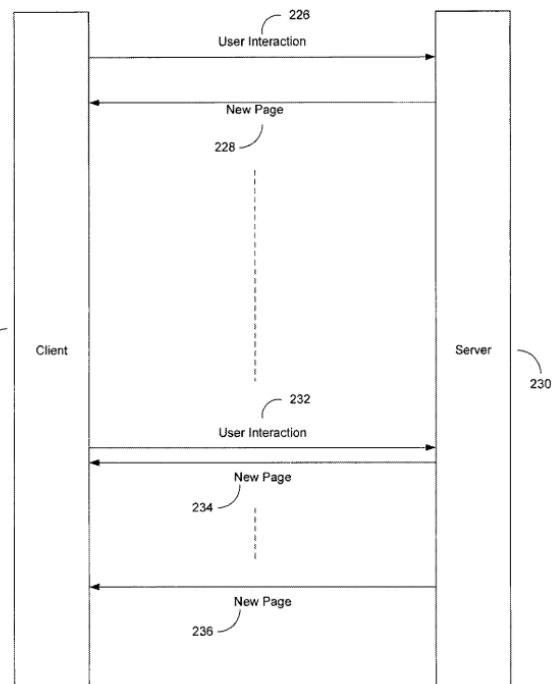
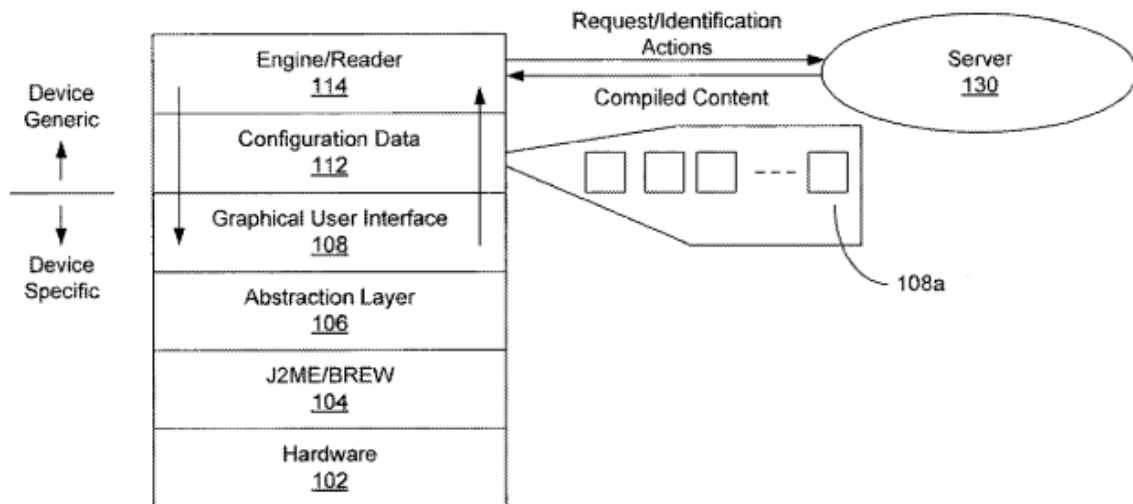


FIGURE 2B

'865Pat, FIGs. 2A & 2B.

The custom configuration “customizes the look and feel of the content displayed on the wireless device from execution of the requested application.” *Id.*, 18:41-43. The specification explains that a graphical user interface (“GUI” shown in Figure 1B below) in the wireless device includes rendering blocks (*id.*, 7:21-34) that use the custom configuration “to customize the appearance of the requested application and represents programming and stored content of the graphical user interface 108” (*id.*, 7:59-62). “In other words, configuration data may be a set of low level instructions for preprogramming a plurality of rendering blocks 108a of the graphical user interface to operate and render data (e.g., ‘look’) a certain way.” *Id.*, 7:62-66.

100B



'865Pat, FIG. 1B.

The specification further explains (and is shown as step 220 in Figure 2A above) that the wireless device “may receive a compiled content from the server 130 that includes a series of basic commands for rendering the requested application.” *Id.*, 7:42-45, 13:11-25. The GUI in the wireless device receives this compiled content and “render[s] the page of the application based on the received basic commands and the customized preprogrammed plurality of rendering blocks.” *Id.*, 10:64-11:5. The user may interact with the page, causing a new (second) page to be sent from the server, as shown above at steps 226 and 228 of Figure 2B. *Id.*, 14:17-31, FIG. 2B.

C. *The Prosecution History*

The '865 patent issued on January 24, 2012 from U.S. Patent Application 11/888,799, filed August 1, 2007. The application that led to the '865 patent was prosecuted under the pre-AIA statutory regime and included 41 claims. '865*FH*, 231 (EX. 1004). The initial information disclosure statement (IDS) filed on August 1, 2007 only included one patent. *Id.*, 255. A subsequent IDS filed, almost two years later, on June 2, 2009 included only three patents and patent applications, and five non-patent literature documents. *Id.*, 249. A non-final rejection was mailed on June 25, 2010, in which the Examiner rejected all 41 claims using a single published patent application. *Id.*, 231. Additionally, the Examiner listed just four references in the notice of references cited. *Id.*, 242. In response, applicant

heavily amended some claims, while canceling others, on December 21, 2010. *Id.*, 264. The amended claims were again rejected in a final rejection on February 7, 2011 by the Examiner on lack of written description, enablement, and prior art grounds (maintained from the earlier non-final rejection). *Id.*, 532. In the final rejection, the Examiner noted they were unable to find support for “said plurality of content items using both said screen description and said custom configuration.” *Id.*, 533. To overcome the rejection, applicant amended the claims in a request for continued examination on August 8, 2011 to simply replace “screen description” with “page description.” *Id.*, 549. Without providing any reasons, the Examiner allowed all outstanding claims on September 7, 2011. *Id.*, 568.

V. LEVEL OF ORDINARY SKILL IN THE ART

A person of ordinary skill in the art (“POSITA”) at the time of the earliest available priority date (August 1, 2007) for the ’865 patent would have had a bachelor’s degree in electrical or computer engineering, or a closely related scientific field such as computer science, and two years of work experience with remote page display rendering and/or remote server applications. Alternatively, any lack of work experience could be remedied with additional education (*e.g.*, a Ph.D.) concentrating on multimedia content transmission and/or remote server applications, and likewise, a lack of education can be remedied with additional

work experience in multimedia content transmission and/or remote server applications (e.g., 4-5 years). *Bederson*, ¶38.

VI. CLAIM CONSTRUCTION

Petitioner is unaware of any “prior claim construction determination” related to the ’865 patent. *See* 37 C.F.R. §42.100(b). Given the close correlation and substantial identity between the prior art references and the challenged claims, Petitioner believes that no express constructions of the claims are necessary to assess whether the prior art reads on the challenged claims. Thus, the claims should be given their plain and ordinary meaning.⁴ *Phillips v. AWH Corp.*, 415 F.3d 1303, 1312–13 (Fed. Cir. 2005) (en banc).

VII. THE PRIMARY PRIOR ART REFERENCES

A. *Overview of Hariki*

Hariki is directed to “a mobile device incorporating a resource application program interface for downloading user interface components for a resident application.” *Hariki*, Abstract. *Hariki* explains that embodiments may be

⁴ Petitioner reserves all rights to raise claim construction arguments and other arguments in any litigation concerning the ’865 patent. For example, comparing the claims to the accused products in the litigation may raise controversies that require construction of certain claim terms.

implemented in a client-server arrangement in a network “that incorporate[s] a telecommunications or similar wireless network system.” *Id.*, ¶18. Figure 1 illustrates this network system. The system comprises mobile devices 108 and 109, such as cellular phones “configured to communicate with one or more client and/or server computers that are directly or indirectly coupled to network 110.” *Id.*, ¶20. *Hariki* explains that “a server computer 102 operated by the service provider or a third party vendor is a content provider server computer that provides content data, application programs, diagnostic tools, program components, or any other content or executable objects to the mobile device 108.” *Id.*, ¶21. The server can execute programs, such as a web server process, to send HTML pages to the mobile devices. *Id.* The server can also “generate a customized user interface for a plurality of different makes and types of mobile devices” that “allow a user to customize the ‘look and feel’ or application program environment of a device by altering display and/or sound output aspects of the device, such as backgrounds, title bars, buttons, alert sounds, and so on.” *Id.*, ¶24. This customized “look and feel” is delivered in a content package and is “specific to the type of device, manufacturer of the device, operating system, application programs, and other relevant information regarding the mobile device.” *Id.*, ¶28.

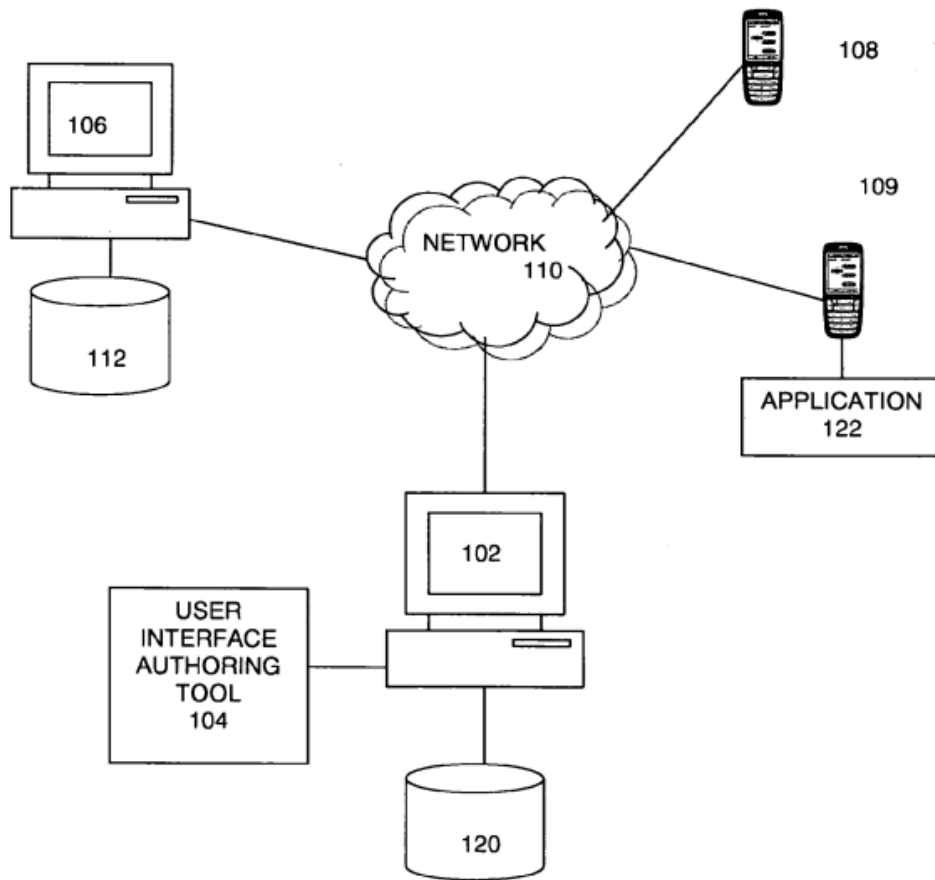


FIGURE 1

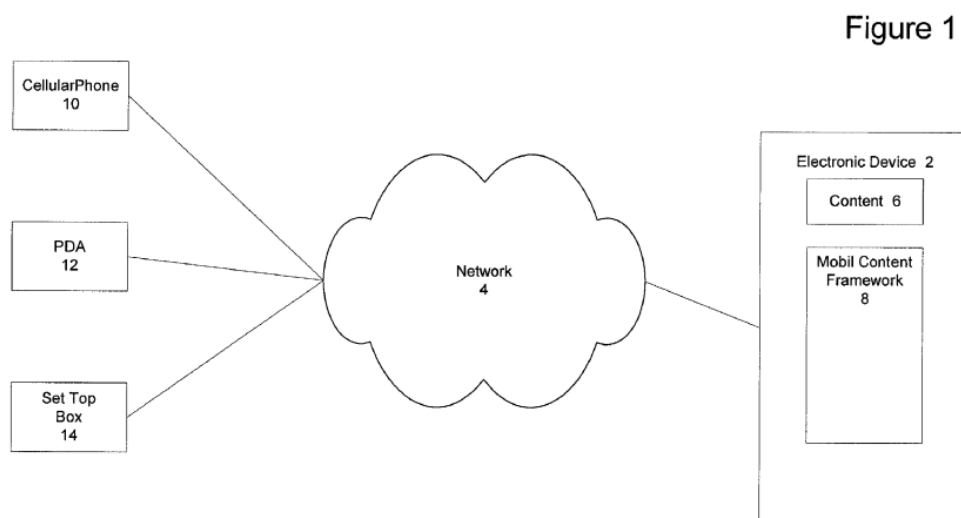
Hariki, FIG. 1.

B. Overview of *Harris*

Harris discloses “a platform that enables a content developer to distribute uniform content to multiple [sic] types of requesting mobile devices.” *Harris*, Abstract. *Harris* explains that “[b]y authoring content in a generic markup language and using the technologies included in the Mobile Content Framework (MCF), the content developer is able to provide content to multiple types devices

[sic] without providing different versions of the content.” *Id.* *Harris* further explains that “[c]ontent is abstracted in a manner specifically tailored to take into account the limited resources of certain devices such as mobile devices” and that “[c]ontent is generated specifically for each device, both from a display standpoint and a content navigation standpoint.” *Id.*, ¶7. Using the *Harris* platform, the “interface may be dynamically personalized to the taste of the individual.” *Id.* The content is first generated in a generic markup language and then translated “into a device appropriate language for a requesting mobile device.” *Id.*

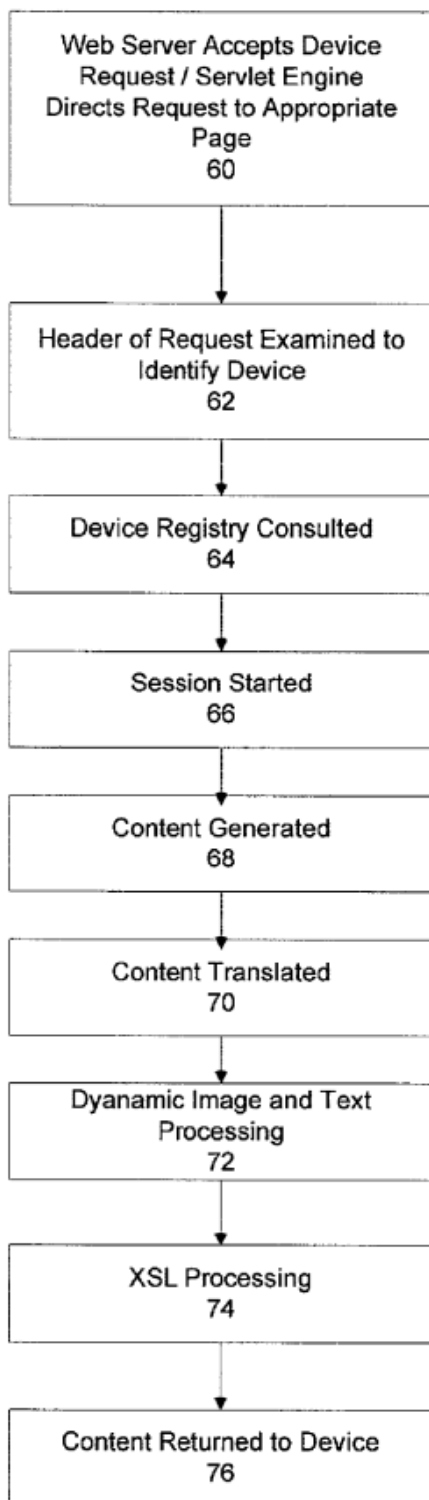
Harris discloses its environment with reference to Figure 1, reproduced below. The environment includes a server (electronic device 2) with content and an MCF. *Id.*, ¶15. The server is interfaced to a plurality of wireless devices through a network. *Id.*



Harris, FIG. 1.

Harris explains, with reference to Figure 3, below, that the server receives a request from a wireless device (step 60) and the device type is determined from the request (step 62). *Id.*, ¶24. The server determines device capabilities (step 64) and a session with an application is started to store application data (step 66). *Id.* *Harris* explains that the data can come from applications, such as a “web site, a static file, a Servlet or a Java Server Page.” *Id.*, ¶26. A wireless device independent template is generated in the WAX (XML) language (step 68) and subsequently translated to a device-specific language in step 70. *Id.*, ¶24. After further processing, “customized content is then returned to requesting device (step 76).” *Id.*

Figure 3



C. Overview of *Dees*

Dees is directed to “a method of styling a user interface for a device” (*Dees*, ¶1) in a client-server system (*id.*, ¶33). *Dees* explains that “[t]he user interface module 420 [of a client device] activates the networking module 410 to cause it to download the first style sheet document 211 from the server 101.” *Dees*, ¶51. *Dees* further explains that “[h]aving downloaded the document 211, the networking module 410 feeds the document 211 to the user interface module 420. The user interface module 420 then uses the style information contained in the document 211 in generating the user interface.” *Id.* Furthermore, *Dees* discloses the “style sheet document [] can comprise style information in any language suitable for marking up style information, such as CSS or DSSSL. It is also possible to use XML to mark up style information.” *Id.*, ¶53. Additionally, *Dees* discloses “HyperText Markup Language HTML also provides [] ways to provide a user interface definition.” *Id.*, ¶37. *Dees* further states that “[u]sing a language such as CSS or DSSSL, style properties such as the font or the color of a button or the position of a checkbox on the screen can easily be defined.” *Id.*, ¶3.

Figure 1 of *Dees*, shown below, “schematically shows a system 100 according to the invention.” *Id.*, ¶33. The system comprises server 101 connected to a network 110, along with laptop computer 120, a desktop computer 125, a mobile phone 130 and a handheld computer 135. *Id.* *Dees* explains that the “[t]he

mobile phone 130 and the handheld computer 135 have a wireless connection to the network 110.” *Id.*, ¶34.

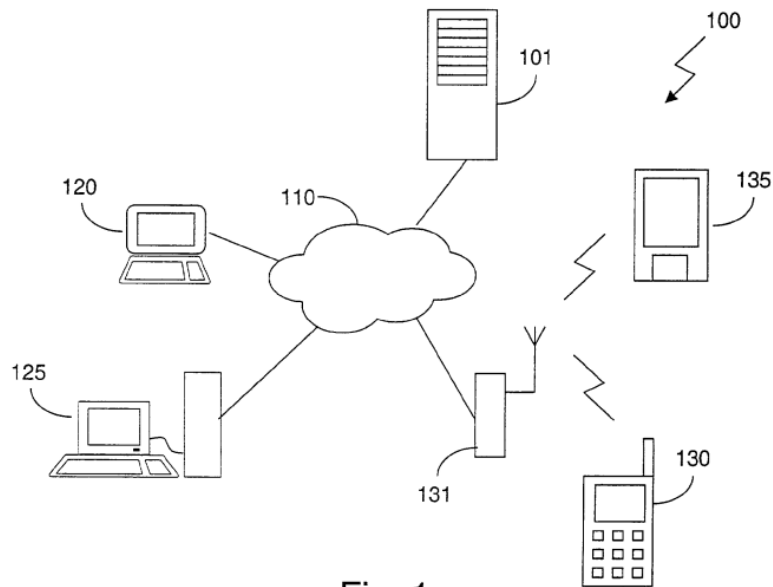


Fig. 1

Figure 4 of *Dees*, shown below, “schematically shows a client device 400 in more detail,” including “a networking module 410 which obtains the user interface definition document 210 from the server 101, via the network 110. The document 210 is then fed to a user interface module 420, which processes the document 210 in order to generate a user interface. A rendering module 450 then presents the generated user interface to a user of the client device 400 using output means such as a display 452 or a speaker 451. Style information is applied to this user interface, as will become apparent below.” *Id.*, ¶¶48-49.

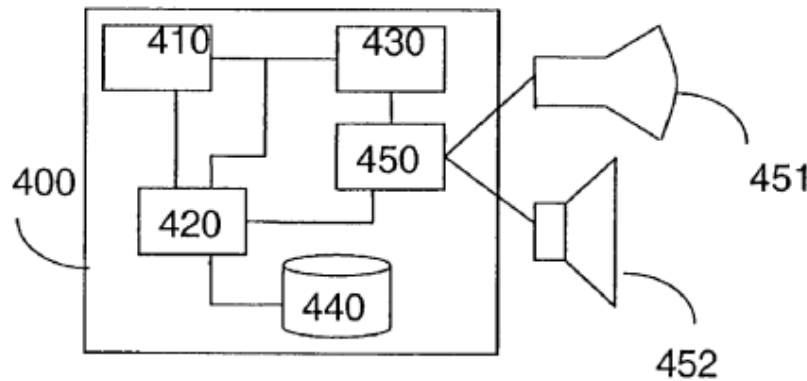


Fig. 4

D. Other Evidence Regarding the State of the Art

The prior art in Exhibits 1008-1021 and 1024-1030 further reflects the state of the art, level of ordinary skill in the art, common knowledge in the art, and/or common sense in the art, and is therefore also relevant to the background of the '865 patent and the invalidity analysis herein. *Bederson*, ¶91.

VIII. DETAILED EXPLANATION OF THE GROUNDS

A. Ground 1: *Hariki* in view of *Harris* renders obvious Claims 1-20, 31, 34.

1. Claim 1

a. [1pre]

To the extent the preamble is limiting, *Hariki* discloses this feature. *Bederson*, ¶¶105-113. With reference to Figure 1 below, *Hariki* discloses, “server computer 102... provides **content data, application programs, diagnostic tools, program components, or any other content or executable objects to the mobile**

device 108.... For example, server 102 can execute[] **a web server process to serve web pages,**⁵ such as HTML files over a network to the mobile device. *Hariki*, ¶21. Furthermore, the server “can generate a customized user interface for a plurality of different makes and types of mobile devices.” *Id.*, ¶24. *Hariki* explains that “the customized user interface is referred to as a ‘UI skin’” to “allow a user to customize the ‘look and feel’ or application program environment of a device by altering **display and/or sound output aspects of the device.**” *Id.*

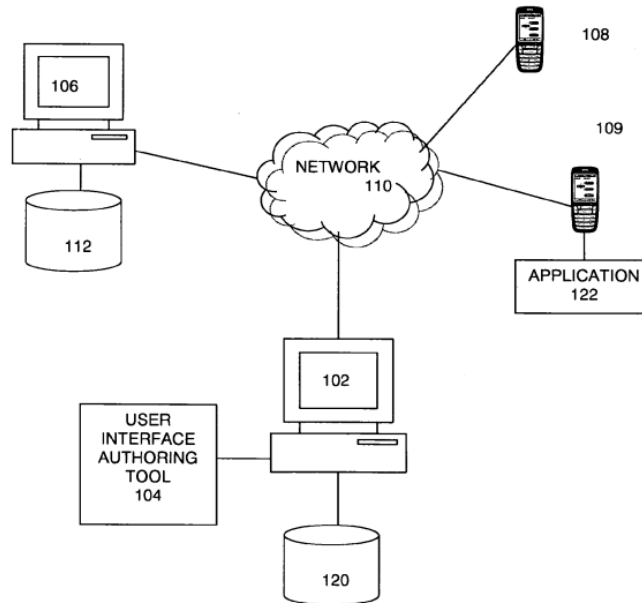


FIGURE 1

Hariki, FIG. 1.

⁵ Unless otherwise stated, all emphases are added.

Because the *Hariki* server generates executable content, such as a UI skin that alters the “look and feel” of the display and/or sound of mobile devices, and provides content data to mobile devices, such as by executing a web server process to serve web pages to mobile devices, POSITAs would have understood that the server processes data for mobile devices. *Bederson*, ¶108.

Furthermore, *Hariki* discloses mobile devices 108 and 109 may be cellular phones. *Id.*, ¶20. With reference to Figure 1 above, *Hariki* explains that “mobile devices 108 and 109 communicate with each other and other mobile devices over the **wireless portion of network 110.**” *Id.* POSITAs would have thus understood that the mobile devices are wireless devices. *Bederson*, ¶¶109-110.

Hariki further discloses “a **method** of generating and downloading UI skins.” *Hariki*, ¶36. As shown in Figure 4 below, “the method begins in step 401 with selection and retrieval of the appropriate resource profile for the device from the resource profiles” and concludes with “[t]he UI content package... downloaded to the target mobile device” in step 410. *Id.*, ¶36. *Hariki* explains that the UI content package (in other words, the UI skin) alters the display of the mobile device. *Id.*, ¶24; *Bederson*, ¶¶111-112.

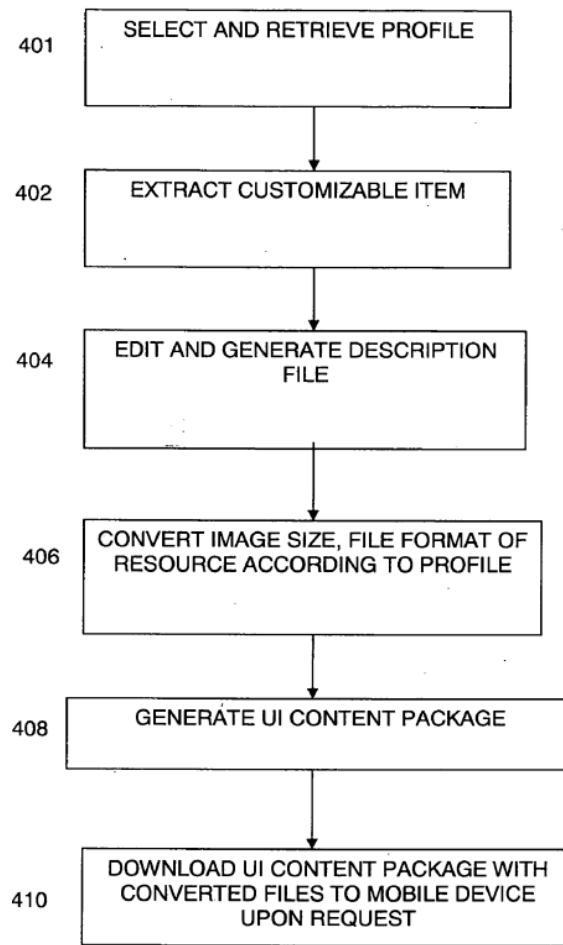


FIGURE 4

Hariki, FIG. 4.

Accordingly, *Hariki* discloses a method of generating UI skins and providing content data by a server (i.e., “server implemented method for processing data”) to a mobile device (i.e., “for a wireless device”). *Bederson*, ¶113.

b. [1a]

Hariki discloses this feature. *Bederson*, ¶¶114-132. As explained above, *Hariki* discloses a “server computer 102 operated by the service provider or a third party vendor is a content provider server computer that provides content data, **application programs**, diagnostic tools, **program components**, or any other content or **executable objects** to the mobile device 108.” *Hariki*, ¶21. Furthermore, *Hariki*’s server “can be a World-Wide Web (WWW) server that stores data in the form of web pages and transmits these pages as Hypertext Markup Language (HTML) files over the Internet 110 to the mobile devices 108 and 109.” *Id.* The WWW server “executes a web server process to serve web pages.” *Id.* The mobile device “runs a web browser program to access the web pages served by server computer 102 and any other available content provider or supplemental server, such as computer 106.” *Id.* Accordingly, POSITAs would have understood that the web browser running on the mobile device sends a request to the web server process, which receives the request, to access the corresponding HTML files (i.e., “receiving from the wireless device a request for an application program”). *Bederson*, ¶117. POSITAs would not have understood a distinction in “accessing” and “requesting” the web server process to receive the HTML files. *Id.*, ¶118.

Hariki further discloses “mobile devices 108 and 109 can represent devices made by different manufacturers, different models of devices made by the same

manufacturer or any **type of devices** that have different user interface elements from one another.” *Hariki*, ¶20. *Hariki* explains “server computer 102 is operated by a content provider that can generate a customized user interface for a plurality of different makes and types of mobile devices” (*id.*, ¶24), and “[v]arious resource profiles and resource files for the different mobile devices can be stored in a database” (*id.*, ¶25). *Hariki*’s server “downloads the user interface to the appropriate mobile device upon request of the mobile device user.” *Id.*, ¶25. Moreover, in order to download the appropriate interface, *Hariki* discloses “method begins in step 401 with selection and retrieval of the appropriate resource profile for the device.”⁶ *Id.*, ¶36. The resource profile is selected “depending upon the device model selected by the content provider.” *Id.*, ¶32. POSITAs would have understood that to provide requested content to a mobile device, after checking stored device profiles, the server would have determined what type of mobile device is making the request from the information received from the mobile device. *Bederson*, ¶121. Furthermore, POSITAs would have understood that in order for the server to determine what type of mobile device is accessing it, the device would have sent its type or other identifying information to identify the

⁶ “Resource profiles are defined by device manufacturers for each model of mobile device.” *Hariki*, ¶19.

type. *Bederson*, ¶122. Sending device characteristics would have identified a type of device, such as large screen or small screen, or sending a manufacturer model number would have identified the type of mobile device. Accordingly, POSITAs would have understood that *Hariki* discloses the mobile device identifying what type it is (i.e., “including an indication of a type of the wireless device”). *Bederson*, ¶123.

Harris, like *Hariki*, discloses “a platform that enables a content developer to distribute uniform content to multiple types of requesting mobile devices.” *Harris*, Abstract. As explained above in Section VII.B, *Harris* describes that a wireless device template independent of mobile device type is generated in the WAX (XML) language and subsequently translated to a device specific language. *Harris*, ¶24. After further processing, “customized content is then returned to requesting device (step 76).” *Id.* *Harris* explains that the customized content is in “a language the requesting device can understand” (*id.*, ¶24), “such as WML, HDML, HTML, compact-HTML and Palm webClippings” (*id.*, ¶3). Thus, *Harris* discloses a detailed process of sending HTML files to a client mobile device where the HTML files are tailored to the particular mobile device. *Bederson*, ¶125.

POSITAs would have understood that the methods disclosed by *Harris* were compatible with and could be adapted and applied to *Hariki* because *Harris*’ platform is of the same general architecture as *Hariki*’s client-server system.

Bederson, ¶126. Thus, to the extent that PO contends that *Hariki* does not disclose that the mobile device sends a request to the server that indicated its device type, POSITAs would have been motivated to combine the client-server system of *Hariki* with the teachings of *Harris* directed to serving HTML pages tailored to a particular mobile device. *Id.*, ¶127. In particular, to the extent PO contends that while *Hariki* discloses sending HTML files to a mobile device, it does not expressly disclose how the HTML files are generated, POSITAs would have been motivated to find ways of generating and sending information from applications such as a web server process customized to a particular mobile device and looked to *Harris*. *Id.*, ¶127.

Moreover, POSITAs would have had a reasonable expectation of success in implementing *Harris*' content generation scheme with *Hariki*'s customization scheme. *Bederson*, ¶128. Should PO contend that *Harris*' disclosure that an "interface may be dynamically personalized to the taste of the individual" (*Harris*, ¶7) does not explain how this is accomplished other than "a client API to interact directly with a personalization service" or "personalization engines" (*id.*, ¶24), *Hariki* fills this gap by explaining how "[r]esource profiles are defined by device manufacturers for each model of mobile device. The resource profiles specify the composition, format, and appearance of one or more user interface elements displayed or executed by the mobile device." *Hariki*, ¶19. *Hariki* further explains

how to customize the user interface (“UI skin”) by “altering display and/or sound output aspects of the device, such as backgrounds, title bars, buttons, alert sounds, and so on.” (*id.*, ¶24), and “UI skins can be downloaded by the user and installed on the mobile device to alter the default UI or permanently change the UI of the device” (*id.*). *Bederson*, ¶129. Therefore, POSITAs would have been motivated to combine *Hariki* and *Harris* because, while each discloses sending content specific to a mobile device, each describes in detail different aspects of the customization. *Id.*, ¶130.

Harris discloses “when an electronic device, such as a web server accepts a connection and an HTTP request from a mobile device... [t]he headers of the HTTP request are examined by the MCF 8 in order to determine the type of device.” *Harris*, ¶24.

Accordingly, the *Hariki-Harris* system discloses receiving by a web server process a content request from a mobile device (i.e., “receiving from the wireless device a request for an application program”), where the request includes a header containing the mobile device type (i.e., “said request including an indication of a type of the wireless device”). *Bederson*, ¶132.

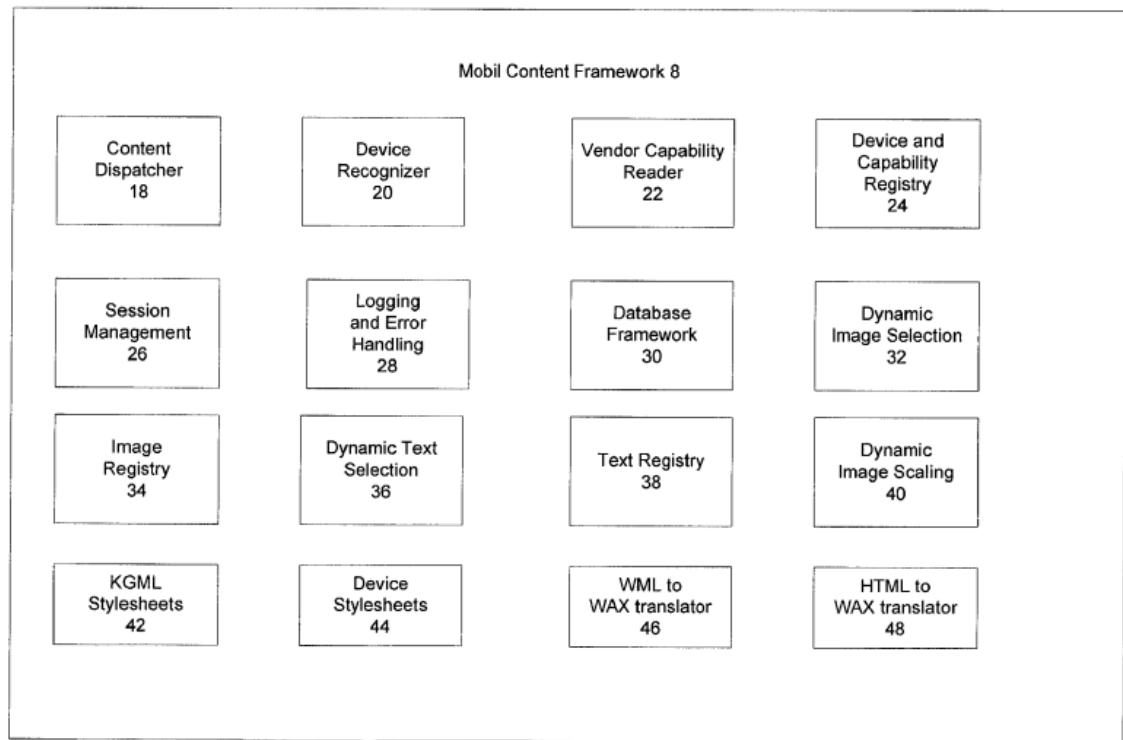
c. [1b]

Hariki in combination with *Harris* discloses this feature. *Bederson*, ¶¶133-141. The ’865 patent discloses “the template engine provides a high level template

(e.g., extensible markup language (XML) format) that includes the merged screens of the requested application and dynamic data provided by the business logic.”
'865Pat, 3:38-42.

In the *Hariki-Harris* system, *Harris* discloses, referring to Fig. 2 below, that the MCF's “dynamic image selection component 32 uses the image registry 34, and the dynamic text selection component 36 uses the text registry 38 to determine the best content to deliver to a requesting device at any given time.” *Harris*, ¶22. Additionally, “MCF 8 includes a **WAX stylesheets component 42** [and] a **device stylesheets component 44**.” *Id.*, ¶22.

Figure 2



Harris, FIG. 2.

Harris explains that “[t]he **WAX stylesheets component 42 dictates the presentation of the WAX content**, and the device stylesheets component 40 tailors the presentation of the content to the requesting device based on the attributes possessed by the requesting device.” *Id.*, ¶22. Moreover, *Harris* explains that “WAX is designed to enable the content developer to describe content at a more **abstract level** than that used in individual protocols. The greater level of abstraction **enables the subsequent conversion of WAX into the languages used by the requesting mobile devices such as WML, HDML and HTML.**” *Id.*, ¶7. Accordingly, *Harris* discloses generating WAX stylesheets, which are abstract and not specific to a mobile device (i.e., “wireless device generic template”) that include content (i.e., “plurality of content items”) tailored by the device stylesheets. *Bederson*, ¶137.

In *Harris*, the template generation process begins when “a web server accepts a connection and **an HTTP request from a mobile device...** [where] headers of the HTTP request are examined by the MCF 8 in order to determine the type of [requesting] device,” in response to which, a “registry of devices is consulted to determine device capabilities (screen size, browser version, etc.).” *Id.*, ¶24. WAX is then translated “into a language the requesting device can understand” and customized content is returned to requesting device. *Id.*, ¶24. Accordingly, *Harris* discloses generating WAX stylesheets with content in

response to receiving a request from a mobile device (i.e., “in response to receiving said request”). *Bederson*, ¶138.

Additionally, in the *Hariki-Harris* system *Hariki* discloses “server computer 102... provides content data, **application programs**, diagnostic tools, **program components**, or any other content or executable objects **to the mobile device 108**.... For example, **server 102 can execute[] a web server process** to serve web pages,”⁷ such as HTML files over a network to the mobile devices. *Hariki*, ¶21. Accordingly, the server in the *Hariki-Harris* system executes application web server process to generate content for the mobile device. *Bederson*, ¶140.

Accordingly, *Harris* discloses a server receiving a HTTP request from a mobile device (i.e., “receiving said request”) and in response, executing an application, such as the web server process, (i.e., “executing said application program to generate”), to generate a WAX stylesheet (i.e., “to generate a wireless device generic template”) that includes content for presentation on the mobile device (i.e., “including a plurality of content items”). *Bederson*, ¶141.

d. [1c]

Hariki in combination with *Harris* discloses this feature. *Bederson*, ¶¶142-148. *Hariki* explains that “[m]any different aspects of a mobile device may be

⁷ Unless otherwise stated, all emphases are added.

customizable however, **such as the format** and display of menus, commands, subwindows, and so on,” and that the **server can provide the customized UI to the mobile device**. *Hariki*, ¶26. *Hariki* further explains that to alter the UI skin, a “UI package generator produces a downloadable UI content package consisting of [a] description file and [] converted resource files.” *Hariki*, ¶19. POSITAs would have understood that the converted resource files and the description file in *Hariki*’s UI content package corresponded to “custom configuration” and “identification of [the] custom configuration,” respectively. *Bederson*, ¶143. The UI content package includes a converted resource such as a Flash, PNG, or JPEG object used by engines that “process the applicable resource using the appropriate playback format,” such as the Flash engine for an applet or JPEG engine for a photograph. *Id.*, ¶43. The engine “converts each resource file to a format or embodiment that is compatible with the application 506,” which is “in terms of parameters such as **image size, color, position, and so on.**” *Id.*, ¶47. Accordingly, *Hariki* discloses a server “sending a custom configuration to the wireless device.” *Bederson*, ¶144.

Furthermore, *Hariki* discloses “server computer 102... is a content provider server computer that provides **content data, application programs, diagnostic tools, program components, or any other content or executable objects** to the mobile device 108.” *Hariki*, ¶21. The server “can be a World-Wide Web (WWW)

server that stores data in the form of web pages and transmits these pages as Hypertext Markup Language (HTML) files over the Internet 110 to the mobile devices 108 and 109.” *Id.* POSITAs would have understood that in the context of accessing a WWW server to receive HTML files, the *Hariki-Harris* system would have sent the customized UI when executing the web server process. *Bederson*, ¶145. Furthermore, the configurable aspects of the user interface may be configurable buttons and sounds. *Id.*, ¶24; *Bederson*, ¶146. *Hariki* explains that the converted resources in the UI content package are “files, links to files, and/or data or program objects **associated with the configurable aspect of the user interface** for each mobile device.” *Id.*, ¶31. POSITAs would have thus understood that, when the web server process is accessed, the converted resources in the UI content package configure the configurable aspects of the UI, by e.g., changing a display image or color. *Bederson*, ¶147. Accordingly, *Hariki* discloses “said custom configuration being specific to said application program.” *Id.*, ¶147.

Accordingly, *Hariki* discloses a server running a web server process sending converted resource files as part of a UI content package to a mobile device (i.e., “sending a custom configuration to the wireless device”), where the converted resources are specific to an application program on the server (“said custom configuration being specific to said application program”). *Bederson*, ¶148.

e. [1d]

Hariki in combination with *Harris* discloses this feature. *Bederson*, ¶¶149-153. The '865 patent explains that “wireless device... communicates with the server 130 **via a device generic syntax** to read the basic commands of a page description.” ’865*Pat*, 7:35-38. Furthermore, “the requested application content is received as compiled content that includes a series of basic commands representing a page description. Basic commands are precompiled by the server 230 and ready for rendering by the client.” *Id.*, 13:15-19. POSITAs would have understood that device specific HTML files are written in a generic syntax but include tags that are ready for rendering. *Bederson*, ¶150.

Harris discloses “[t]he MCF 8 dynamically translates WAX into other markup languages using XML-based technologies. **Content is translated from WAX to** languages like WML, HDML and **HTML**. The content is not only translated into a specific language appropriate for the device, but also (unlike traditional XSL stylesheets) is **tailored based upon the attributes of the requesting device**.” *Harris*, ¶18. *Harris* further explains that in the translation process, elements (or tags) in the WAX file are translated to be rendered in different ways, specific to the mobile device. *Bederson*, ¶¶152. For example, “[t]he <wax:button> element is displayed as a ‘soft-key’ for WAP devices, but as a ‘link’ for devices which understand only HTML.” *Harris*, ¶30. Accordingly, the *Hariki*-

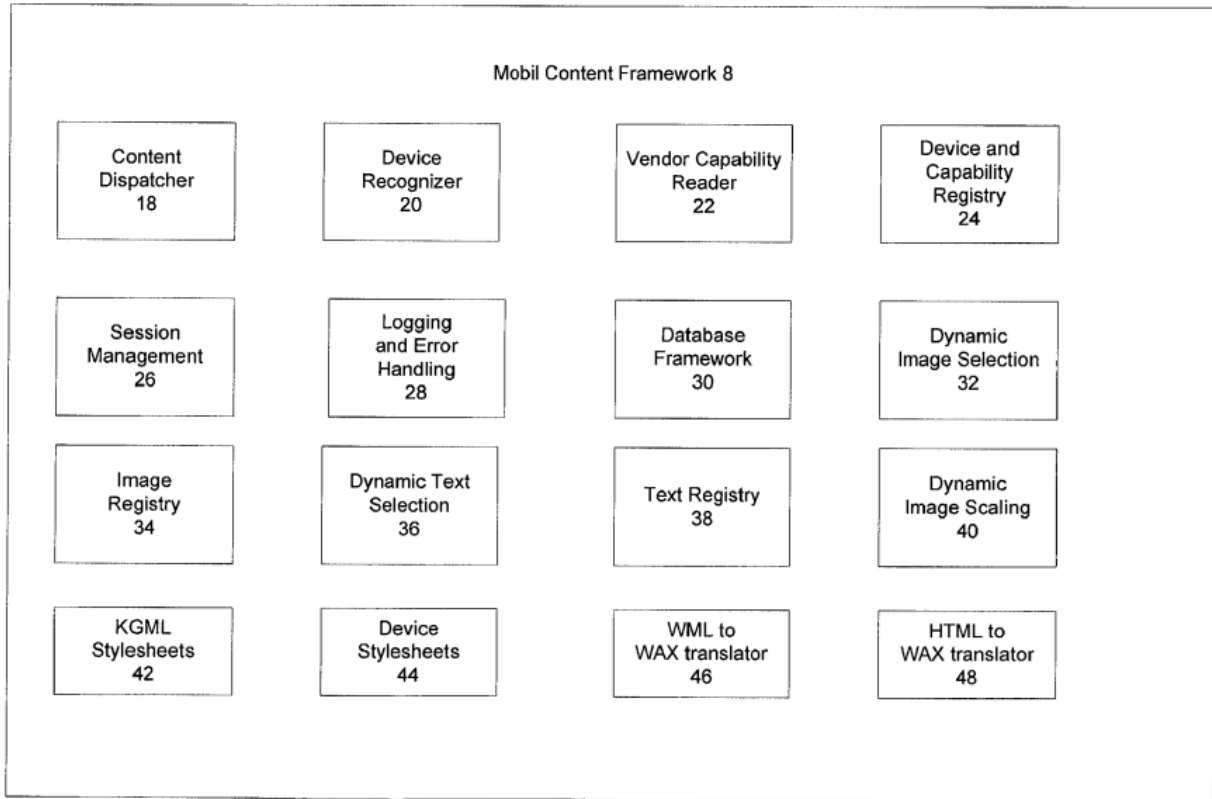
Harris system discloses generating HTML files (i.e., “generating a page description”) from WAX content from WAX and device stylesheets (i.e., “based on said wireless device generic template”) that is tailored based upon attributes of the requesting device (i.e., “based on capability of the wireless device”). *Bederson*, ¶153.

f. [1e]

Hariki in combination with *Harris* discloses this feature. *Bederson*, ¶¶154-166. In the *Hariki-Harris* system, *Harris* explains that a generic WAX template “is easily extended and can be translated into a variety of different mobile device markup languages.” *Harris*, ¶7. *Harris* explains that WAX “is ‘device-smart’ in that it detects the idiosyncrasies of each device,” by “adjust[ing] to the limitations and features of each device.” *Id.*, ¶17. Before sending content to a mobile device, “[t]he WAX language may be dynamically translated to a requesting device’s native language” (*id.*, ¶18), thereby “tailor[ing] content to specific requesting devices” (*id.*, ¶19). To do this, a Mobile Content Framework (“MCF”) “includes registries listing device capabilities which are utilized to provide appropriate amounts and types of content to the requesting device.” *Id.* With reference to Figure 2, below, “[a]ll mobile devices and their capabilities are kept in a device and capability registry 24.” *Id.*, ¶21. The device and capability registry 24 provides

“attributes such as screen size, color depth, browser version and type, and translation rules” based on device type and application. *Id.*; *Bederson*, ¶156.

Figure 2



Harris, FIG. 2.

Accordingly, the *Hariki-Harris* system tailors the HTML pages to the specific attributes (i.e., the claimed “rendering capabilities”) of the mobile device. *Bederson*, ¶157.

The '865 patent explains that the page description, i.e., render commands,⁸ are “written in a syntax that is device generic regardless of the device type but whose parameters are tailored based on the wireless device capabilities.” *'865Pat*, 13:19-22. The “syntax” may include:

a description of the scrolling area (e.g., starting and ending vertical positions), the horizontal and vertical coordinates, the width, the height, the type of component to be displayed (e.g., text, image, video, audio and the like), the unique identification of the rendering block to be used to render the component, related parameters for the rendering block and for display components (e.g., version number of the image) and the like.

Id., 13:27-34.

The '865 patent shows an example of basic commands in a “compiled page description” in Figure 4, reproduced below. The '865 patent explains:

Each basic command may describe a given component on the page of the requested application to be rendered and includes a rendering block to be used. For example, a basic command 410 may be a description for rendering an image whereas a basic command 430 may be the description for rendering a video clip. Accordingly, a collection of basic commands 410, 430 and 490 forms a single unified page to be rendered by the wireless device.”

⁸ The '865 patent does not disclose how to distinguish a “low level rendering command” from any other type of rendering command. Thus, to the extent that “low level rendering command” is disclosed, the rendering commands in HTML files disclose this feature. *Bederson*, ¶158. Petitioner reserves all rights to raise §112 arguments in any litigation involving the '865 patent.

Id., 15:59-67.

400

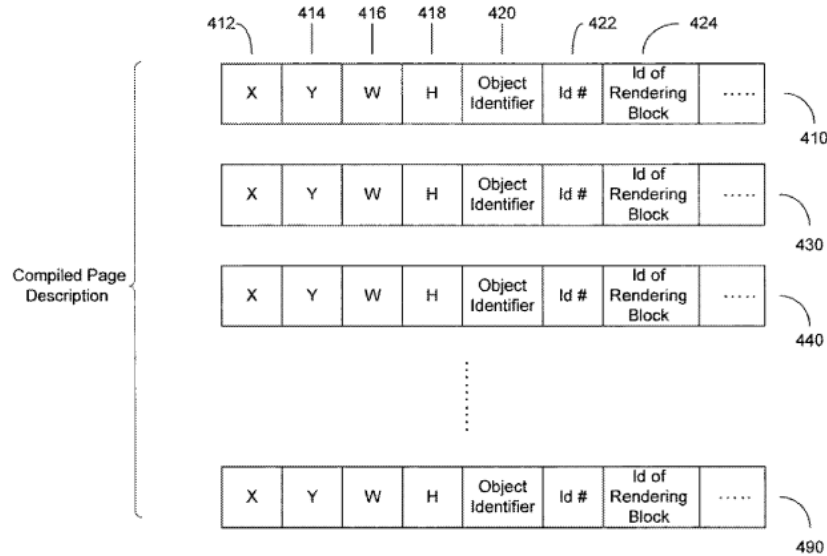


FIGURE 4

'865Pat, FIG. 4.

In the *Harris-Hariki* system, *Harris* discloses such basic rendering commands. *Bederson*, ¶160. With respect to WAX, the generic high-level syntax used to make the HTML pages, *Harris* proves an example in the Table below:

```
<?xml version=" 1. 0" encoding="utf-8"?>
<wax:wax xmlns:wax="http://www.kargo.com/wax" version="O.9">
  <wax:doc version="1.0">
    <wax:title>NY Nightlife</wax:title>
    <wax:block id="splash">
      <! --EXAMPLE 1 -->
      <wax : button href="index?a=rnain"
keytype="accept" type="go" labelid="enter" />
      <wax:p align="center">
        <! --EXAMPLE 2 -->
        <wax: img srcid="logo" alt="My Nightlife"
          border="O" />
    </wax:block>
  </wax:doc>
</wax:wax>
```

-continued

```

        <wax:br/>
        <!--EXAMPLE 3 -->
        <wax: text id="welcome" />
        <!--EXAMPLE 4 -->
        <% if ((String)session.getAttribute("lang") ==
            null)
            out.println ("<wax:a
            href=\ "index?a=chlang\ ">Choose
            Language</wax:a>");
        %>
        <wax:br/>
        <wax:br/>
        </wax:p>
        </wax:block>
        </wax:doc>
    </wax:wax>

```

Harris, ¶29. POSITAs would have looked at the above Table and understood that the WAX language used a syntax that was device generic. *Bederson*, ¶161. *Harris* explains that commands such “<wax:button>” are rendered differently in different devices, and text may be displayed in different languages. *Harris*, ¶30; *Bederson*, ¶161.

Moreover, POSITAs would have also understood that HTML itself is in a syntax that is generic to the mobile device. *Bederson*, ¶162. Indeed, HTML is a standardized language that has the same syntax no matter what device is receiving it. *Id.*, ¶162. Accordingly, in the *Hariki-Harris* system the content data sent to the mobile device is expressed in WAX or HTML, both using a syntax that is device independent. *Id.*, ¶162. POSITAs would also have had a reasonable expectation of success of displaying content on different mobile devices using a generic syntax

because it was well within POSITAs' knowledge that different client devices could receive HTML-based files from a server through the World Wide Web and render them for display. *Bederson*, ¶162. Accordingly, in the *Hariki-Harris* system, content data received by the mobile device for rendering is expressed in WAX or HTML, both using a syntax that is device independent (i.e., "that is of a syntax that is wireless device generic."). *Bederson*, ¶162.

Furthermore, POSITAs would have understood that HTML files include tags that include low level commands to render content on a client device. *Bederson*, ¶163. POSITAs would also have understood that HTML files sent by a web server are rendered into pages in a web browser on the mobile device. *Id.*, ¶163. Moreover, POSITAs would have understood that HTML files are a compilation of tags (e.g., the tags are aggregated into a file) that direct a web browser to display content on device. *Bederson*, ¶164. Indeed, POSITAs would have understood that such a compilation of commands is how the '865 patent describes the page description. *Id.*, ¶164; citing '865*Pat*, 17:66-18:1 ("It is appreciated that **basic commands may be low level compilation operable** to render application content on the client 510 using the rendering blocks of the wireless device."). Accordingly, POSITAs would have understood that *Hariki* discloses a server that sends HTML files that include low level commands to a mobile device. *Bederson*, ¶165.

Accordingly, the *Hariki-Harris* system discloses tailoring HTML pages to specific attributes of mobile devices, including providing rendering commands, including low level rendering commands, (i.e., “said page description ...of said wireless device”), where content data sent to the mobile device is expressed in WAX or HTML using a syntax that is device independent (i.e., “but that is of a syntax that is wireless device generic”). *Bederson*, ¶166.

g. [1f]

Hariki in combination with *Harris* discloses this feature. *Bederson*, ¶¶167-175. As explained above, the *Hariki-Harris* system sends HTML files (i.e., “page description”) to the mobile device. *Id.*, ¶168.

The '865 patent explains that a “requested application content is received as **compiled content that includes a series of basic commands representing a page description.**” ’865*Pat*, 13:15-17. Furthermore, the '865 patent functionally describes that a “graphical user interface [‘GUI’] is operable for rendering basic commands... and for customizing the operation and appearance of the requested application based on the custom configuration.” ’865*Pat*, 2:60-63. The '865 patent further explains that the graphical user interface “render[s] content and enable[s] a user to interact” with an application, for example, by allowing a user to click on a “submit” button. *Id.*, 8:2-25. With respect to Figure 6 of the '865 patent, the '865 patent paraphrases the claim language stating that “the wireless device uses a

graphical user interface to generate renderable content based on the received compiled content and the custom configuration.” *Id.*, 18:63-65. The patent uses “page description” interchangeably with “compiled content” and states “the requested application content is received as compiled content that includes a series of basic commands representing a page description.” *Id.*, 13:15-18.

In the *Hariki-Harris* system, *Hariki* discloses engines in a resource API that “process the applicable resource using the appropriate playback format depending upon the type of data or program elements in the resource.” *Hariki*, ¶43. With reference to Figure 5, below, *Hariki* discloses a general block diagram of the mobile device. *Hariki*, ¶41.

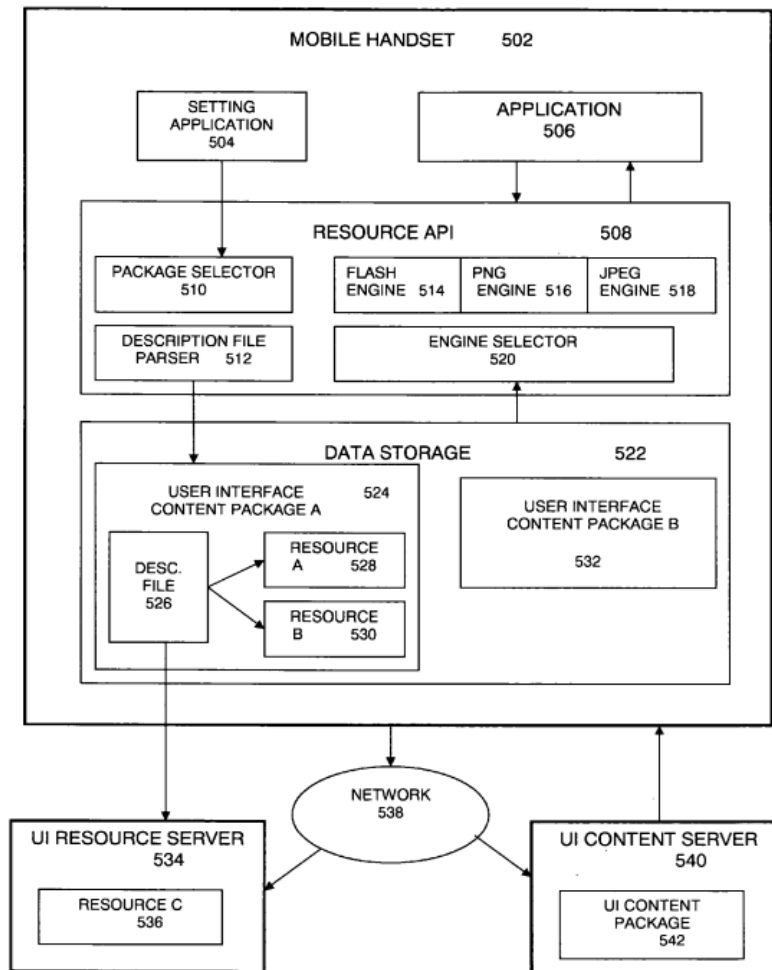


FIGURE 5

Hariki, FIG. 5.

Hariki explains that the mobile device includes application 506 that “may be a software program or utility that alters the appearance or functionality of the mobile device, or it may be a program that, when executed, provides a service to the user,” such as the web browser discussed above. *Id.*; *Bederson*, ¶171. As explained above, the application, e.g., a web browser, that receives both HTML files (i.e., “page description”) and converted resources in the UI package (i.e.,

“custom configuration”), interacts with a “resource API 508, which contains a number of functional components such as package selector 510, description file parser 512, an engine selector 520, and one or more engines.” *Hariki*, ¶43. *Hariki* explains that the engines “process the applicable resource using the appropriate playback format,” such as the Flash engine for an applet or JPEG engine for a photograph. *Id.* The engine “converts each resource file to a format or embodiment that is compatible with the application 506,” which is “in terms of parameters such as image size, color, position, and so on.” *Id.*, ¶47. POSITAs would have thus understood that *Hariki*’s resource API and engines use the converted resources in the UI package and HTML file resources and format them into renderable content for the application for display as “the ‘look and feel’ or application program environment of a device” (*id.*, ¶24). *Bederson*, ¶173. Additionally, POSITAs would have understood that client device rendering could have used JavaScript™ for displaying content received from the server. *Bederson*, ¶174 (citing *Linburn*, ¶50).

Accordingly, *Hariki* discloses a server that sends HTML files that include low level commands to a mobile device (i.e., “sending said page description to the wireless device”) and the mobile device that uses the received converted resources in the UI package and HTML file resources and formats them into renderable content for the application for display (i.e., “such that the wireless device is

capable of presenting at least one content item from said plurality of content items using both said page description and said custom configuration”).

2. Claim 2

Hariki in combination with *Harris* discloses this feature. *Bederson*, ¶¶176-179. In the *Hariki-Harris* system, *Harris* discloses WAX stylesheets to be generic templates that include content as explained in claim [1b]. *See* Section VIII.A.1.c. Additionally, *Harris* discloses “[t]he WAX stylesheets component 42 **dictates the presentation of the WAX content.**” *Harris*, ¶22. *Harris* further explains that presentation of WAX content includes content alignment commands, such as “<wax:p align= “center”>” for center alignment. *Harris*, ¶29.

```
<?xml version=" 1.0" encoding="utf-8"?>
<wax:wax xmlns:wax="http://www.kargo.com/wax" version="0.9">
  <wax:doc version="1.0">
    <wax:title>NY Nightlife</wax:title>
    <wax:block id="splash">
      <!--EXAMPLE 1 -->
      <wax : button href="index?a=main"
        keytype="accept" type="go" labelid="enter" />
      <wax:p align="center">
      <!--EXAMPLE 2 -->
      <wax: img srcid="logo" alt="My Nightlife"
        border="0" />
```

Harris’ system “ensures the best type and length of text is used, the **best type and size of image is used**, and that the **content is well suited and customized for the device attributes.**” *Harris*, ¶7. Further, *Harris* discloses the “WAX language may be dynamically translated to a requesting device's native language... For example, a WAX ‘Date-Chooser’ renders one way on the limited

screen sizes of mobile phones (possibly through a series of WML decks and another way on a PDA (all on one Screen).” *Harris*, ¶18.

POSITAs would have understood that in order to optimally present content to various types of mobile devices and PDAs, *Harris*’ generic WAX stylesheet includes specific location and size information for its content that are implemented through alignment commands such as “center,” which is a physical location relative to the edges of the containing window. *Bederson*, ¶178.

Accordingly, the *Hariki-Harris* system discloses the WAX stylesheet including content location and size information (i.e., “wireless device generic template includes relative screen location and display object size information”) and that the WAX stylesheet and content can be optimally displayed across various types of devices (i.e., “independent of screen dimensions of the wireless device”). *Bederson*, ¶179.

3. Claim 3

Hariki in combination with *Harris* discloses this feature. *Bederson*, ¶180. In the *Hariki-Harris* system, *Harris* discloses WAX stylesheets to be generic templates that include content as explained in claim [1b]. *See* Section VIII.A.1.c. Additionally, *Harris* discloses the WAX is “Wireless Abstract XML.” *Harris*, ¶7.

Accordingly the *Hariki-Harris* system discloses XML compliant WAX stylesheets (i.e., “wireless device generic template ... (XML) syntax”). *Bederson*, ¶180.

4. Claim 4

Hariki in combination with *Harris* discloses this feature. *Bederson*, ¶¶181-185. In the *Hariki-Harris* system, as explained in claim [1d], *Harris* discloses generating HTML files to deliver content for rendering on mobile devices. *See* Section VIII.A.1.e.

Additionally, *Hariki* explains that the UI skin may consist “of a particular screen configuration, size, color scheme, font, menu scheme, keypad button assignment, ringtone selection.” *Hariki*, ¶26. *Hariki*, explains that “[m]any different aspects of a mobile device may be customizable however, such as the format and display of menus, commands, subwindows, and so on,” and that the server can provide the customized UI. *Id.* *Hariki* explains that the UI skin allows configuration of “backgrounds, title bars, **buttons**, **alert sounds**, and so on.” *Id.*, ¶24. POSITAs would have understood that these backgrounds, title bars, buttons, and alert sounds comprise “rendering blocks” within the context of the ’865 patent. *Bederson*, ¶183. POSITAs would have understood that a HTML file can reference multiple rendering blocks, which use the received HTML files to render content at a mobile device. *Bederson*, ¶183. Furthermore, POSITAs would have understood

that to render a command is to execute that command to cause a UI to display content because the tags and commands in HTML files are interpreted by a web browser to display content. *Id.*, ¶184.

Accordingly, the *Hariki-Harris* system discloses a server sending HTML files (i.e., “said at least one discrete low level rendering command of said page description”) to a mobile device, where rendering blocks use the HTML files to render content (i.e., “is operable ...of the wireless device”).

5. Claim 5

Hariki in combination with *Harris* discloses this feature. *Bederson*, ¶¶186-189. The *Hariki-Harris* system discloses low level rendering commands in HTML files as explained in claim [1e]. See Section VIII.A.1.f. Furthermore, *Harris* discloses the MCF “generate[s] images that are **displayed correctly** on each device. The image-scaling framework scales and crops images to the right size and translates between image formats.” *Harris*, ¶22. Moreover, *Harris* discloses aligning content, such as a button, to the “center” of the screen. See, e.g., *id.*, ¶29 (showing that the “wax : button” is center aligned (i.e., “<wax:p align=‘center’”)); *Bederson*, ¶187. POSITAs would have understood that the center of the screen is a “physical screen position,” as claimed. *Id.*, ¶188. POSITAs would have also understood that to correctly display images on a device, for example, in the center of the screen, *Harris*’ HTML file would include comparable alignment commands

specifying content position and location on the device. *Id.*, ¶188. Furthermore, it was well-known that HTML files were commonly used alongside of CSS files, and CSS files had express “absolute” positioning commands used for positioning HTML content. *Id.*, ¶188. It was also well-known that on mobile devices, HTML applications often ran in full-screen mode, meaning that the window would fill the screen. *Id.*, ¶188. Accordingly, any positioning that was specified for the window would be positioned on the screen in the same way. *Id.*, ¶188.

Accordingly, the *Hariki-Harris* system discloses a server transmitting HTML files to a client device (i.e., “at least one discrete low level rendering command”), which include commands for content position and location on the client device’s display (i.e., “includes a physical screen position ...one content item”).⁹ *Bederson*, ¶189.

⁹ To the extent that this claim is not enabled or lacks written description and/or fails to point out and distinctly claim “physical screen position,” *Hariki* in combination with *Harris* discloses this feature. Petitioner reserves all rights to raise §112 arguments in any litigation involving the ’865 patent.

6. Claim 6

Hariki in combination with *Harris* discloses this feature. *Bederson*, ¶¶190-195. The *Hariki-Harris* system discloses low level rendering commands in HTML files as explained in claim [1e]. *See* Section VIII.A.1.f.

Additionally, *Hariki* discloses “[u]ser interface descriptor information comprising the customizable UI elements are downloaded to a mobile device, such as 108 or 109. In one embodiment, the UI descriptor information can comprise **screen parameter definitions (e.g., size, aspect ratio, icon definitions, and so on).**” *Hariki*, ¶27. *Harris* discloses “a WAX ‘Date-Chooser’ renders one way on the limited screen sizes of mobile phones (possibly through a series of WML decks[]) and another way on a PDA (all on one Screen).” *Harris*, ¶18. *Harris* further discloses “some screens are large and some are small. For example, it may be best to limit the size of a restaurant review to one sentence when communicating with a mobile phone, but one paragraph may be more appropriate when communicating with a PDA with much [sic] larger screen.” *Harris*, ¶32. POSITAs would have understood that the *Hariki-Harris* system specifies location and size of screen areas in client devices to optimally display content received from a server. *Bederson*, ¶192.

Furthermore, *Hariki* explains that the UI skin allows configuration of “backgrounds, title bars, **buttons, alert sounds**, and so on.” *Id.*, ¶24. POSITAs

would have understood that these backgrounds, title bars, buttons, and alert sounds comprise “rendering blocks” within the context of the ’865 patent. *Bederson*, ¶193. POSITAs would have understood that to render content on user interfaces of mobile devices using HTML, a HTML file can reference multiple rendering blocks. *Bederson*, ¶194.

Accordingly, the *Hariki-Harris* system discloses specifying screen parameter definitions and WML decks (i.e., “a specification of location and a size of a screen area”) and configuration of title bars, buttons, and alert sounds (i.e., “an identification of a rendering block ... to operate within said screen area”) for display on a client device. *Bederson*, ¶195.

7. Claim 7

Hariki in combination with *Harris* discloses this feature. *Bederson*, ¶¶196-197. *Harris* describes examples of WAX programming code that includes attributes for elements, such as buttons. *Harris*, ¶29. Furthermore, *Harris* explains that the WAX code is translated into HTML with other certain attributes. *Id.*, ¶¶30-32; *Bederson*, ¶196. *Harris* shows that a “button” includes a “keytype” and “align” attribute. *Harris*, ¶29; *Bederson*, ¶196. Furthermore, POSITAs would have understood that a button comprises “rendering block” within the context of the ’865 patent. *Bederson*, ¶197. POSITAs would have recognized that the button has

associated parameters such as labels and positions that allow them to be rendered on a mobile device. *Bederson*, ¶197.

8. Claim 8

Hariki in combination with *Harris* discloses this feature. *Bederson*, ¶¶198-200. As explained in claim [1d], the *Hariki-Harris* system discloses generating HTML files (i.e., “generating a page description”) from WAX content from WAX and device stylesheets. *See* VIII.A.1.e. *Harris* further discloses “in the translation process, elements (or tags) in the WAX file are translated to be rendered in different ways, specific to the mobile device. *Bederson*, ¶198. For example, “[t]he <wax:button> element is displayed as a ‘soft-key’ for WAP devices, but as a ‘link’ for devices which understand only HTML.” *Harris*, ¶30. Accordingly, *Harris* discloses a display rendering command. *Bederson*, ¶198.

Furthermore, *Hariki*’s UI content package “allows a user to customize the ‘look and feel’ or application program environment of a device by altering display and/or **sound output aspects** of the device, such as backgrounds, title bars, buttons, alert sounds, and so on.” *Hariki*, ¶24. The resource files in the UI content package “can contain various types of data objects relating to the user interface elements of the mobile devices, such as ...**sound files**, ... movies.” *Id.*, 31. Accordingly, *Hariki* discloses renderable content comprises audio content.

Bederson, ¶199. POSITAs would have understood that to render audio content, an audio rendering command is required. *Bederson*, ¶199.

Therefore, the *Hariki-Harris* system discloses display and audio rendering commands. *Bederson*, ¶200.

9. Claim 9

Hariki in combination with *Harris* discloses this feature. *Bederson*, ¶¶201-203. As explained above, the *Hariki-Harris* system discloses a web server process that sends HTML files to mobile devices. *See* Sections VIII.A.1.a, VIII.A.1.b, and VIII.A.1.g. As also explained above, the mobile devices may be computers or mobile devices from different manufacturers. *See* Section VIII.A.1.a. Accordingly, *Hariki* discloses the web server process (i.e., “application program”) is mobile device generic. *Bederson*, ¶202.

Furthermore, *Harris* explains that WAX is “a generic markup language, referred to as Wireless Abstract XML (hereafter WAX) that is easily extended and can be translated into a variety of different mobile device markup languages.” *Harris*, ¶7. *Harris* further discloses “WAX is designed to enable the content developer to describe content at a more abstract level than that used in individual protocols. The greater level of abstraction enables the subsequent conversion of WAX into the languages used by the requesting mobile devices such as WML, HDML and HTML.” *Id.* Thus, POSITAs would have understood that the WAX

code (i.e., “wireless device generic template”) describes a page to be displayed in an abstract way that is not dependent on the type of device requesting the page of information (i.e., “wireless device generic page description template”). *Bederson*, ¶203.

10. Claim 10

Hariki in combination with *Harris* discloses this feature. *Bederson*, ¶¶204-210. *Hariki* explains that the UI skin may consist “of a particular screen configuration, size, color scheme, font, menu scheme, keypad button assignment, ringtone selection.” *Hariki*, ¶26. *Hariki*, explains that “[m]any different aspects of a mobile device may be customizable however, such as the format and display of menus, commands, subwindows, and so on,” and that the server can provide the customized UI. *Id.* *Hariki* explains that the UI skin allows configuration of “backgrounds, title bars, **buttons**, **alert sounds**, and so on.” *Id.*, ¶24.

POSITAs would have understood that these backgrounds, title bars, buttons, and alert sounds comprise “rendering blocks” within the context of the ’865 patent. *Bederson*, ¶205. Indeed, the ’865 patent explains that rendering blocks include but are not limited to:

an edit box for entering text, static text for displaying text, an image, a pop-up menu which may appear in response to a user interaction, a drop-down menu list, tabbed menu for displaying several pages where each tab may display a text and an optional icon, **sound** for controlling audio (e.g., pause, rewind, stop, play

and the like), video to display a video with visual control panel (e.g., pause, rewind, stop, play and the like), ticker to display horizontal scrolling text, **check box/radio button** to enable selection/de-selection of items, rating control for rating content (e.g., movies), poll control for displaying the current poll result, canvas for drawing objects, a tree for displaying hierarchical data, scroll bar for scrolling up/down and/or left/right, a progress bar to display download progress, a table for displaying data in a tabular form, a calendar for displaying and enabling selection/de-selection of a date and the like.

'865Pat, 8:27-43. POSITAs would have understood that *Hariki*'s configurable buttons and sounds, fall squarely within the '865 patent's examples of "rendering blocks." *Bederson*, ¶207.

Furthermore, in the *Hariki-Harris* system, *Hariki* discloses engines in a resource API that "process the applicable resource using the appropriate playback format depending upon the type of data or program elements in the resource." *Hariki*, ¶43. With reference to Figure 5, below, *Hariki* discloses a general block diagram of the mobile device. *Hariki*, ¶41.

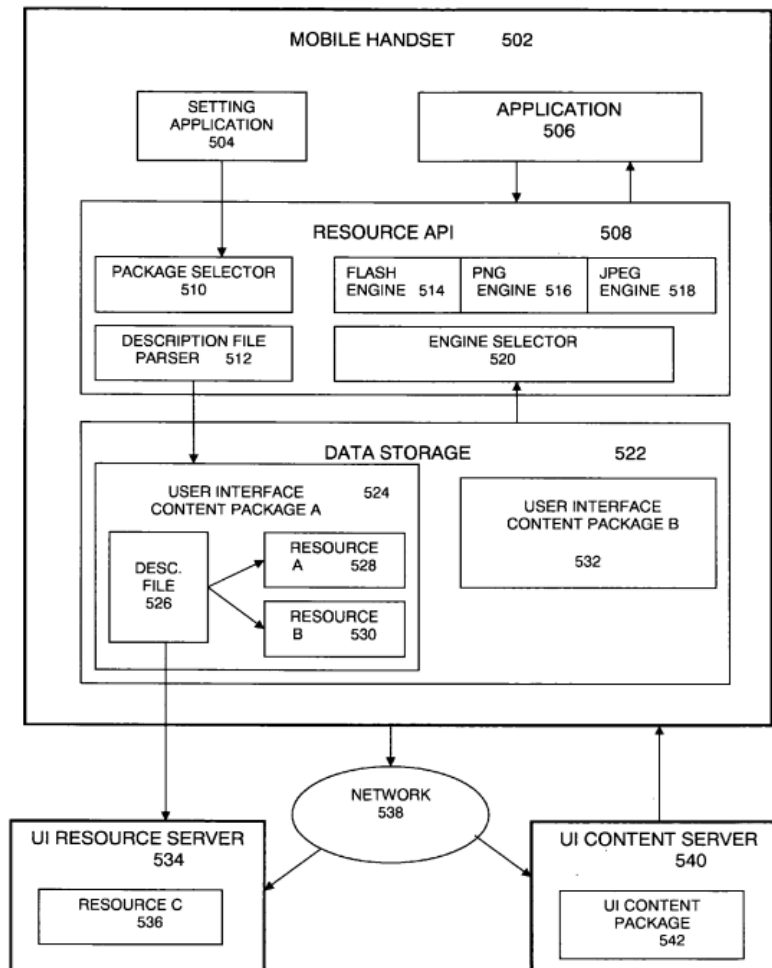


FIGURE 5

Hariki, FIG. 5.

Hariki explains that the mobile device includes application 506 that “may be a software program or utility that alters the appearance or functionality of the mobile device, or it may be a program that, when executed, provides a service to the user,” such as the web browser discussed above. *Id.*; *Bederson*, ¶209. The application, e.g., a web browser that, receives converted resources (i.e., “custom

configuration”) in the UI content package,¹⁰ interacts with a “resource API 508, which contains a number of functional components such as package selector 510, description file parser 512, an engine selector 520, and one or more engines.” *Hariki*, ¶43. *Hariki* explains that the engines “process the applicable resource using the appropriate playback format,” such as the Flash engine for an applet or JPEG engine for a photograph. *Id.* The engine “converts each resource file to a format or embodiment that is compatible with the application 506,” which is “in terms of parameters such as image size, color, position, and so on.” *Id.*, ¶47. POSITAs would have thus understood that *Hariki*’s resource API and engines use the converted resources in the UI content package and format them into renderable content for the application for display as “the ‘look and feel’ or application program environment of a device.” *Id.*, ¶24. *Bederson*, ¶210. POSITAs would have further understood that the “look and feel” environment of the device was a user

¹⁰ *Hariki* uses several terms interchangeably for UI content package. *Hariki* describes it as “UI content” ¶19, “UI skin content” (*Hariki*, ¶27), “UI skin package data” (*id.*, ¶27), and “UI package” (*id.*, ¶40). *Bederson*, ¶209. For the purposes of this petition, Petitioner will uniformly refer to the downloadable UI content as “UI content package.”

interface displayed to the user and corresponded to the “graphical user interface,” as claimed. *Bederson*, ¶210.

Accordingly, the *Hariki-Harris* system discloses using converted resources (i.e., “custom configuration”) to configure a UI content package’s backgrounds, title bars, buttons, alert sounds (i.e., “configure a plurality of rendering blocks”) to render content on a user interface (i.e., “of a graphical user interface”). *Bederson*, ¶210.

11. Claim 11

Hariki in combination with *Harris* discloses this feature. *Bederson*, ¶¶211-213. In the *Hariki-Harris* system, *Harris* explains that “[a]ll mobile devices and their capabilities are kept in a device and capability registry 24 [of a server’s MCF]. The device and capability registry 24 is an XML file, and allows the content developer to group devices together, while also keeping track of the Specific features of each device.” *Harris*, ¶21. Furthermore, *Harris* discloses “a **web server accepts a connection and an HTTP request from a mobile device...** The **headers of the HTTP request are examined by the MCF 8 in order to determine the type of device** and a suitable XSL stylesheet to use for later device-specific translation (step 62). An **XML registry of devices is consulted to determine device capabilities** (Screen size, browser version, etc.) (step 64).”

Accordingly, the *Hariki-Harris* system discloses a server's MCF storing mobile devices and their capabilities in a registry (i.e., "stores a plurality of device profiles"), identifying a content-requesting mobile device's capabilities from the registry (i.e., "identifying a device profile from said plurality of device profiles associated with the wireless device") based on an identified type of the mobile device (i.e., "based on said type of said wireless device"). *Bederson*, ¶213.

12. Claim 31¹¹

Hariki in combination with *Harris* discloses this feature. *Bederson*, ¶¶214-215. *Hariki* discloses its "mobile device UI configuration system [] allows a UI skin package content provider to easily generate **a new UI resource file** for new or different mobile devices." *Hariki*, ¶8. Furthermore, *Hariki* discloses "[t]ypically customized **UI content is downloaded upon demand**, such as **when the user requests a UI upgrade or modification from a particular content provider.**" *Hariki*, ¶27. POSITAs would have understood that an upgrade from the content provider is needed when the updated UI content is not available at the mobile device. *Bederson*, ¶214.

Accordingly, the *Hariki-Harris* system discloses a content provider sending customized UI content to a mobile device (i.e., "sending the custom configuration

¹¹ Claim 31 depends from claim 1.

to the wireless device”) upon the mobile device requesting an upgrade or modification to the content (i.e., “is in response to receiving an indication from the wireless device that the custom configuration is not stored at the wireless device”).

Bederson, ¶215.

13. Claim 12

a. [12pre]

To the extent the preamble is limiting, *Hariki* discloses this feature. *Bederson*, ¶¶216-218. *Hariki* discloses, with reference to Figure 1 below, “server computer 102... provides content data, application programs, diagnostic tools, program components, or any other content or executable objects to the mobile device 108.... For example, server 102 can execute[] a web server process to serve web pages,” such as HTML files over a network to the mobile devices. *Hariki*, ¶21. Furthermore, the server “can generate a customized user interface for a plurality of different makes and types of mobile devices.” *Id.*, ¶24. *Hariki* explains that “the customized user interface is referred to as a ‘UI skin’” to “allow a user to customize the ‘look and feel’ or application program environment of a device by altering display and/or sound output aspects of the device.” *Id.*

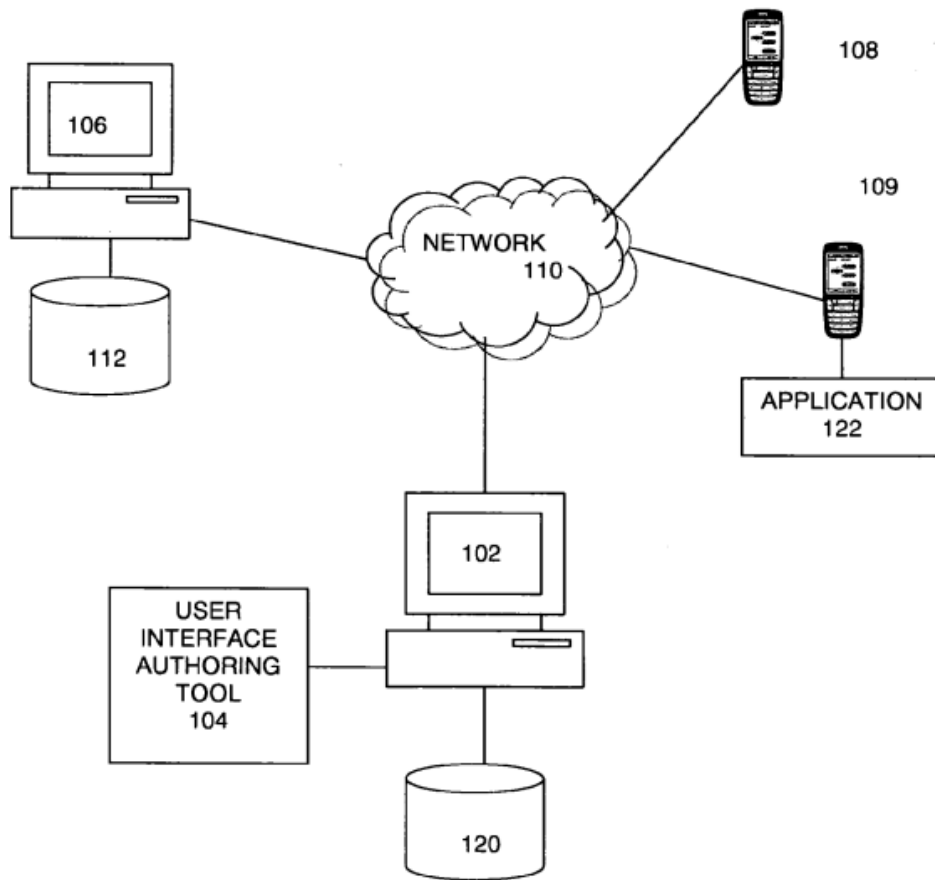


FIGURE 1

Hariki, FIG. 1.

Hariki further discloses “while only one aspect of the system is recited as embodied in machine-readable medium, other aspects may likewise be embodied in **machine-readable medium**.” *Hariki*, ¶59. Therefore, *Hariki* discloses a processor-readable medium. *Bederson*, ¶217.

Accordingly, *Hariki* discloses generating UI skins and providing content data by a server through a machine-readable medium (i.e., “non-transitory

processor-readable medium storing code representing instructions to cause a processor”) to a mobile device. *Bederson*, ¶218.

b. [12a]

Each of these elements is substantively identical to that in claim [1a]. Accordingly, *Hariki* discloses these features as discussed in Sections VIII.A.1.b. *Bederson*, ¶219.

c. [12b]

Each of these elements is substantively identical to that in claim [1b]. Accordingly, *Hariki* discloses these features as discussed in Sections VIII.A.1.c. *Bederson*, ¶220.

d. [12c]

Hariki in combination with *Harris* discloses this feature. *Bederson*, ¶¶221-222. In the *Hariki-Harris* system, *Harris* explains that “[a]ll mobile devices and their capabilities are kept in a **device and capability registry 24** [of a server’s MCF]. The device and capability registry 24 is an XML file, and allows the content developer to group devices together, while also keeping track of the specific features of each device.” *Harris*, ¶21. Furthermore, *Harris* discloses “a **web server accepts a connection and an HTTP request from a mobile device....** The **headers of the HTTP request are examined by the MCF 8 in order to determine the type of device** and a suitable XSL stylesheet to use for later device-

specific translation (step 62). An **XML registry of devices is consulted to determine device capabilities** (screen size, browser version, etc.) (step 64).” *Harris*, ¶24.

Accordingly, the *Hariki-Harris* system discloses storing mobile devices and their capabilities in a registry and identifying a content-requesting mobile device’s capabilities from the registry (i.e., the claimed “request a device profile from a library of device profiles”) based on an identified type of the mobile device (i.e., “using said indication of said type of the wireless device”). *Bederson*, ¶222.

e. [12d]

Hariki in combination with *Harris* discloses this feature. *Bederson*, ¶223. In the *Hariki-Harris* system, *Harris* explains that “[a]n **XML registry of devices is consulted to determine device capabilities** (screen size, browser version, etc.) (step 64).” *Harris*, ¶24.

Accordingly, the *Hariki-Harris* system discloses the stored mobile device information including capability information (i.e., “device profile providing an indication of a rendering capability of the wireless device”). *Bederson*, ¶223.

f. [12e]

Hariki in combination with *Harris* discloses this feature. *Bederson*, ¶¶224-225. *Harris* discloses that the MCF “[t]ags can be added and removed from the **WAX language**.... Alternately, a subscription plan can ensure that an enterprise

always obtains the most recent version of the device file. Database classes may be added which allow many different options for pooling database connections in order to perform fast database access.” *Harris*, ¶28. *Harris* further discloses in the context of translating WAX content to HTML that a “‘welcome’ may be **translated to ‘Bonjour’ for users who specify French as their preferred language** or possibly ‘Welcome to our site, please take a look around’ for **wireless devices, having a larger display.**” *Id.*, ¶30. POSITAs would have thus understood that content on WAX stylesheets can be removed, not converted to a language, or reduced in length to eliminate portions of the content (i.e., “eliminate, ... one content item”) based on a specific mobile device’s capability such as small display size stored in registry of devices. *Bederson*, ¶224.

Accordingly, the *Hariki-Harris* system discloses not converting content to a language, reducing content in length, removing tags (i.e. “eliminate... at least one content item from said plurality of content items from said wireless device generic template”) based on device capability such as small screen size (i.e. “using said device profile”) to create a WAX stylesheet and content specifically for the mobile device (i.e. “to define a device specific template associated with the wireless device”). *Bederson*, ¶225.

g. [12f]

Hariki in combination with *Harris* discloses this feature. *Bederson*, ¶¶226-227. As explained above, *Harris* discloses removing, not converting, or reducing in length content items in WAX stylesheets based on a specific mobile device's capability such as small display size stored in registry of devices. *Bederson*, ¶226. POSITAs would have understood that content on WAX stylesheets can be removed (i.e., eliminate all other terms except "welcome"), when a small screen size is not capable of displaying the entire text (i.e., "please take a look around"). *Bederson*, ¶226.

Accordingly, the *Hariki-Harris* system discloses eliminating extra text that is unsupported by a mobile device's small screen size (i.e., "said ...by the wireless device"). *Bederson*, ¶227.

h. [12g]

Each of these elements is substantively identical to that in claim [1d] and [1e]. Accordingly, *Hariki* discloses these features as discussed in Sections VIII.A.1.e and VIII.A.1.f. *Bederson*, ¶228.

i. [12h]

Each of these elements is substantively identical to that in claim [1f] and [1e]. Accordingly, *Hariki* discloses these features as discussed in Sections VIII.A.1.g and VIII.A.1.f. *Bederson*, ¶229.

14. Claim 13

Each of these elements is substantively identical to that in claim [2]. Accordingly, *Hariki* discloses these features as discussed in Sections VIII.A.2. *Bederson*, ¶230.

15. Claim 14

Each of these elements is substantively identical to that in claim [3]. Accordingly, *Hariki* discloses these features as discussed in Sections VIII.A.3. *Bederson*, ¶231.

16. Claim 15

Each of these elements is substantively identical to that in claim [4]. Accordingly, *Hariki* discloses these features as discussed in Sections VIII.A.4. *Bederson*, ¶232.

17. Claim 16

Each of these elements is substantively identical to that in claim [5]. Accordingly, *Hariki* discloses these features as discussed in Sections VIII.A.5. *Bederson*, ¶233.

18. Claim 17

Each of these elements is substantively identical to that in claim [6]. Accordingly, *Hariki* discloses these features as discussed in Sections VIII.A.6. *Bederson*, ¶234.

19. Claim 18

Each of these elements is substantively identical to that in claim [7]. Accordingly, *Hariki* discloses these features as discussed in Sections VIII.A.7. *Bederson*, ¶235.

20. Claim 19

Each of these elements is substantively identical to that in claim [8]. Accordingly, *Hariki* discloses these features as discussed in Sections VIII.A.8. *Bederson*, ¶236.

21. Claim 20

a. [20a]

Each of these elements is substantively identical to that in claim [1c]. Accordingly, *Hariki* discloses these features as discussed in Sections VIII.A.1.d. *Bederson*, ¶237.

b. [20b]

Each of these elements is substantively identical to that in claim [10]. Accordingly, *Hariki* discloses these features as discussed in Sections VIII.A.10. *Bederson*, ¶238.

22. Claim 34

a. [34a]

The elements of this claim are substantively identical to those in claim [1c]. Accordingly, *Hariki* in combination with *Harris* discloses these features as discussed in Section VIII.A.1.d. *Bederson*, ¶239.

b. [34b]

The elements of this claim are substantively identical to those in claim [1f]. Accordingly, *Hariki* in combination with *Harris* discloses these features as discussed in Section VIII.A.1.g. *Bederson*, ¶240.

B. *Ground 2: Hariki in view of Harris and Dees renders obvious Claims 1-20, 31, 34.*

1. Claim 1

Each of these elements is disclosed by the combination of *Hariki* and *Harris* as explained in Sections VIII.A.1.a – VIII.A.1.g. *Bederson*, ¶241.

2. Claim 2

Each of these elements is disclosed by the combination of *Hariki* and *Harris* as explained in Section VIII.A.2. *Bederson*, ¶242.

3. Claim 3

Each of these elements is disclosed by the combination of *Hariki* and *Harris* as explained in Section VIII.A.3. *Bederson*, ¶243.

4. Claim 4

Each of these elements is disclosed by the combination of *Hariki* and *Harris* as explained in Section VIII.A.4. *Bederson*, ¶244.

5. Claim 5

Hariki in combination with *Harris* discloses this feature for the same reason as explained above in Section VIII.A.5. *Bederson*, ¶¶245-253. In the *Hariki-Harris* system, *Harris* discloses the MCF “generate[s] images that are **displayed correctly** on each device. The image-scaling framework scales and crops images to the right size and translates between image formats.” *Harris*, ¶22.

Moreover, *Harris* discloses aligning content, such as a button, to the “center” of the screen. *See, e.g., id.*, ¶29 (showing that the “wax : button” is center aligned (i.e., “<wax:p align=‘center’”)); *Bederson*, ¶246. POSITAs would have understood that the center of the screen is a “physical screen position,” as claimed. *Id.*, ¶246. POSITAs would have also understood that to correctly display images on a device, for example, in the center of the screen, *Harris*’ HTML file would include comparable alignment commands specifying content position and location on the device. *Id.*, ¶246. Additionally, it was well-known that HTML files were commonly used alongside of CSS files, and CSS files had express “absolute” positioning commands used for positioning HTML content. It was also well-known that on mobile devices, HTML applications often ran in full-screen mode, meaning

that the window would fill the screen. Accordingly, any positioning that was set for the window would be positioned for the screen in the same way. *Id.*, ¶246.

Dees, like the *Hariki-Harris* system of a transmitting HTML files and WAX stylesheets from server to client, discloses a client device that receives HTML, CSS files, and stylesheet documents from a server and renders a user interface based on the received documents. *Dees*, ¶51. Furthermore, *Dees* explains that “style sheet document [] can comprise style information in any language suitable for marking up style information, such as CSS or DSSSL. It is also possible to use XML to mark up style information.” *Id.*, ¶53. Additionally, *Dees* discloses “HyperText Markup Language HTML also provides [] ways to provide a user interface definition.” *Id.*, ¶37. *Dees* further states that “[u]sing a language such as CSS or DSSSL, style properties such as the font or the color of a button or the **position of a checkbox on the screen** can easily be defined.” *Id.*, ¶3.

POSITAs would have understood that the system disclosed by *Dees* would be compatible with and could be adapted and applied to the *Hariki-Harris* system because it has the same general architecture as *Hariki-Harris*’ client-server system because both client-server systems are directed towards rendering user interfaces on client devices based on commands transmitted from the server. *Bederson*, ¶248. Thus, to the extent that PO contends that the *Hariki-Harris* system does not disclose “discrete low level rendering command includes a physical screen position

of a display element of said at least one content item,” POSITAs would have been motivated to combine the *Hariki-Harris* client-server system with the teachings of *Dees* directed to a server transmitting HTML and CSS commands that define positions of content display elements on the client device’s screen. *Bederson*, ¶248. In particular, to the extent PO contends that, while the *Hariki-Harris* system discloses proper display of content on a mobile device, it does not expressly disclose the converted HTML file, POSITAs would have been motivated to find a method to specify screen position of a content display element on a client device and looked to *Dees* for that method. *Bederson*, ¶249.

Moreover, POSITAs would have had a reasonable expectation of success in implementing *Dees*’ system of receiving HTML, CSS files, and stylesheet documents from a server and rendering a user interface based on the received documents with *Hariki-Harris*’ content display system. *Bederson*, ¶250. If PO should contend that while the *Hariki-Harris* system discusses aligning content, such as a button, to the “center” of the screen (*Harris*, ¶29), it does not show that the HTML file commands specifying the physical “center” location on the screen (*Bederson*, ¶251), *Dees* does disclose such commands. *Id.*, ¶76.

Accordingly, to the extent that PO argues the *Hariki-Harris* system does not disclose “discrete low level rendering command includes a physical screen position

of a display element of said at least one content item,”¹² *Dees* fills such a gap in the *Hariki-Harris* system by explaining it “provide[s] style information” such as “the X- and Y- coordinates on the screen.” *Dees*, ¶38. It further explains that “[u]sing a language such as CSS or DSSSL, style properties such as the font or the color of a button or the position of a checkbox on the screen can easily be defined.” *Dees*, ¶3. Furthermore, *Dees* discloses “generic style definitions such as the font to be used can be provided in a generic style sheet document, and platform-specific style definitions such as the **positioning of the user interface elements in a window on the screen** can be provided in specific style sheet documents.” *Dees*, ¶4. Moreover, it was well-known that CSS includes absolute positioning commands that specify the position of an item. *Bederson*, ¶252. And furthermore, on mobile devices, windows were often full-screen so that the position of an item that was specified for a window was also specified for the screen. *Id.*, ¶252. Therefore, POSITAs would have been motivated to combine *Dees* and the *Hariki-Harris* system because, while each discloses proper display of content on a device, *Dees*’ disclosure of using translated commands included in HTML files to specify content location on a device further specifies a way for the *Hariki-Harris* system to correctly display content. *Id.*, ¶252.

¹² Petitioner does not agree with such a contention.

Accordingly, the *Hariki-Harris-Dees* system discloses a server transmitting HTML and CSS files (i.e., “at least one discrete low level rendering command”) that identify user interface elements (i.e., “of a display element of said at least one content item”) on a client device’s screen (i.e., “includes a physical screen position”). *Bederson*, ¶253.

6. Claim 6

Hariki in combination with *Harris* and *Dees* discloses this feature. *Bederson*, ¶¶254-257. As explained in claim [1e], *Harris* discloses a discrete low level rendering command. *See* Section VIII.A.1.f. And as explained in claim [5], *Dees* discloses a physical screen position of a display element. *See* Section VIII.B.5. Furthermore, as shown below, *Dees* illustrates a device screen center location in an embodiment of the device-specific style sheet. *Dees*, ¶62.

```
(1) <?xml version="1.0"?>
(2) <uisl level="2">
(3)   <4x3>
(4)     <halign>
(5)       <all class="settings">center</all>
(6)       <all class="effects">left</all>
(7)     </halign>
(8)     <level3 stylesheet="stylesheet-landscape-4x3.uisl"/>
(9)   </4x3>
(10) </uisl>
```

Dees further explains “style sheet document 212 **provides more specific style information for a graphical display** with a landscape orientation, by providing style information for a graphical display with a 4:3 aspect ratio.” *Dees*, ¶65. *Dees* explains that “[t]he user interface module 420 parses the second style

sheet document 212 and determines that more **specific style information is available for displays with resolutions in a ratio of 4:3.**” *Id.* Furthermore, “[s]ince the display 452 has a 640x480 resolution, which corresponds to a ratio of 4:3, the user interface module 420 collects this information.” *Id.* *Dees* further explains “style sheet document provides specific style sheet information **for graphical devices with a 640x480 or with a 1024x768 pixel resolution display.** This style sheet document could also be made valid by defining an appropriate DTD.” *Id.*, ¶68. Accordingly, POSITAs would have understood that the *Hariki-Harris-Dees* system provides HTML files that specify location such as center, and size such as 640x480 pixel resolution display of a mobile device’s display. *Bederson*, ¶255.

Dees further discloses “style properties such as the font or the color of a button or the **position of a checkbox on the screen** can easily be defined” (*Dees*, ¶3), and “platform-specific style definitions such as the **positioning of the user interface elements in a window on the screen** can be provided in specific style sheet documents” (*id.*, ¶4). Accordingly, POSITAs would have understood that the *Hariki-Harris-Dees* system identifies interface elements to be positioned in specific locations on a user’s display window. *Bederson*, ¶256.

Accordingly, the *Hariki-Harris-Dees* system provides HTML files that specify location such as center, and size such as 640x480 pixel resolution display

of a mobile device's display (i.e., "wherein said at least one discrete low level ... a size of a screen area") and identifies interface elements to be positioned in specific locations on a user's display window (i.e., "identification ...of said graphical user interface to operate within said Screen area"). *Bederson*, ¶257.

7. Claim 7

Hariki in combination with *Harris* and *Dees* discloses this feature. *Bederson*, ¶¶258-260. *Harris* describes examples of WAX programming code that includes attributes for elements, such as buttons. *Harris*, ¶29. Furthermore, *Harris* explains that the WAX code is translated into HTML with other certain attributes. *Id.*, ¶¶30-32; *Bederson*, ¶259. *Harris* shows that a "button" includes a "keytype" and "align" attribute. *Id.*, ¶259. Furthermore, POSITAs would have understood that a button comprises "rendering block" within the context of the '865 patent. *Bederson*, ¶259. POSITAs would have recognized that the buttons have associated parameters such as labels and positions that allow them to be rendered on a mobile device. *Bederson*, ¶259. Additionally, *Dees* also lists (shown below) an "interpretation of the elements and attributes used in the...style sheet document." *Dees*, ¶75.

Tag	Attributes	Meaning
<general> </general>	none	Contains general screen based style attributes, such as background colors.
<bgcolor> </bgcolor>	none	Defines the background color of the user interface elements. This is done by mapping all or certain UI elements to a color value. Color values are defined using the following syntax: '#' + 2-char hex value for red (between 00 to ff) + 2-char hex value for green + 2-char hex value for blue, E.g.: #ff00ff for purple color.
 	none	Specifies a hint about which kind of font should be used. The possible values are: - serif-proportional - sansserif-proportional - serif-nonproportional - sansserif-nonproportional Serif fonts are e.g. "Times" and "Garamond" Sans-serif fonts are e.g. "Arial" and "Helvetica". All these fonts are proportional. An example of a non-proportional font is "Courier".
<portrait> </portrait>	none	Provides style hints for portrait screens (i.e. when the top is narrower than the sides in the position normally used by people). The style attributes that can be defined here are the layout and the orientation of the elements. Others are defined in level 2 of the UI stylesheet language (referred to by a <level2>-tag inside this <portrait>-tag).
<landscape> </landscape>	none	Provides style hints for landscape screens (i.e. when the top is broader than the sides in the position normally used by people). The style attributes, which can be defined here, are the layout and the orientation of the elements. Others are defined in level 2 of the UI stylesheet language (referred to by a <level2>-tag inside this <landscape>-tag).
<layout> </layout>	none	Specifies how the elements should be ordered inside the containers. All layout "algorithms" which can be chosen here, start in the upper-left corner of the screen and place the elements one by one in vertical direction (default) or horizontal if the orientation of this container is specified as horizontal (see <orientation>-tag). The "algorithms" which can be chosen are: - panelsplit: place the elements one after another including complete subcontainers with their elements and if the next element does not fit anymore, place it on a new panel, in this way splitting the user interface over several panels. It is required for a client to provide some means of navigation between the different panels if this layout algorithm is used. - panelgrid: place the elements one by one and continue in another column (orientation: vertical) or in another row (orientation: horizontal) if the next element does not fit anymore, forming a grid. A grid means that all columns and all rows have the same number of grid cells the size of which is determined by the biggest element in the row and the column. If no more columns (or rows) fit on the screen, the client can place them on another panel (i.e. split the UI over several panels) or provide a means of scrolling. - paneltree: place the elements one by one and continue in the next column (orientation: vertical) or in the next row (orientation: horizontal), but independent from the previous column/row, i.e. does not have to form a grid, so each column can contain a different number of UI elements. - menusplit: same as panelsplit, but now the subcontainers can be hidden. Only the label of the subcontainer has to be shown and not all its elements, in this way forming a hierarchy of menus and submenus. - menutree: (the default algorithm for containers)

POSITAs would have understood that the element interpretation and attributes in *Dees* are the parameters associated with interface elements (i.e., “rendering blocks”). *Bederson*, ¶260.

8. Claim 8

Each of these elements is disclosed by the combination of *Hariki* and *Harris* as explained in Section VIII.A.8. *Bederson*, ¶261.

9. Claim 9

Each of these elements is disclosed by the combination of *Hariki* and *Harris* as explained in Section VIII.A.9. *Bederson*, ¶262.

10. Claim 10

Each of these elements is disclosed by the combination of *Hariki* and *Harris* as explained in Section VIII.A.10. *Bederson*, ¶263.

11. Claim 11

Each of these elements is disclosed by the combination of *Hariki* and *Harris* as explained in Section VIII.A.11. *Bederson*, ¶264.

12. Claim 31

Each of these elements is disclosed by the combination of *Hariki* and *Harris* as explained in Section VIII.A.12. *Bederson*, ¶265.

13. Claim 12

Each of these elements is disclosed by the combination of *Hariki* and *Harris* as explained in Section VIII.A.13. *Bederson*, ¶266.

14. Claim 13

Each of these elements is disclosed by the combination of *Hariki* and *Harris* as explained in Section VIII.A.14. *Bederson*, ¶267.

15. Claim 14

Each of these elements is disclosed by the combination of *Hariki* and *Harris* as explained in Section VIII.A.15. *Bederson*, ¶268.

16. Claim 15

Each of these elements is disclosed by the combination of *Hariki* and *Harris* as explained in Section VIII.A.16. *Bederson*, ¶269.

17. Claim 16

Each of these elements is substantively identical to that in claim [5]. Accordingly, the combination of *Hariki*, *Harris*, and *Dees* discloses these features as discussed in Sections VIII.B.5. *Bederson*, ¶270.

18. Claim 17

Each of these elements is substantively identical to that in claim [6]. Accordingly, the combination of *Hariki*, *Harris*, and *Dees* discloses these features as discussed in Sections VIII.B.6. *Bederson*, ¶271.

19. Claim 18

Each of these elements is substantively identical to that in claim [7]. Accordingly, the combination of *Hariki*, *Harris*, and *Dees* discloses these features as discussed in Sections VIII.B.7. *Bederson*, ¶272.

20. Claim 19

Each of these elements is disclosed by the combination of *Hariki* and *Harris* as explained in Section VIII.A.20. *Bederson*, ¶273.

21. Claim 20

Each of these elements is disclosed by the combination of *Hariki* and *Harris* as explained in Section VIII.A.21. *Bederson*, ¶274.

22. Claim 34

Each of these elements is disclosed by the combination of *Hariki* and *Harris* as explained in Section VIII.A.22. *Bederson*, ¶275.

IX. SECONDARY CONSIDERATIONS

Petitioner is unaware of any evidence of secondary considerations that would support a finding of non-obviousness. The asserted prior art demonstrates there is no evidence of failure by others and that the features recited in the challenged claims were readily available in the prior art. *Bederson*, ¶276.

X. DISCRETIONARY DENIAL IS NOT WARRANTED UNDER EITHER § 314 OR § 325

A. *Fintiv* Factors

The *Fintiv* framework should not be followed here because the framework exceeds the Director's authority, is arbitrary and capricious, and was adopted without notice-and-comment rulemaking. Even if considered, the Board should not exercise its discretion to deny institution under 35 U.S.C. § 314(a) based on *Apple*

Inc. v. Fintiv, Inc., IPR2020-00019, Paper 11 at 2–3, 6 (Mar. 20, 2020)), and the USPTO’s recent Interim Procedure for Discretionary Denials¹³ (“Interim *Fintiv* Guidance”) because, on balance, the *Fintiv* factors do not weigh in favor of denying institution of trial. Moreover, the Petition presents a compelling, meritorious challenge, and there is no evidence of abuse by Petitioners. The Board should not exercise its discretion to deny institution on this basis alone. *See* Interim *Fintiv* Guidance, 3-5.

Fintiv factor one is neutral. Upon institution of this petition, Petitioners may file a motion to stay the parallel litigation. Furthermore, while no motion to stay pending IPR has yet been filed in the California district court case,¹⁴ courts commonly stay cases upon IPR institution. *VMWare, Inc. v. Intell. Ventures II LLC*, IPR2020-00859, Paper 13 at 12 (PTAB Nov. 5, 2020) (finding factor one neutral, even though Petitioner had not previously sought a stay, and despite Patent

¹³ Available at https://www.uspto.gov/sites/default/files/documents/interim_proc_discretionary_denials_aia_parallel_district_court_litigation_memo_20220621_.pdf

¹⁴ *See GoTV Streaming, LLC v. Netflix Inc.*, 2:22-CV-07556-RGK-SHK (C.D. Cal.).

Owner’s argument that the district court judge was “unlikely” to issue a stay pending IPR institution).

Fintiv factor two weighs against discretionary denial. The Interim *Fintiv* Guidance recognizes that “[a] court’s scheduled trial date... is not by itself a good indicator of whether the district court trial will occur before the statutory deadline for a final written decision” and “allows parties to present evidence regarding the most recent statistics on median time-to-trial for civil actions in the applicable district court and identifies the report providing those statistics.”¹⁵ Interim *Fintiv* Guidance, 8-9 & n.12. That report shows the “most recent statistics on median time-to-trial for civil actions” in the Central District of California as 22.9 months.¹⁶ Here, the complaint in the Netflix litigation was filed on October 17, 2022, making the calculated trial date in September 2024. *See* EX1023. Although the trial date in the Netflix litigation is currently October 17, 2023, that date could get pushed

¹⁵ Available at <https://www.uscourts.gov/statistics/table/na/federal-court-managementstatistics/2022/03/31-1>.

¹⁶ *See* https://www.uscourts.gov/sites/default/files/data_tables/fcms_na_distprofile1231.2022.pdf.

back. EX1022. The Board's final written decision (FWD) in this case is expected in approximately October 2024.

Fintiv factor three weighs against discretionary denial because the court has not yet issued substantive orders relating to the '865 patent and Petitioner has been diligent. To date, expert reports have not been prepared on any issues in the Netflix litigation. Further, this Petition comes over six months before petitioner's bar date.

Fintiv factor four weighs against discretionary denial. If this Petition is instituted, Petitioner stipulates that it will not pursue the ground identified in this Petition before the district court. *Sand Revolution II, LLC v. Cont'l Intermodal Grp.-Trucking LLC*, IPR2019-01393, Paper 24 at 11-12 (June 16, 2020) (informative). Given the word limit imposed by 37 C.F.R. § 42.24(a)(1)(i), this Petition presents all grounds that Petitioner could have reasonably raised in one petition. Additionally, GoTV asserts only twenty-one claims in the district court, while this Petition challenges twenty-two claims.

Fintiv factor five is neutral because Petitioner and Patent Owner are the same parties as in the district court.

Regarding *Fintiv* factor six, the strong merits of this Petition weigh heavily against discretionary denial. Petitioner raises a compelling, meritorious challenge that the PO cannot meaningful rebut, which underscores the conclusion that one or more claims of the '865 patent are unpatentable by a preponderance of evidence.

Considering the *Fintiv* factors overall, institution would best serve the efficiency and integrity of the system.

B. *Becton Dickinson/Advanced Bionics/General Plastics Factors*

Denying institution under 35 U.S.C. § 325(d) would not be appropriate here, because the grounds and arguments relied on by this Petition were not previously presented to the Patent Office. None of the relied-upon art was considered during prosecution. This is the only IPR that has ever been filed against the challenged claims.

Because this is Petitioner's first petition against the '865 patent, the *General Plastic* factors do not weigh against institution. *See General Plastic Indus. Co. v. Canon Kabushiki Kaisha*, IPR2016-01357, Paper 19, 16 (PTAB Sept. 6, 2016) (precedential).

XI. MANDATORY NOTICES

A. *Real Parties-in-Interest*

The real parties-in-interests here are Netflix, Inc., and Netflix Streaming Services, Inc. No other parties directed, controlled, or funded this *Inter Partes* Review proceeding (IPR).

B. *Related Matters*

As of the filing date of this Petition, and to the best knowledge of Petitioner, the '865 patent is the subject of the following pending civil actions:

- *GoTV Streaming, LLC. v. Netflix Inc.*, Case 2:22-cv-07556 (C.D. Cal.)

C. *Lead and Back-Up Counsel, and Service Information Under 37 C.F.R. § 42.8(b)(3)*

Lead Counsel	Back-up Counsel
Aliza George Carrano, Reg. No. 70,637 (acarrano@willkie.com)	Indranil Mukerji, Reg. No. 46,944(imukerji@willkie.com)
Willkie Farr & Gallagher LLP 1875 K Street NW Washington, DC 20006	J. Christopher Moulder, Reg. No. 70,490 (cmoulder@willkie.com)
	Willkie Farr & Gallagher LLP 1875 K Street NW Washington, DC 20006

Petitioner has paid the required fee under 37 C.F.R. §§ 42.15(a) and 42.103 through P.R.P.S. Please assess any fee deficiency or credit to Deposit Account No. 232405.

D. *Service Information Under 37 C.F.R. § 42.8(b)(4)*

Petitioner consents to e-mail service at the addresses of lead and back-up counsel and Netflix-GTS-WFG@willkie.com. Hand delivery is also available to the addresses of lead and back-up counsel.

E. *Fees*

Petitioner has paid the required fee under 37 C.F.R. §§ 42.15(a) and 42.103 through P.R.P.S. Please assess any fee deficiency or credit to Deposit Account No. 232405.

XII. CONCLUSION

For the foregoing reasons, Petitioner respectfully requests that the Board institute *inter partes* review trial and cancel claims 1-20, 31, and 34 of the '865 patent as unpatentable.

Respectfully submitted,

Dated: April 20, 2023

By: //Aliza George Carrano //
Lead Counsel for Petitioner

CERTIFICATE OF COMPLIANCE

The undersigned certifies that this Petition complies with the applicable type-volume limitations of 37 CFR § 42.24. Exclusive of the portions exempted by 37 CFR § 42.24(a), this Petition contains 13,985 words as counted by the word processing program used for its preparation (Microsoft Word 2016) and is in compliance with the 14,000 word limit set forth in 37 C.F.R. § 42.24(a)(1)(i).

// Aliza George Carrano //
Lead Counsel for Petitioner

CERTIFICATE OF SERVICE

I hereby certify that on April 20, 2023, I caused a true and correct copy of the foregoing materials:

- Petition for *Inter Partes* Review of U.S. Patent No. 8,103,865
- Exhibits 1001-1030
- Petitioner's Powers of Attorney

to be served via: FedEx Express and e-mail to the Patent Owner's Attorney of

Record as follows:

David Stein
Brown Rudnick LLP
2211 Michelson Drive, Seventh Floor
Irvine CA, 92612
dstein@brownrudnick.com

Amir Alavi
Alavi & Anaipakos
3417 Mercer Street, Suite C
Houston, TX 77027
aalavi@aatriallaw.com

And to be served via: FedEx Express and e-mail to the PAIR Correspondence

Address for U.S. Patent No. Patent 8,103,865 as follows

David L. Wang
WILSON SONSINI GOODRICH & ROSATI
650 Page Mill Road
Palo Alto, CA 94304-1050
dawang@wsgr.com

IPR2023-00759
U.S. Patent No. 8,103,865

Aliza George Carrano

Lead Counsel for Petitioner