

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

NETFLIX, INC.,
Petitioner,

v.

GoTV Streaming, LLC,
Patent Owner

Case: IPR2023-00757
U.S. Patent No. 8,478,245

PETITION FOR *INTER PARTES* REVIEW

Petition Filing Date: April 7, 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 AND STUTORY GROUNDS (37 C.F.R. §42.204(B)).....	2
IV.	OVERVIEW OF THE '245 PATENT	3
A.	Client Devices Receiving Remotely Executed Applications.....	3
	Graphics, Rendering, and Graphical User Interfaces.....	4
	Wireless mobile devices	7
	Client-Server Architectures	8
B.	The Specification.....	12
C.	The Prosecution History	16
V.	LEVEL OF ORDINARY SKILL IN THE ART	17
VI.	CLAIM CONSTRUCTION	17
VII.	THE PRIMARY PRIOR ART REFERENCES	18
A.	Overview of Hariki.....	18
B.	Overview of Harris.....	20
C.	Other Evidence Regarding the State of the Art.....	24
VIII.	DETAILED EXPLANATION OF THE GROUNDS	24
A.	Ground 1: Hariki in view of Harris renders obvious Claims 1-33.....	24
1.	Claim 1	24
2.	Claim 2	52
3.	Claim 3	55

4.	Claim 4.....	56
5.	Claim 5.....	57
6.	Claim 6.....	58
7.	Claim 7.....	62
8.	Claim 8.....	62
9.	Claim 9.....	64
10.	Claim 10.....	65
11.	Claim 11.....	65
12.	Claim 12.....	66
13.	Claim 13.....	68
14.	Claim 14.....	68
15.	Claim 15.....	68
16.	Claim 16.....	69
17.	Claim 17.....	69
18.	Claim 18.....	69
19.	Claim 19.....	69
20.	Claim 20.....	70
21.	Claim 21.....	70
22.	Claim 22.....	70
23.	Claim 23.....	70
24.	Claim 24.....	78
25.	Claim 25.....	78
26.	Claim 26.....	80

27.	Claim 27	81
28.	Claim 28	81
29.	Claim 29	81
30.	Claim 30	81
31.	Claim 31	82
32.	Claim 32	82
33.	Claim 33	82
IX.	SECONDARY CONSIDERATIONS	83
X.	DISCRETIONARY DENIAL IS NOT WARRANTED UNDER EITHER § 314 OR § 325	83
A.	Fintiv Factors.....	83
B.	Becton Dickinson/Advanced Bionics/General Plastics Factors	86
XI.	MANDATORY NOTICES	87
A.	Real Parties-in-Interest.....	87
B.	Related Matters.....	87
C.	Lead and Back-Up Counsel, and Service Information Under 37 C.F.R. § 42.8(b)(3).....	87
D.	Service Information Under 37 C.F.R. § 42.8(b)(4).....	88
E.	Fees.....	88
XII.	CONCLUSION.....	88

EXHIBIT LIST

Exhibit Number	Description
1001	U.S. Patent No. 8,478,245 (“’245Pat”)
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. 8,478,245 (“’245FH”)
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 No. 7,447,486 (“ <i>Tamura</i> ”)
1008	U.S. Patent No. 6,996,627 (“ <i>Carden</i> ”)
1009	U.S. Patent No. 6,669,564 (“ <i>Young</i> ”)
1010	U.S. Patent No. 6,732,183 (“ <i>Graham</i> ”)
1011	U.S. Patent No. 6,507,727 (“ <i>Henrick</i> ”)
1012	Webpage excerpt from Computer History Museum at https://www.computerhistory.org/tdih/april/6/ (last accessed March 29, 2023)
1013	https://www.webdesignerdepot.com/2009/05/the-evolution-of-cell-phone-design-between-1983-2009/
1014	Excerpts from Dan R. Olsen, Jr., Developing user interfaces (1998)
1015	Alok Sinha, Client-server computing, in Communications of the ACM, 35, 7 (1992)
1016	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
1017	Eric Kasten, HTML: A Gentle Introduction, Linux Journal (July 1995),

Exhibit Number	Description
	available at https://www.linuxjournal.com/article/1081
1018	U.S. Patent Application Publication No. 2013/0124961 (“ <i>Linburn</i> ”)
1019	U.S. Patent No. 8,111,326 (“ <i>Talwar</i> ”)
1020	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)
1021	Complaint in <i>GovTV Streaming, LLC v. Netflix, Inc.</i> , No. 2:22-cv-07556-RGK-SHK (C.D. Cal. October 17, 2022)
1022	U.S. Patent Application Publication No. 2003/0106022 (“ <i>Goodacre</i> ”)
1023	https://web.archive.org/web/20070406171650/http://www.zenzui.com/products.html

LISTING OF CHALLENGED CLAIMS

Claim 1	
[1pre]	A method of rendering content on a wireless device, said method comprising:
1[a]	receiving an identification of a custom configuration of a plurality of rendering blocks of said wireless device,
1[b]	wherein said custom configuration is associated with an application and configures said plurality of rendering blocks to render content in a manner customized to said application;
[1c]	receiving compiled content generated in part from execution of said application
[1d]	wherein said compiled content comprises render commands expressed in a syntax that is generic to said wireless device;
[1e]	using a graphical user interface comprising said plurality of rendering blocks to generate renderable content based on said compiled content and said custom configuration; and
[1f]	rendering said renderable content on said wireless device, wherein said receiving compiled content comprises: receiving first compiled content specific to a first page of said application; and receiving second compiled content specific to a second page of said application
[1g]	wherein said custom configuration is applicable to both said first and second compiled content.
Claim 2	
[2]	A method as described in claim 1 wherein said using a graphical user interface comprising said plurality of rendering blocks to generate renderable content comprises: processing said compiled content using a reader of said wireless device; and issuing commands from said reader to individual rendering blocks of said graphical user interface based on said rendering commands of said compiled content.

Claim 3	
[3]	A method as described in claim 1 wherein said render able content comprises audio content and display content.
Claim 4	
[4]	A method as described in claim 1 wherein said compiled content is partially resultant from said application operating on a remote server.
Claim 5	
[5]	A method as described in claim 1 wherein said compiled content is specific to the rendering capabilities of said wireless device.
Claim 6	
[6]	A method as described in claim 1 wherein each of said plurality of rendering blocks operates specific to a wireless device type of said wireless device and each is instructed using a syntax that is generic to said wireless device type.
Claim 7	
[7]	A method as described in claim 6 wherein said custom configuration comprises a syntax that is generic regarding said wireless device type.
Claim 8	
[8]	A method as described in claim 1 wherein said custom configuration comprises configuration information and content specific to said application.
Claim 9	
[9]	A method as described in claim 1 wherein said custom configuration is one of a plurality of memory-stored custom configurations stored by said wireless device and wherein further said identifying said custom configuration comprises receiving an identifier that identifies said custom configuration.
Claim 10	

[10]	A method as described in claim 1 further comprising receiving and storing said custom configuration.
Claim 11	
[11]	A method as described in claim 1 wherein said plurality of rendering blocks of said graphical user interface comprises: a first block that controls the rendering of ticker information across a display screen of said wireless device; a second block that controls the rendering of button images on said display screen; and a third block that controls the rendering of audio on a speaker of said wireless device.
Claim 12	
[12pre]	A non-transitory computer usable medium comprising instructions therein that when executed by a processor implement a method of rendering content on a wireless device, said method comprising:
[12a]	receiving an identification of a custom configuration of a plurality of rendering blocks of said wireless device,
[12b]	wherein said custom configuration is associated with an application and configures said plurality of rendering blocks to render content in a manner customized to said application;
[12c]	receiving compiled content generated in part from execution of said application
[12d]	wherein said compiled content comprises render commands expressed in a syntax that is generic to said wireless device;
[12e]	using a graphical user interface comprising said plurality of rendering blocks to generate renderable content based on said compiled content and said custom configuration; and
[12f]	rendering said renderable content on said wireless device, wherein said receiving compiled content comprises: receiving first compiled content specific to a first page of said application; and receiving second compiled content specific to a second page of said application,

[12g]	wherein said custom configuration is applicable to both said first and second compiled content.
Claim 13	
[13]	A non-transitory computer usable medium as described in claim 12 wherein said using a graphical user interface comprising said plurality of rendering blocks to generate renderable content comprises: processing said compiled content using a reader of said wireless device; and issuing commands from said reader to individual rendering blocks of said graphical user interface based on said rendering commands of said compiled content.
Claim 14	
[14]	A non-transitory computer usable medium as described in claim 12 wherein said renderable content comprises audio content and display content.
Claim 15	
[15]	A non-transitory computer usable medium as described in claim 12 wherein said compiled content is partially result ant from said application operating on a remote server.
Claim 16	
[16]	A non-transitory computer usable medium as described in claim 12 wherein said compiled content is specific to the rendering capabilities of said wireless device.
Claim 17	
[17]	A non-transitory computer usable medium as described in claim 12 wherein each of said plurality of rendering blocks operates specific to a wireless device type of said wireless device and each is instructed using a syntax that is generic to said wireless device type.
Claim 18	

[18]	A non-transitory computer usable medium as described in claim 17 wherein said custom configuration comprises a Syntax that is generic regarding said wireless device type.
Claim 19	
[19]	A non-transitory computer usable medium as described in claim 12 wherein said custom configuration comprises configuration information and content specific to said application.
Claim 20	
[20]	A non-transitory computer usable medium as described in claim 12 wherein said custom configuration is one of a plurality of memory-stored custom configurations stored by said wireless device and wherein further said identifying said custom configuration comprises receiving an identifier that identifies said custom configuration.
Claim 21	
[21]	A non-transitory computer usable medium as described in claim 12 wherein said method further comprises receiving and storing said custom configuration.
Claim 22	
[22]	A non-transitory computer usable medium as described in claim 12 wherein said plurality of rendering blocks of said graphical user interface comprises: a first block that controls the rendering of ticker information across a display screen of said wireless device; a second block that controls the rendering of button images on said display screen; and a third block that controls the rendering of audio on a speaker of said wireless device.
Claim 23	
[23pre]	A wireless device operable to communicate with a remote server, said wireless device comprising:
[23a]	a transceiver coupled to a bus and operable to receive a custom configuration that is associated with an application,
[23b]	said transceiver also operable to receive compiled content generated in part from execution of said application and

[23c]	comprising a plurality of rendering commands expressed in a syntax that is generic to said wireless device;
[23d]	a memory coupled to said bus and operable to store said compiled content and said custom configuration;
[23e]	a processor coupled to said bus and operable to implement the following: a graphical user interface comprising a plurality of rendering blocks and operable to generate renderable content based on said compiled content and said custom configuration
[23f]	wherein said custom configuration is operable to configure said plurality of rendering blocks to render content in a manner customized to said application;
[23g]	an engine for reading said compiled content and responsive thereto for causing said graphical user interface to generate said renderable content based on said render commands;
[23h]	a display device coupled to said bus and operable to render a portion of said renderable content, wherein said compiled content comprises: a first compiled content specific to a first page of said application; and a second compiled content specific to a second page of said application, wherein said custom configuration is applicable to both said first and second compiled content.
Claim 24	
[24]	The wireless device as described in claim 23 wherein said engine is further operable to issue commands to individual rendering blocks of said graphical user interface based on said plurality of rendering commands of said compiled content.
Claim 25	
[25]	The wireless device as described in claim 23 further comprising an audio rendering device coupled to said bus and wherein said renderable content comprises audio content and display content.
Claim 26	
[26]	The wireless device as described in claim 23 wherein 25 said compiled content is partially resultant from said application operating on a remote server.
Claim 27	

[27]	The wireless device as described in claim 23 wherein said compiled content is specific to the rendering capabilities of said wireless device.
Claim 28	
[28]	The wireless device as described in claim 23 wherein each of said plurality of rendering blocks operates specific to a wireless device type of said wireless device and each is instructed using a syntax that is generic to said wireless device type.
Claim 29	
[29]	The wireless device as described in claim 28 wherein said custom configuration is expressed in a syntax that is generic to said wireless device type.
Claim 30	
[30]	The wireless device as described in claim 23 wherein said custom configuration comprises configuration information and content specific to said application.
Claim 31	
[31]	The wireless device as described in claim 23 wherein said custom configuration is one of a plurality of memory-stored custom configurations stored in said memory.
Claim 32	
[32]	The wireless device as described in claim 23 wherein said plurality of rendering blocks of said graphical user interface comprises: a first block that controls the rendering of ticker information across said display screen of said wireless device; a second block that controls the rendering of button images on said display screen; and a third block that controls the rendering of audio on a speaker of said wireless device.
Claim 33	

[33]	The wireless device as described in claim 23 wherein said plurality of rendering blocks of said graphical user interface comprises: a first block that controls the rendering of a scroll bar on said display screen of said wireless device; a second block that controls the rendering of an input box on said display screen of said wireless device; and a third block that control the rendering of a text display region on said display screen of said wireless device.
------	---

I. INTRODUCTION

Netflix, Inc. (“Petitioner”) respectfully requests *inter partes* review of claims 1-33 of U.S. Patent No. 8,478,245 (“the ’245 patent” or “’245Pat”), filed on August 1, 2007, 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. Please assess any fee deficiency or credit to Deposit Account No. 232405.

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

Petitioner certifies that the ’245 patent is available for *inter partes* review (“IPR”). 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) (served 10/25/2022).

III. STATEMENT OF PRECISE RELIEF REQUESTED FOR EACH CLAIM CHALLENGED AND STUTORY 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 Error! Reference source not found.).
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 Error! Reference source not found.).

Ground	Claims	Basis
1	1-33	35 U.S.C. §103 <i>Hariki</i> in view of <i>Harris</i>

¹ Petitioners do not concede that any challenged claims satisfy other requirements for patentability that cannot be raised in IPR, including 35 U.S.C. §§101 and 112.

IV. OVERVIEW OF THE '245 PATENT²

A. *Client Devices Receiving Remotely Executed Applications*

Downloading content to a client device 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. Users were turning from traditional media sources, such as television and newspaper, to the Internet to receive media content. **EXError! Reference source not found.**, 1:12-16.³ A user might go to media servers containing information from various content providers. *Id.*, 1:16-19. For example, techniques for transmitting/receiving data via a cellular

² This background and the Grounds of Unpatentability herein, are further supported by the Declaration of Petitioner's expert, Dr. Ben Bederson, who has over 30 years of experience related to remote application execution. (**EXError! Reference source not found.**)

³ Additional references cited in this Petition that are not part of the unpatentability grounds are used to demonstrate 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.”)

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.

EXError! Reference source not found., 1:17-20.

Those in the art recognized that, at least by 2007, servers could provide to a remote client automatic delivery of multimedia content, such as entertainment content. **EXError! Reference source not found.**, 1:29-34; **EXError! Reference source not found.**, Abstract; *see also* **EXError! Reference source not found.**, Abstract. Prior to the '245 patent, it was a known problem that transmission of content was limited by, for example, bandwidth and connection rates of mobile devices. **EXError! Reference source not found.**, 1:42-57. A common goal was to improve network efficiency from a transmitter to a terminal. *Id.*

As described below, client/server architectures, rendering technologies, and wireless devices were well-known to POSITAs at the time of the alleged inventions claimed in the '245 Patent. *Bederson*, ¶¶46.

Graphics, Rendering, and Graphical User Interfaces

In computing, whether mobile or desktop, a user interface is the boundary where interactions between humans and computers occur. An early type of user interface is the command-line interface, where the user enters text commands to the computer via the keyboard and the computer executes those commands. The command-line interface was simple and efficient, but it suffered from several

drawbacks. For instance, the interface was difficult for users to learn, because to operate the computer, they first had to learn all the commands. Generally, commands in command-line interfaces were not expressed in natural language, but rather arbitrary abbreviations and acronyms that users had to memorize. For example, in the early Microsoft DOS operation system, text commands included “cd,” “dir,” and “chkdsk.” This resulted in a user interface that was not very intuitive—a novice user could not use the user interface by simply experimenting with the interface.

An advancement to the command-line interface was the Graphical User Interface (“GUI”). GUIs offer visual interfaces on display screens that are designed for simplicity and ease of learning. A common and well-known example of a GUI uses the “desktop metaphor,” where files and applications of the computer system are represented as graphical icons on a virtual desktop shown on the computer screen, and where the user interacts with those icons using a pointing device such as a mouse cursor. Windows 3.1, released by Microsoft in 1992,⁴ was one example of such a GUI:

⁴ See, e.g., <https://www.computerhistory.org/tdih/april/6/> (last accessed March 29, 2023) (EXError! Reference source not found.).

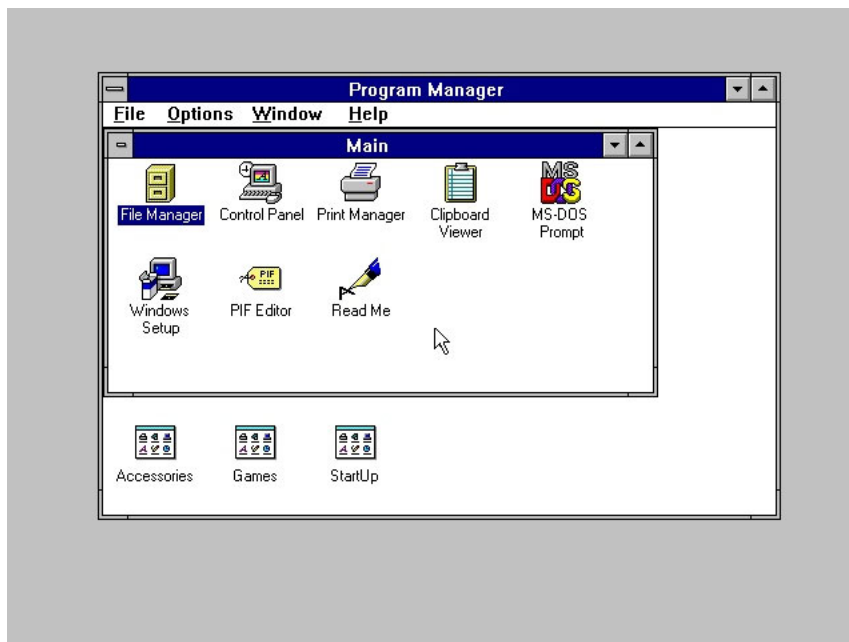


Figure A. Screenshot of Microsoft Windows 3.1 taken by Benjamin B. Bederson on March 28, 2023 using an emulator running in VMWare Fusion. *Bederson*, ¶48.

One main goal of GUIs was to make the user interface more intuitive and easy to learn by implementing concepts that are analogues to the real—dragging an icon to move across the computer screen “desktop,” for example, resembles the act of moving an object in the real world. Part of the process for generating GUIs involves “rendering” the graphics—that is, deciding which specific color each pixel on the screen should be so as to form the visual elements of the interface. *Id.*, ¶50. This rendering technology has been well-known for decades. *Id.*, ¶50. For example, a 1998 book titled “Developing User Interfaces” describes many of the technical elements of this rendering. *Id.*, ¶50 (citing Dan R. Olsen, Jr., *Developing user interfaces* (1998) (EX). For example, on page 51, it discusses “the basic 2D

primitives that are required to present information to the users so they can interactively manipulate it.” *Id.*, p. 51. It covers topics including the coordinate systems of 2D computer graphics, how to represent shapes such as lines with pixels, how to manage the drawing of text at different sizes with different fonts, etc. *Bederson*, ¶50.

By 2007, such rendering technologies were fairly standardized and built into essentially every computer operating system that supported GUIs, including desktop and mobile devices. While many systems rendered application GUIs on the same device that executed the application, there were also other well-known architectures that rendered GUIs on remote devices (e.g., “clients”). In this case, the application ran on other computers and sent graphics commands to a client computer to be rendered.

Wireless mobile devices

Mobile phones, a common type of wireless device, have been around since the 1980s. In the 2000s, mobile phones using Third Generation technology, or 3G, were introduced. This technology was capable of transferring voice data (such as a phone call) and transferring other types of data, including emails and instant messages. By 2000, wireless phones included touch screens and increased graphics capacity. In 2003, the PalmOne Treo 600 and BlackBerry 7210 included color

screens. **EXError! Reference source not found.** PDAs (Personal Digital Assistants) were another common wireless device available in the early 2000s.

Client-Server Architectures

Historically, computers were big and expensive and they provided “time shared” access to multiple terminals. Sometimes those terminals (a.k.a. “clients”) were “dumb” or “thin,” and provided primarily input and output capabilities while most of the computation was provided at the “server.” Other times, those clients had greater computation capability. In the early 1980s with the introduction of “personal computers,” those clients (while still often connected to servers) had sufficient computational capabilities to run independently.

The relationship between “clients” and “servers” is so foundational to the history of computers that it is often used to define the nature of the computing architecture as well as the user experience. The state of client-server computing was described in a 1992 article in Communications of the ACM—a broad interest magazine of the main professional international computer society. *See EXError! Reference source not found. (“Sinha”).* *Sinha* characterized some of the key elements of client-side computing, including that the client typically communicates with the server, transmitting queries or commands, performs data analysis on query or command results from the server and presents the GUI. On the server-side, servers typically provide a service to the client, responding to the queries or

commands—and do so in a way that hides much of the details of its implementation so that clients with various kinds of software and hardware can work with it. *Sinha* also explains that as far back as 1992, it was common to have the GUI of an application displayed on a computer remote from the application. It describes the X Window System that commonly ran on Unix environments which “provides network transparency so that an application displaying a chart on a screen could be running on a local machine or a remote machine thousands of miles away.” *Id.*, 91.

One particularly important set of clients and servers communicate over the Internet in what is called the World Wide Web (“WWW”), which was initially developed in 1990. While Internet-based communications had been around for some time at that point, the WWW simplified things by combining a few basic technologies, including web servers, web browsers, HTTP, and HTML. The Internet was an enabling technology, allowing computers to communicate arbitrary information between any location in the world. *See* **EXError! Reference source not found..**

Web servers are computers on the Internet that run software to listen to web “requests” and respond to those requests, typically by returning (i.e., serving) a document that was requested. Subject to hardware resource limitations, a web server may serve any number of web pages. Static web pages may be stored on a server’s hard drives and processed in memory. Server-generated web pages, such

as those generated in response to a user's search request, are generated and processed in memory. *See EXError! Reference source not found.*, 76-78. Web servers used the Hypertext Transfer Protocol (HTTP), a simple request/response network protocol that was developed, which allows an HTTP connection to be made from a client to a host.

Web browsers are client applications, that provide a user interface for displaying data fetched from web servers. Somewhat like a word processing program that can open a document and render it, a web browser can open a special kind of document written using Hypertext Markup Language (HTML) and render it. However, web browsers have additional features including that they can fetch those documents across the Internet from web servers using the HTTP protocol. Web browser requests are made to specific servers identified by Uniform Resource Locators ("URLs"). HTML documents can include references to images or other data, such as audio files, that are specified via URLs and are fetched from specific web servers via those URLs. The browser then displays those separately retrieved images integrated with the textual document. Web browsers also can interpret the HTML documents they display to recognize hyperlinks that, when clicked on by a user, fetch additional documents or data referenced by another URL specified by the HTML's hyperlink. If the link points to a document, the newly fetched HTML document is displayed, typically replacing the previously displayed document. In

1994, these URLs identified a single specific server, offering a “Universal” way of locating and accessing for any network communications protocol, including HTTP. *See* **EXError! Reference source not found.**, 76-78.

HTML is a textual document format for describing the visual presentation of documents, including embedded objects such as images and forms as well as links to other documents. It consists of plain text that will be displayed as such along with “tags” which surround portions of text and are used to modify how that text is displayed. Tags are formed by a word surrounded by a pair of angle brackets with a matching closing tag that is prefixed with a forward slash. For example, this is a simple HTML snippet: “Here is some bold text.” *See* **EXError! Reference source not found.** When this HTML is interpreted by a web browser, it would be displayed as following:

Here is some **bold text**.

HTML can include links to other documents. It specifies those links with an “anchor” tag (i.e., <A...> and a closing tag) along with URLs to indicate the location of the other document. URLs are created in text with a specific structured format including several elements such as the address (often represented by a domain name) of a web server the other document is on, and the location (or “path”) of that document on the web server. An example of an anchor tag follows where the “HREF” parameter specifies the URL of the document to be fetched and

displayed when the link is clicked (*see* **EXError! Reference source not found.**, 3):

Foodocument

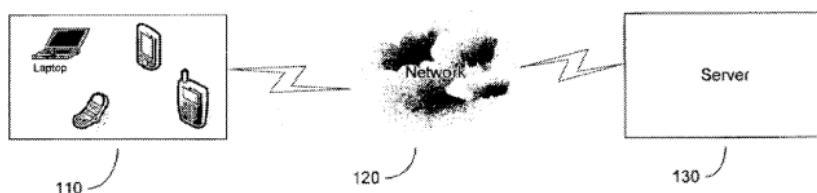
These examples demonstrate that HTML provides a syntax that is generic and can be used to define applications in a way that can work on client computer. Generally, web sites are collections of HTML-based web pages that are stored or generated on a web server that is typically accessed by a single domain address. In order for a user to view those web sites, they would direct their web browser to load the web page located at a specific URL. At that point, the web browser would communicate with the web server specified by the URL via the World Wide Web to fetch the desired web page and would then render the page for display. *See* **EXError! Reference source not found.**, 1-3. Web pages supported multimedia and rich interactive content well before 2007. A key part of HTML is that it allowed the ability to download separately defined images or icons from web servers and embed them in a web page.

B. The Specification

The '245 patent discloses “[a] method of rendering content on a wireless device.” ’245*Pat*, Abstract. The specification explains “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 are coupled to a server 130 through a network 120.” *Id.*, 6:1-3.

100A



'245Pat, FIG. 1.

Figure 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

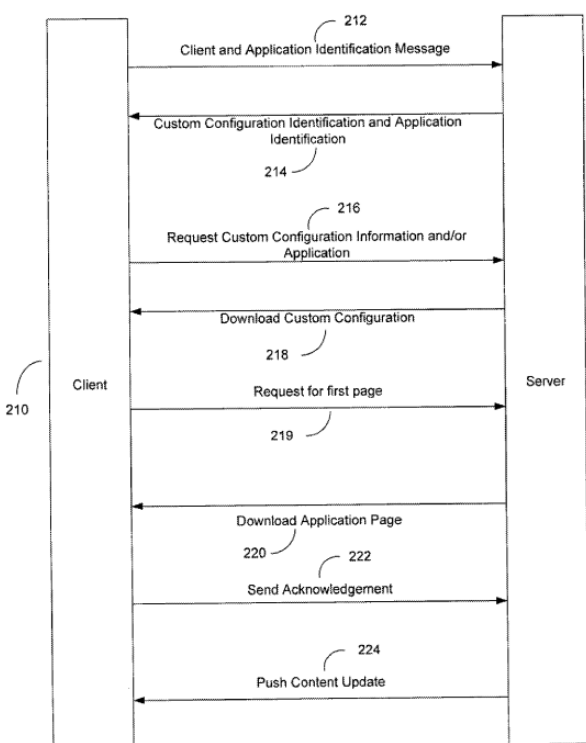


FIGURE 2A

200B

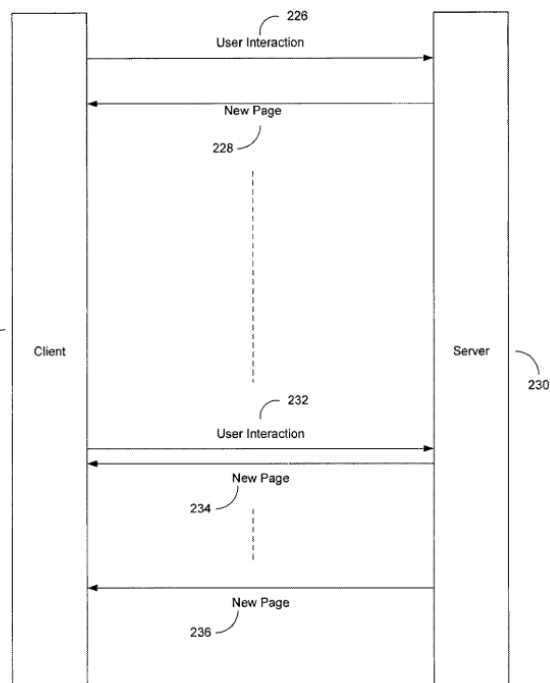
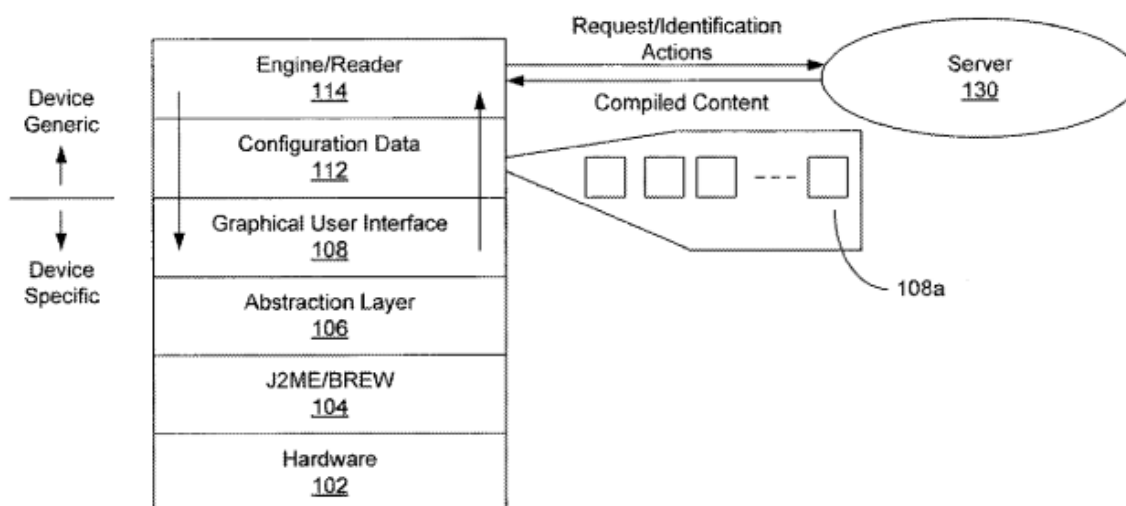


FIGURE 2B

'245Pat, 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

'245Pat, 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 wireless device’s GUI 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 '245 patent issued on July 2, 2013, from U.S. Patent Application 11/888,803 (“the '803 application”), filed August 1, 2007. The application that led to the '245 patent was prosecuted under the pre-AIA statutory regime. '245PH, 206.

On December 13, 2012, Applicant amended the independent claims to recite “wherein said receiving compiled content comprises: receiving first compiled content specific to a first page of said application; and receiving second compiled content specific to a second page of said application, wherein said custom configuration is applicable to both said first and second compiled content.” *Id.*,

189. Applicant also changed “computer readable media” to “non-transitory computer usable medium” in eleven claims. These amendments overcame a previous non-final rejection mailed by the USPTO on June 14, 2012. ’245FH, 34-38. On December 31, 2012, following the amendments, the claims were allowed.

V. LEVEL OF ORDINARY SKILL IN THE ART

Persons of ordinary skill in the art (“POSITAs”) at the time of the earliest available priority date (August 1, 2007) for the ’245 patent would have had a bachelor’s degree in electrical or computer engineering, or closely related scientific field such as computer science, and two to three 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 ’245 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, Petitioners believe 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.⁵ *Phillips v. AWH Corp.*, 415 F.3d 1303, 1312–13 (Fed. Cir. 2005) (en banc).

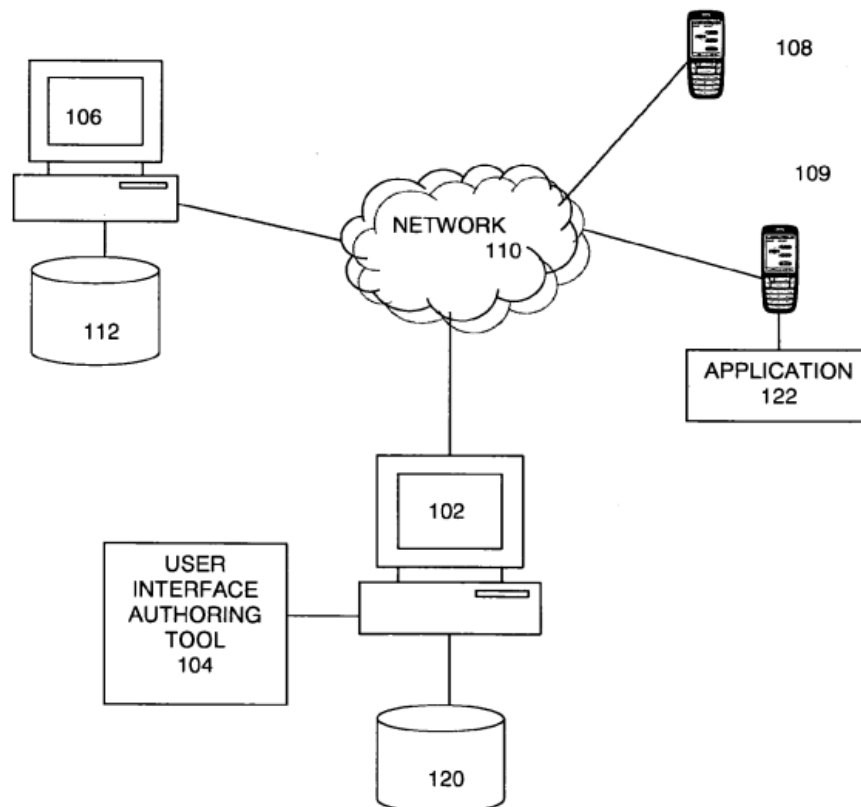
VII. THE PRIMARY PRIOR ART REFERENCES

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 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 mobile device 108 receives content data, application programs, diagnostic tools, program components, or any other content or executable objects from “a server computer 102 operated by the service provider or

⁵ Petitioner reserves all rights to raise claim construction arguments and other arguments in any litigation concerning the ’245 patent. For example, comparing the claims to the accused products in the litigation may raise controversies that require construction of certain claim terms.

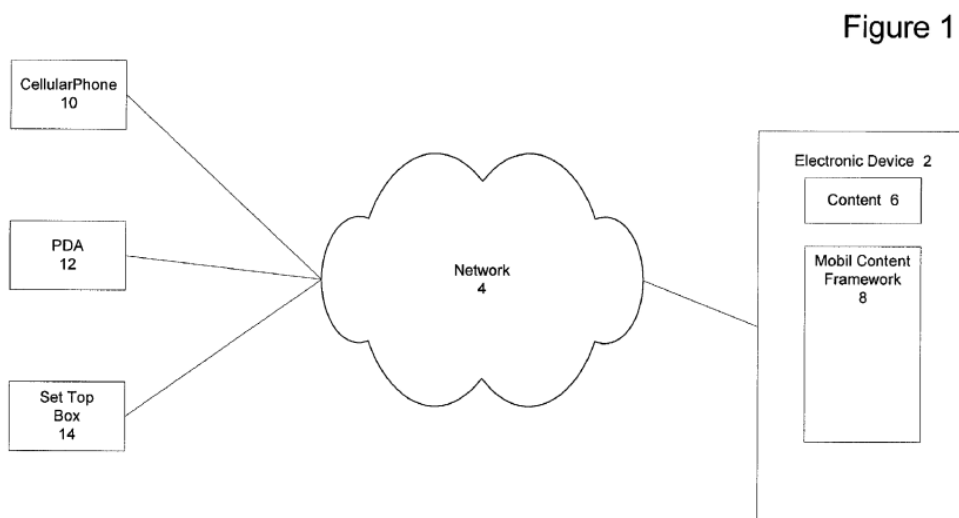
a third party vendor.” *Id.*, ¶21. The mobile device receives HTML pages from a server executing a web server process. *Id.* Furthermore, the mobile device receives “customized user interface [generated] for a plurality of different makes and types of mobile devices” by a server that “allow[s] a user 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 received by the mobile device 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 content delivery system where mobile devices request content from a platform, which “enables a content developer to distribute uniform content to multiple 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 [sic] devices without

providing different versions of the content.” *Id. Harris* further explains that based on the resource availability of requesting devices, “[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 requesting mobile device “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 (below). The environment includes a plurality of mobile devices (cellular phone 10, PDA 12) interfaced through a network to a server (electronic device 2) with content and an MCF. *Id.*, ¶15.



Harris, FIG. 1

Harris explains, with reference to Figure 3, below, that a wireless device requests content from the server (step 60). *Harris*, ¶24. 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 “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, the requesting device receives the customized content (step 76).*Id.*

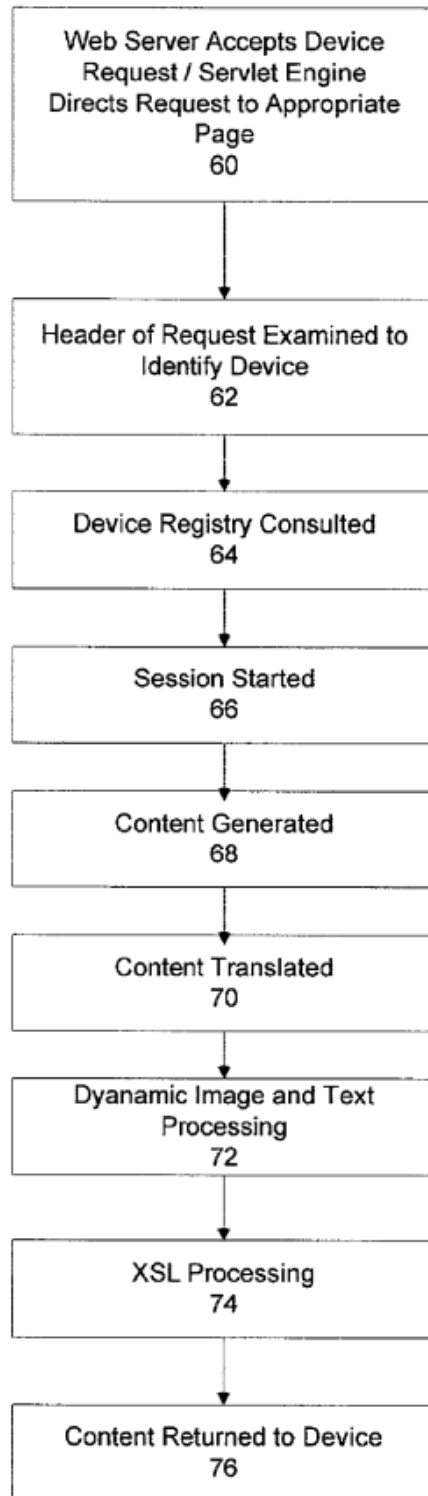


Figure 3

C. Other Evidence Regarding the State of the Art

Prior art in Exhibits 1007-1023 further reflects the state of the art, and is relevant to the background of the '245 patent and the invalidity analysis herein. *Bederson*, ¶75.

VIII. DETAILED EXPLANATION OF THE GROUNDS

A. Ground 1: Hariki in view of Harris renders obvious Claims 1-33

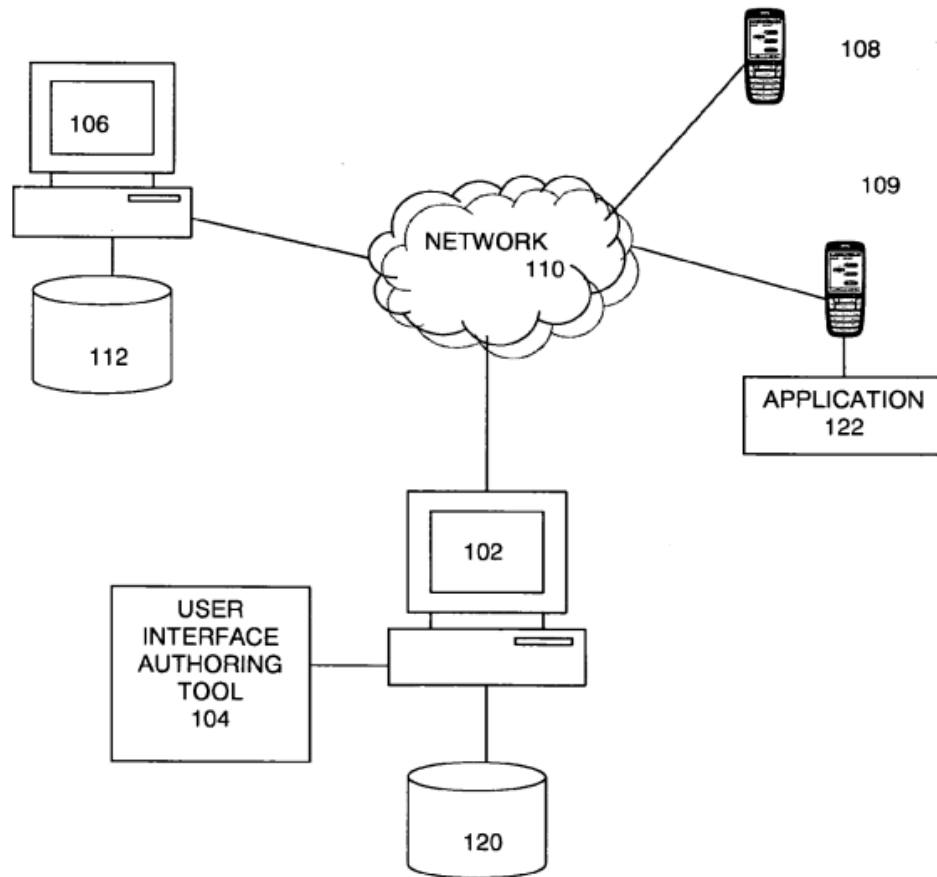
1. Claim 1

a. [1pre]

To the extent the preamble is limiting, *Hariki* discloses this feature. *Bederson* ¶¶105-113. For example, *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,”⁶ such as HTML files over a network to the mobile devices. *Hariki*, ¶21. The mobile devices “run[] a web browser program to access the web pages served by server.” *Id.* 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

⁶ Unless otherwise stated, all emphases are added.

program environment of a device by altering **display and/or sound output aspects of the device.**” *Id.* The UI skin for a target mobile device is contained in a “UI content package.” *Id.*, ¶33. The “UI content package can [] be **downloaded to the target mobile device.**” *Id.*, ¶36. Additionally, “the **mobile device ... is configured to run one or more application programs ...to execute or access the resources provided by the UI content package.**” *Id.*, ¶40. Thus, *Hariki’s* mobile device receives “a customized UI that allows the user to change certain aspects of the UI skin.” *Id.*, ¶26.

**FIGURE 1***Hariki, FIG. 1*

The '245 patent does not disclose the nature of “content that is renderable”; however, it acknowledges that “renderable content comprises audio content and

display content.”⁷ *See, e.g., ’ 245Pat*, 21:6-7 (Claim 3). POSITAs would have understood, based on the ’245 patent’s specification, that “rendering” means anything that can eventually be displayed (e.g., rendered) on a mobile device. *Bederson*, ¶108.

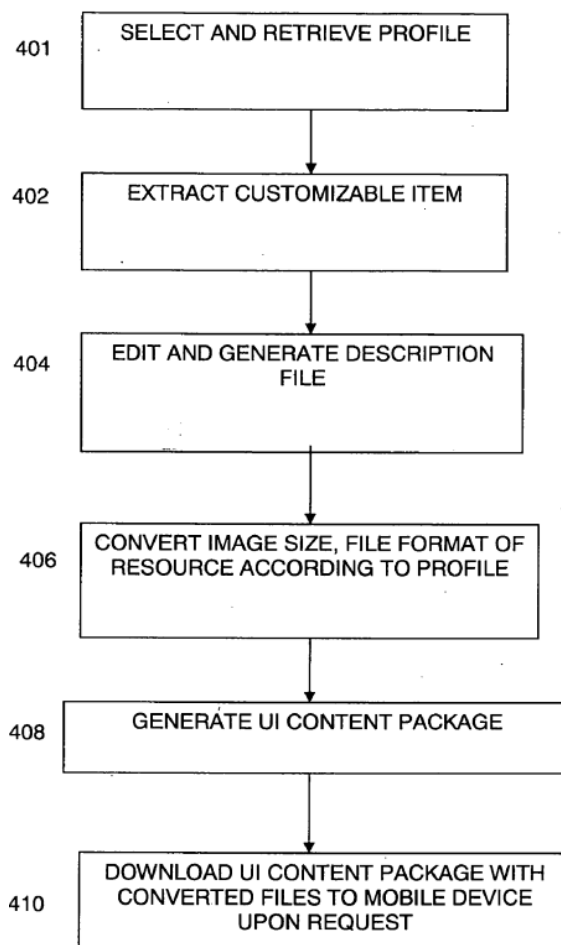
Because the mobile device in *Hariki* downloads and runs executable content such as a UI skin that alters the “look and feel” of its display and/or sound, and comprises audio content and display content, POSITAs would have understood that the mobile device renders content. *Bederson* ¶¶109-110.

Hariki’s disclosed mobile devices 108 and 109, which may be cellular phones, are “wireless devices.” *Id.*, ¶20. With reference to Figure 1 above, *Hariki* explains “mobile devices 108 and 109 communicate with each other and other

⁷ The ’245 patent functionally describes “renderable content.” Specifically, it discloses “the wireless device uses a graphical user interface to generate renderable content based on the received compiled content and the custom configuration.” *’245Pat*, 18:63-65. The specification explains “content is rendered by processing the compiled content using the reader and by issuing commands from the reader to rendering blocks of the graphical user interface based on the rendering commands of the compiled content.” *Id.*, 18-66-19:2.

mobile devices over the wireless portion of network 110.” *Id*POSITAs would have understood that the mobile devices are wireless devices. *Bederson*, ¶111.

Hariki further discloses “a **method** of generating and downloading UI skins.” *Hariki*, ¶36. For example 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 (i.e., the UI skin) alters the display of the mobile device. *Id.*, ¶24; *Bederson*, ¶112.

**FIGURE 4**

Hariki, FIG. 4

Accordingly, *Hariki* discloses a method of rendering a UI skin to change the look and feel of display or sound (i.e., “method of rendering content”) of a mobile device (i.e., a “wireless device”). *Bederson*, ¶113.

b. [1a]

Hariki discloses this feature. *Bederson* ¶¶114-124. As explained above, the mobile device receives HTML files and a customized UI skin (e.g., UI package)

that alters its look and feel. *See* Section VIII.A.1.a. *Hariki* discloses that this alteration is based on receiving at the mobile device a “downloadable UI content package consisting of [a] description file and [] converted resource files.” *Hariki*, ¶19. As explained below, POSITAs would have understood that the converted resource files and the description file in *Hariki*’s UI content package⁸ corresponded to the claimed “custom configuration” and “identification of [the] custom configuration,” respectively. *Bederson*, ¶114.

Hariki explains that a specific content package downloaded at a requesting mobile device was received from a “server computer 102 [that] is operated by a content provider, and executes a user interface authoring tool 104 that generates [that] content package.” *Hariki*, ¶25. The user interface authoring tool “can represent a program or **suite of programs**, or even hardware circuits, or any combination thereof embodying instructions executed by one or more processing units in server 102.” *Id.*, ¶27.

⁸ *Hariki* uses several terms interchangeably for UI content package. For example, *Hariki* describes it as “UI content” ¶19, “UI skin content” (*Hariki*, ¶27), “UI skin package data” (*id.*, ¶27), and “UI package” (*id.*, ¶40). *Bederson*, ¶114. For the purposes of this petition, Petitioner will uniformly refer to the downloadable UI content as “UI content package.”

Hariki explains that the UI skin may consist “of a particular screen configuration, size, color scheme, font, menu scheme, keypad button assignment, ringtone selection.” *Id.*, ¶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 can configure “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 ’245 patent. *Bederson*, ¶117. Indeed, the ’245 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.

’245Pat, 8:27-43.

POSITAs would have understood that *Hariki*'s configurable buttons and sounds, for example, fall squarely within the '245 patent's examples of "rendering blocks." *Bederson*, ¶118.

When received by the mobile device, the "[u]ser interface descriptor information comprising the customizable UI elements are downloaded."⁹ *Id.*, ¶27.

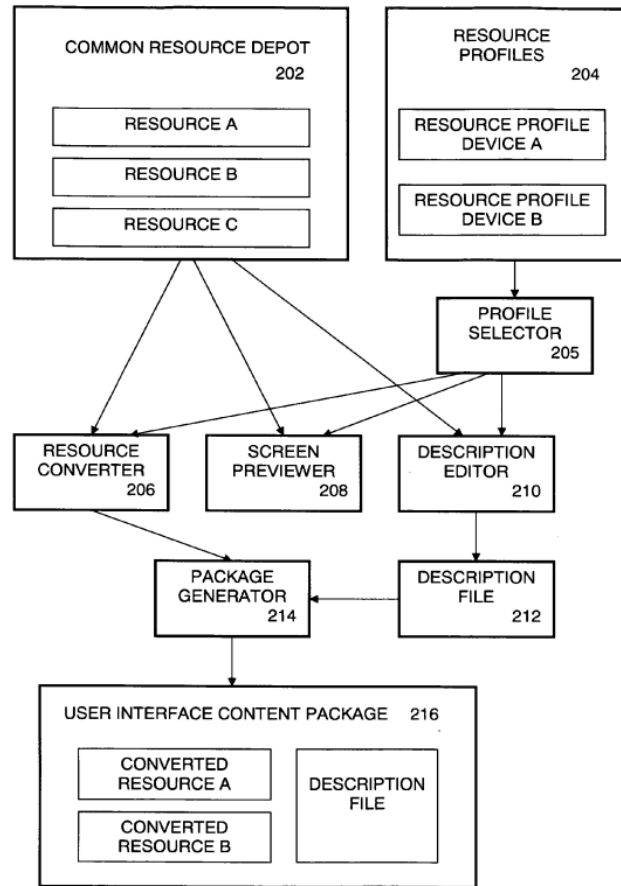
For example, *Hariki* discloses the UI content package:

can comprise screen parameter definitions (e.g., size, aspect ratio, **icon definitions**, and so on), **images, video clips, music or other sound clips**, ringtones, games, small applications (applets), utilities, diagnostic tools, or any other similar data or applications, all of which are referred to as "UI content objects."

Id.

The UI content can either be downloaded by the mobile device or on demand. *Id.*

⁹ Here *Hariki* uses the term "UI descriptor information" interchangeably with UI content package. *Bederson*, ¶119.

**FIGURE 2***Hariki*, FIG. 2

Regarding Figure 2, above, *Hariki* further explains that “UI skin package data specific to each mobile device is generated from common resource data using resource profile information for each mobile device model.” *Id.*, ¶29. *Hariki* explains that “description editor component 210 produces description files 212 based on the selected resource profile 204 and resource file 202” (*id.*, ¶33) and “resource converter 206 converts each resource into a format corresponding to the resource profile” (*id.*, ¶32). Furthermore, **“description file 212 specifies the**

resources and the file type, and file path name.” *Id.*, ¶33. In other words, “[t]he description file contains references to one or more [converted] resources.” *Id.*, ¶46; *Bederson*, ¶121. “The UI content package comprises the appropriate converted resources and the description file.” *Id.*, ¶33.

The configurable aspects of the user interface may be configurable buttons and sounds (i.e., “rendering blocks”). *Id.*, ¶24; *Bederson*, ¶122. *Hariki* explains that the 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 the converted resources in the UI package configure the configurable aspects of the UI. *Bederson*, ¶122. Accordingly, POSITAs would have understood the converted resources as a “custom configuration” of the configurable buttons and sounds (i.e., “plurality of rendering blocks”). *Id.*, ¶122.

Hariki further explains that the description file “may reference resources contained within the same UI content package, or it may reference resources in other content packages that are either pre-installed in the mobile handset or on an external server computer.” *Hariki*, ¶46. *Hariki* discloses that a mobile device receives the UI content package containing the description file from a content server. *Id.*, ¶45. The description file received by the mobile device identifies resources with “certain items of information, such as identifier (ID), path

(location), and file type, among other parameters, relating to a particular resource.” *Id.*, ¶48. The mobile device then uses the received description file to download resources by “request[ing] a resource by specifying the resource ID (e.g., ID 1)” *Id.*, ¶50. POSITAs would have thus understood that the description file received by the mobile device identifies the resources (i.e., “custom configuration”) in the UI skin. *Bederson*, ¶123.

Accordingly, *Hariki* discloses a mobile device receiving (i.e., “receiving” at “wireless device”) a UI package including converted resources (i.e., the claimed “custom configuration”) a description file (i.e., the claimed “identification”), wherein the UI skin package data configures the look and feel of display and sound aspects of the mobile device (i.e., the claimed “rendering blocks”). *Id.*, ¶124.

c. [1b]

Hariki discloses this feature. *Bederson* ¶¶125-129. For example, as explained above, *Hariki* discloses converted resources (“custom configuration”) in a UI content package that alter the look and feel of aspects of the display and sound of the mobile device, such as backgrounds and sounds (“configures said plurality of rendering blocks”). See Section VIII.A.1.b. *Hariki* discloses the UI skin allows customization of the look and feel of the “application program environment of a device.” *Hariki*, ¶24. The customization is effected by the UI content package. *Id.*, ¶27; *Bederson*, ¶125. *Hariki* explains that “the resources

referenced by the descriptor file and **utilized by the application** must be referenced in some way by the application and then retrieved and converted to an appropriate format **for use by the application.**” *Hariki*, ¶39. The application may be “a web browser program to access the web pages served by server computer.” *Id.*, ¶21. Accordingly, POSITAs would have understood that the converted resources (i.e., “custom configuration”) in the UI content package are associated with the web browser (i.e., “an application”). *Bederson*, ¶126.

To the extent PO contends that the “associated” application must be remote to the mobile device,¹⁰ *Hariki* discloses this also. For example, as explained above, *Hariki* discloses the user interface authoring tool represents a suite of programs on the server and that the server also provides content data such as HTML files through a web server process. *See* Section VIII.A.1.a. POSITAs would have recognized the user interface authoring tool and web server process were part of

¹⁰ POSITAs would have understood that the “application” is not limited to one residing only on a client or only on a server. *Bederson*, ¶127. For example, claim 3 (below) further limits the “application” in claim 1 to “operat[e] on a remote server. *See* Section VIII.A.3. Thus, POSITAs would have understood that the “application” in claim 1 could reside anywhere, for example on a client or on a server. *Bederson*, ¶127.

the same suite of programs because they reside on the server together and that the authoring tool generated the custom UI for mobile devices accessing the web server process. *Bederson*, ¶128. Thus, POSITAs would have understood that the converted resources (i.e., “custom configuration”) in the UI content package were associated also with the web server process that serves HTML files to the mobile device. *Id.*, ¶128.

Furthermore, *Hariki* explains that the UI content package is specific to the mobile device type and the application programs. *Hariki*, ¶27. Accordingly, POSITAs would have understood that the UI content package is customized to the web server-web browser interaction and consequently customized to those applications. *Bederson*, ¶129.

d. [1c]

Hariki in combination with *Harris* discloses this feature. *Bederson* ¶¶130-134. As explained above, *Hariki* serves HTML files from a web server that are received at a web browser. *Hariki*, ¶21. Furthermore, *Hariki* expressly discloses the server “executes” the web server process (*id.*, ¶21) and the mobile device “executes” the application, i.e, web browser (*id.*, ¶38). POSITAs would have readily understood that receiving the subsequent HTML files was due at least in part to executing the web server process and/or the web browser because execution

of the applications is required to operate the server-browser system. *Bederson*, ¶130.

In *Harris*, the generation of the HTML page begins when the “web server accepts a connection and an HTTP request from a mobile device and the servlet engine directs the request to the appropriate page or servlet destined to generate WAX (step 60).” *Harris*, ¶24. Harris further explains that the framework “translates the WAX into a device-specific markup language (WML, HTML, etc.).” *Id.* ¶27. POSITAs would have understood, that just as in *Hariki*, the web server and mobile device are each executing applications that cause the generation of the HTML page (i.e., “compiled content”). *Bederson*, ¶133.

Accordingly, the *Hariki-Harris* system generates HTML (i.e., “compiled content”) by executing web server and mobile device browser applications (i.e., “said application”) and received at a mobile device’s web browser. Therefore, the HTML is received from executing one or both applications. *Bederson*, ¶134.

e. [1d]

Hariki in combination with *Harris* discloses this feature. *Bederson* ¶¶64, 135-139. POSITAs would have also understood that HTML itself is in a syntax that is generic to the mobile device. *Bederson*, ¶135-137. For example, HTML is a standardized language that has the same syntax no matter what device is receiving it. *Id.*, ¶64.

Moreover, the '245 patent explains that the compiled content, i.e., render commands, 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.”

'245*Pat*, 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., 13:26-34.

The '245 patent illustrates exemplary basic commands in a “compiled page description” in Figure 4, reproduced below. The '245 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.”

Id., 15:59-67.

400

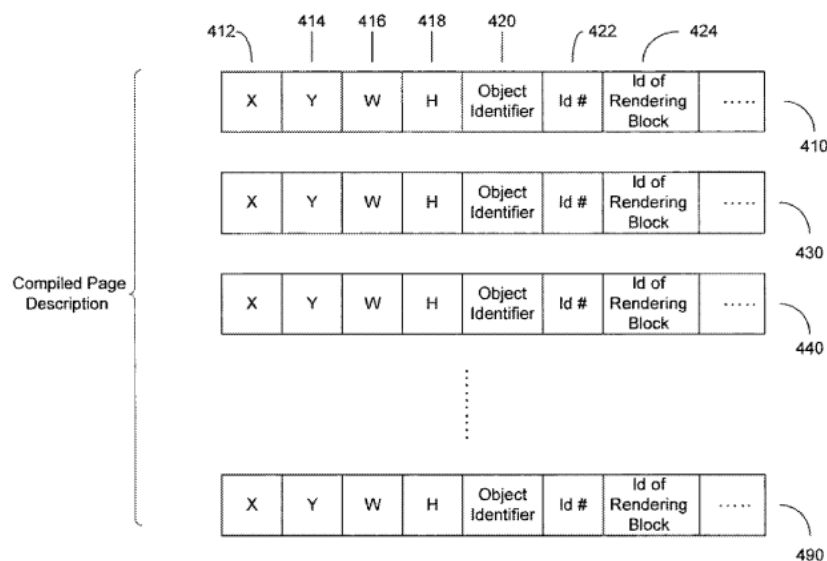


FIGURE 4

'245Pat, FIG. 4

Harris discloses such basic rendering commands. For example, with respect to WAX, the generic high-level syntax used to make the HTML pages, *Harris* provides 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>

```

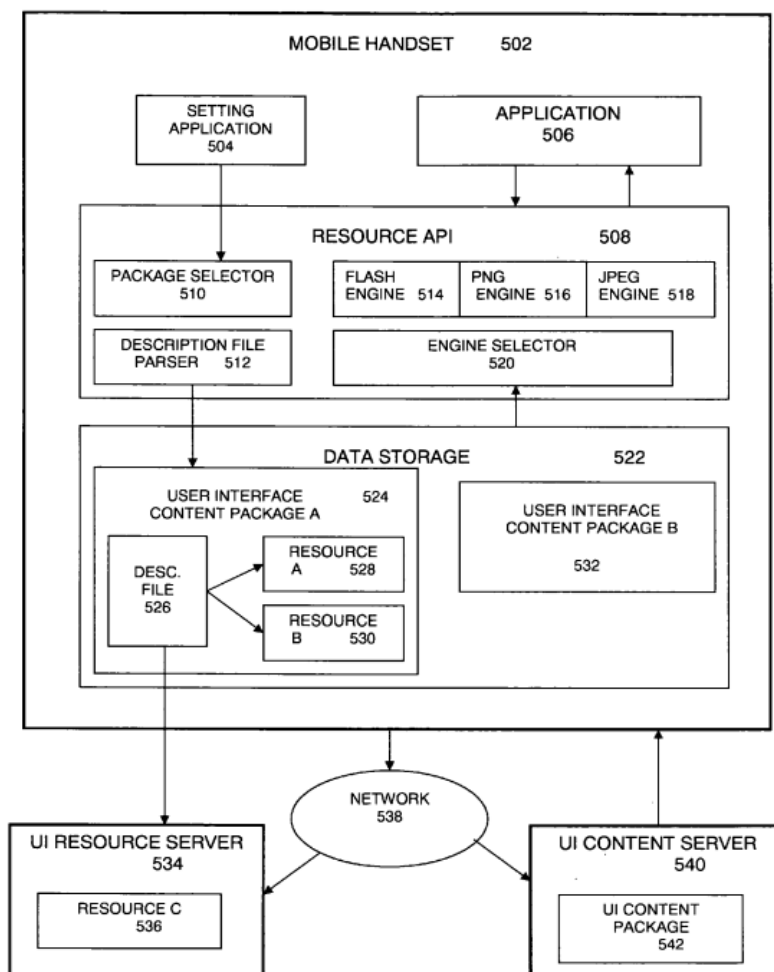
Harris, ¶29

POSITAs would have looked at the above Table and understood that the WAX language used a syntax that was device generic. *Bederson*, ¶138. For example, *Harris* explains that commands such “<wax:button>” are rendered differently in different devices, and text may be displayed in different languages. *Harris*, ¶30; *Bederson*, ¶138. POSITAs would 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*, ¶64. 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., “expressed in a syntax that is generic to said wireless device.”). *Id.*, ¶¶135-139.

f. [1e]

Hariki in combination with *Harris* discloses this feature. *Bederson* ¶¶140-146. The '245 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.” ’245*Pat*, 2:60-64. The '245 patent further states that the graphical user interface rendering blocks “render content and enable a user to interact” with an application, for example by allowing a user to click on a “submit” button. *Id.*, 8:23-26. Regarding Figure 6 of the '245 patent, the '245 patent paraphrases the claim language stating “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 continues: “In one embodiment, content is rendered by processing the compiled content using the reader and by issuing commands from the reader to rendering blocks of the graphical user interface based on the rendering commands of the compiled content.” *Id.*, 8:65-19:2.

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. For example, 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.*, ¶41; *Bederson*, ¶142. The application, e.g., a web browser, as explained above, receives both HTML files (i.e., “compiled content”) and converted resources (i.e., “custom configuration”) in

the UI content package, 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 understood that *Hariki*’s resource API and engines use the converted resources in the UI content 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*, ¶143. POSITAs would have further understood that the “look and feel” environment of the device was a user interface displayed to the user and corresponded to the “graphical user interface,” as claimed. *Bederson*, ¶143.

If PO argues that *Hariki* does not expressly disclose using the engines in the resource API to generate renderable content from the received HTML files, POSITAs would have been motivated to modify *Hariki* to do so. *Bederson*, ¶144-145. For example, *Hariki* expressly contemplates modifying the resource API to

include additional engines for rendering content. *Hariki*, ¶51. Specifically, *Hariki* explains:

elements of the resource API 508 **can be modified or upgraded** without requiring a change to the application program. For example, **different or new versions of engines for different** types of data objects or **programs** can be implemented in the resource API by adding or modifying the engine components.

Id.

Furthermore, *Harris* discloses displaying the content generated for the specific mobile devices. *See, e.g., Harris*, claim 1 (“providing content, said content capable of being displayed to a user of a mobile device interfaced with said network”); *Bederson*, ¶145. *Harris* explains for HTML documents, “an automated testing environment will make sure content will display[] correctly before they are deployed to a device.” *Id.*, ¶35. *Harris* further acknowledges that it was well-known in the art that mobile device displayed content in different ways. *Id.*, ¶¶3-4. POSITAs would have been motivated by the disclosure of *Harris* to modify *Hariki*’s resource API to include additional engines to convert and render HTML files for display on the mobile device. *Id.*, ¶145.

Accordingly, POSITAs would have been motivated to include engines in the resource API to format HTML files and objects, for example, to provide expanded functionality to *Hariki*’s mobile device to render and display content data that *Hariki* contemplates receiving at the mobile device. *Bederson*, ¶146. POSITAs would have had a reasonable expectation of success because engines for rendering

and displaying HTML files were commonplace and well within POSITA's knowledge. *Bederson*, ¶146 (citing *Linburn*, ¶39 and *Talwar*, 12:58-64).¹¹ Thus, modifying *Hariki*'s resource API to include engines to render and display HTML files is nothing more than applying the known technique of including rendering engines in a mobile device ready for improvement to yield predictable results. *Bederson*, ¶146; *KSR Int'l Co. v. Teleflex Inc.*, 550 U.S. 398, 401 (2007).

g. [1f]

Hariki discloses this feature. *Bederson* ¶¶147-157. The '245 patent describes "compiled content" as including "a series of basic commands for rendering the requested application." '245*Pat*, 7:43-45; *see also* 2:46-50 ("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"). As explained above, *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. For example, 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 HTML files include tags that include low level commands to render content on a client device. *See* Section IV.A.3; *Bederson*, ¶148. POSITAs would have understood that HTML files received by a mobile device are rendered into pages in a web browser on the mobile device. *Id.*, ¶148. 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*, ¶148. POSITAs would have understood that such a compilation of commands is how the ’245 patent describes the compiled content. *Id.*, ¶148; citing ’245*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.”) and illustrated in Figure 4.

Furthermore, POSITAs would have understood a client receives embedded images that can be referenced within HTML pages. Additionally, a client receives multiple HTML pages from a server when a client accesses the server (i.e., application) to receive HTML pages. *See* Section IV.A.3; *Bederson*, ¶149. For example, it was common knowledge by 2007 that HTML files included links that would cause subsequent HTML files to be sent to the client after selecting the link. *Id.*, ¶149; *see also* Section IV.A.3. Accordingly, POSITAs would have understood *Hariki* discloses that the mobile device receives embedded images and multiple

(corresponding to the claimed “first” and “second”) HTML files (corresponding to the claimed “compiled content”). *Bederson*, ¶149. Furthermore, POSITAs would have understood that HTML pages received by the mobile device would have been specific to the application (e.g., a web browser) executed on the mobile device because that browser application requested the content in the first place. *Id.*, ¶150.

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 in Section VII.B, *Harris* describes 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 receiving HTML files at a client mobile device where the HTML files are tailored to the particular mobile device. *Bederson*, ¶151.

POSITAs would have understood that the methods disclosed by *Harris* were compatible with and could be adapted and applied to *Hariki* because *Harris*’s platform is the same general architecture as *Hariki*’s server-client system. *Id.*, ¶152. Thus, if PO contends that *Hariki* does not disclose “compiled content,”

POSITAs would have been motivated to combine the server-client system of *Hariki* with the teachings of *Harris* directed to receiving tailored HTML pages at a particular mobile device. *Id.*, ¶152. Specifically, 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 receiving mobile device and looked to *Harris* because, while *Hariki* discloses receiving HTML files at a particular mobile device that are sent from a server, it does not expressly disclose how the HTML files are generated and customized to the particular receiving mobile device. *Id.*, ¶152.

Moreover, POSITAs would have had a reasonable expectation of success in implementing *Harris*' content generation scheme with *Hariki*'s customization scheme. *Id.*, ¶153. For example, while *Harris* discusses that an "interface may be dynamically personalized to the taste of the individual" (*Harris*, ¶7), it 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 in *Harris* 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.”, “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.*, ¶24). *Bederson*, ¶153. Therefore, POSITAs would have been motivated to combine *Hariki* and *Harris* because, while each discloses sending specific content that is received by a mobile device, each describes in detail different aspects of the customization. *Id.*, ¶153.

Harris explains “[c]ontent is generated specifically for each device, both from a display standpoint and a content navigation standpoint.” *Harris*, ¶7. *Harris* explains to do this, a generic markup language called WAX is used, which is “easily extended and can be translated into a variety of different mobile device markup languages.” *Id.*, ¶7. Specifically, *Harris* explains “the framework translates the WAX into a device-specific markup language (WML, HTML, etc.).” *Id.*, ¶27. *Harris* explains 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*, ¶154. 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. POSITAs would have recognized that HTML tags specific to the mobile device (i.e., the link) are present in the HTML files in *Harris*. *Bederson*, ¶154. Accordingly, *Harris* discloses sending basic commands to render display items,

such as a link, that is specific to the application being executed on the mobile device (i.e., “compiled content”).¹² *Bederson*, ¶156. Moreover, HTML is also specific to the web server application on the server because that application is generating the HTML files in the first instance. *Bederson*, ¶156.

Furthermore, *Harris* contemplates serving multiple pages (e.g., HTML files) to the mobile client. For example, *Harris* explains that the mobile device will make “page requests.” *Harris*, ¶27. *Harris* explains that a caching system may be used such that “subsequent requests for the same page do not require processing.” *Id.* This caching is done for multiple pages. *Id.* Accordingly, POSITAs would have understood *Harris* to send at least first and second pages of HTML files which are received by the mobile device, following multiple requests by the mobile device. *Bederson*, ¶157. Furthermore, POSITAs would have understood that the HTML was specific to the pages served by the web server or consumed by the web browser (i.e., “said application”). *Id.*, ¶157.

¹² POSITAs would have understood “commands” as its plain and ordinary meaning, for example, a directive to a computer program to perform a specific task. *Bederson*, ¶155.

h. [1g]

Hariki in combination with *Harris* discloses this feature. *Bederson* ¶¶158-159. For example, *Hariki* discloses that “[i]n general, UI skins 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.” *Hariki*, ¶24. *Hariki* explains, referencing Figure 5, “[t]hrough the resource API functionality, the mobile device is able to change the **UI skin of the application 506** by UI content packages that are downloaded from a UI content server or are pre-installed in the mobile handset, without requiring modification of the application itself.” *Id.*, ¶42. POSITAs would have understood that customizing the UI skin, that is the application program environment or display and sound aspects of the device, using the UI content package would mean that the converted resources (i.e., “custom configuration”) of the UI content package were applicable to all HTML files received from the web server and displayed in application. *Bederson*, ¶¶158-159.

2. Claim 2

Hariki in combination with *Harris* discloses this feature. *Bederson* ¶¶160-163. The ’245 patent describes “reader of said wireless device” as an “engine/reader 114 [that] passes commands to [] rendering blocks to perform the rendering of the application content.” ’245Pat, 8:24-25. Furthermore, the wireless

device “process[es] the compiled content using the reader.” ’245*Pat*, 18:66-67. The reader itself is shown as a generic block 114 in Fig. 1B of the ’245 patent.

100B

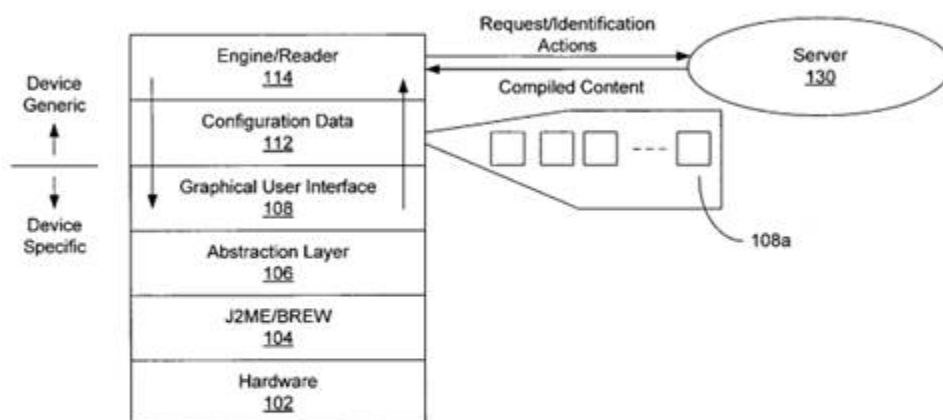


FIGURE 1B

In the *Hariki-Harris* system, described in Sections VIII.A.1.b and VIII.A.1.f, *Harris* discloses a mobile device’s web browser receiving HTML files (i.e., “compiled content”). *Harris*, ¶24. Furthermore, as shown in *Hariki*’s Figure 5 below, the mobile device’s “resource API contains a number of different plug-in content engines and an engine selector component to allow an application program executed on the mobile handset to access the resources.” *Hariki*, ¶19.

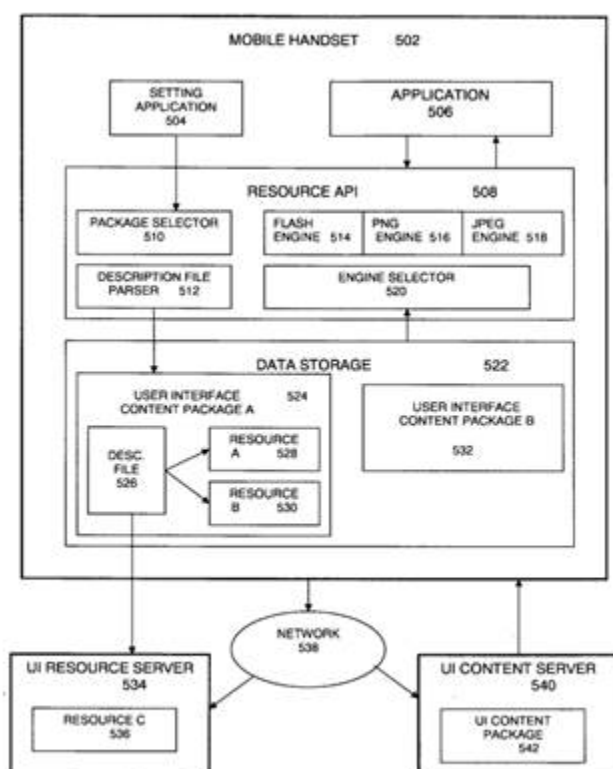


FIGURE 5

The resources “comprise 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. The resource API then “uploads all referenced resources to the application.” *Id.*, ¶51. *Hariki*’s engines communicate with the files, links to files, and/or data or program objects. POSITAs would have understood that these files or objects can be *Harris*’ HTML-based rendering blocks such as WAX buttons. *Harris*, ¶19. Accordingly, POSITAs would have understood that *Hariki*’s engines (i.e., the “reader”) communicate with (i.e., “issuing commands to”) the files, links to files, and/or data or program objects such as buttons (i.e., “rendering blocks”)

based on HTML files (corresponds to “compiled content comprising render commands”¹³).

3. Claim 3

Hariki in combination with *Harris* discloses this feature. *Bederson* ¶¶164-165. For example, regarding the UI content package, *Hariki* discloses that it “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 **image files, sound files**, screen layouts, icons, movies, and so on.” *Id.*, ¶31. Accordingly, the *Hariki*-

¹³ Claim 2 of the ’245 patent claims “**said** rendering commands”. ’245*Pat*, 21:5.

However, the limitation “rendering commands” is not claimed earlier in the claims. Therefore, “rendering commands” in Claim 2 lacks antecedent basis under 35 U.S.C. §112(b). As best understood, “rendering commands” refer to the compiled content render commands of the ’245 patent’s claim 1. If “rendering commands” refer to any other commands, then Claims 2, 13, and 24 are indefinite under 35 U.S.C. §112(b).

Harris system discloses “renderable content comprises audio content and display content.” *Bederson*, ¶¶164-165.

4. Claim 4

Hariki in combination with *Harris* discloses this feature. *Bederson* ¶¶166-167. As explained in Sections VIII.A.1.c and VIII.A.1.f, HTML files (or pages) are sent from a remotely executed web server application when requested by a web browser application executed on a mobile device. *Bederson*, ¶166. As also explained above, the UI skin content package is “associated” with the web server application (*see* Section VIII.A.1.c) and the HTML files are specific to the web server application (*see* Section VIII.A.1.f). *Bederson*, ¶166.

Because the web browser must request the HTML files (i.e., “compiled content”) from the web server, POSITAs would have understood that the HTML files are generated as a result of both the web browser and web server. Furthermore, *Hariki* discloses that the web server is executed on server 102 and the web browser is executed on client 108 or 109, where the server and client are separated by network 110. *Hariki*, ¶27. Accordingly, the HTML files (i.e., “compiled content”) are “partially resultant” from the web server application running on a server that is not part of the client (i.e., “said application operating on a remote server”). *Bederson*, ¶167.

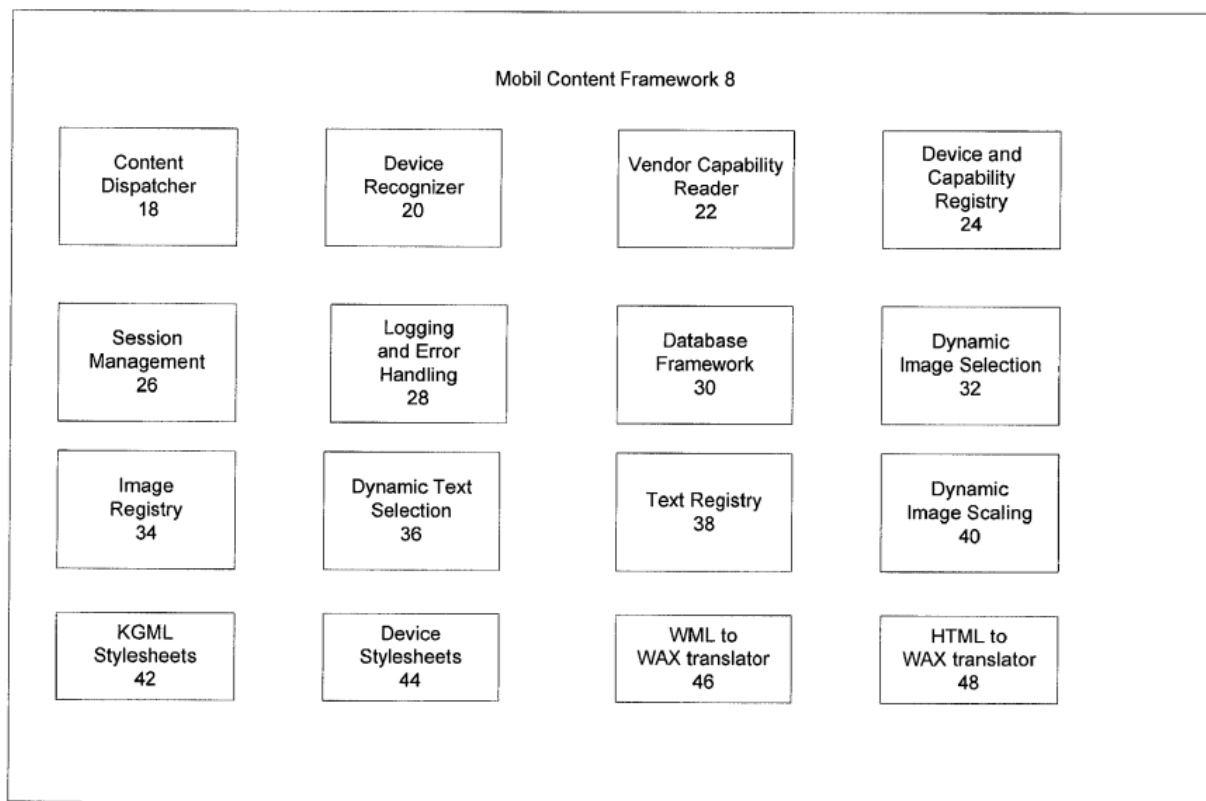
5. Claim 5¹⁴

Hariki in combination with *Harris* discloses this feature. *Bederson* ¶¶168-170. For example, in the *Hariki-Harris* system, *Harris* discloses a generic WAX template “is easily extended and can be translated into a variety of different mobile device markup languages.” *Harris*, ¶7. *Harris* explains 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.* For example, 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

¹⁴ To the extent that this claim is not indefinite due to lack of antecedent basis of “renderable content,” *Hariki* in combination with *Harris* discloses this feature. Petitioner reserves all rights to raise §112 arguments in any litigation involving the ’245 patent.

provides “attributes such as screen size, color depth, browser version and type, and translation rules” based on device type and application. *Id.*; *Bederson*, ¶169.

Figure 2

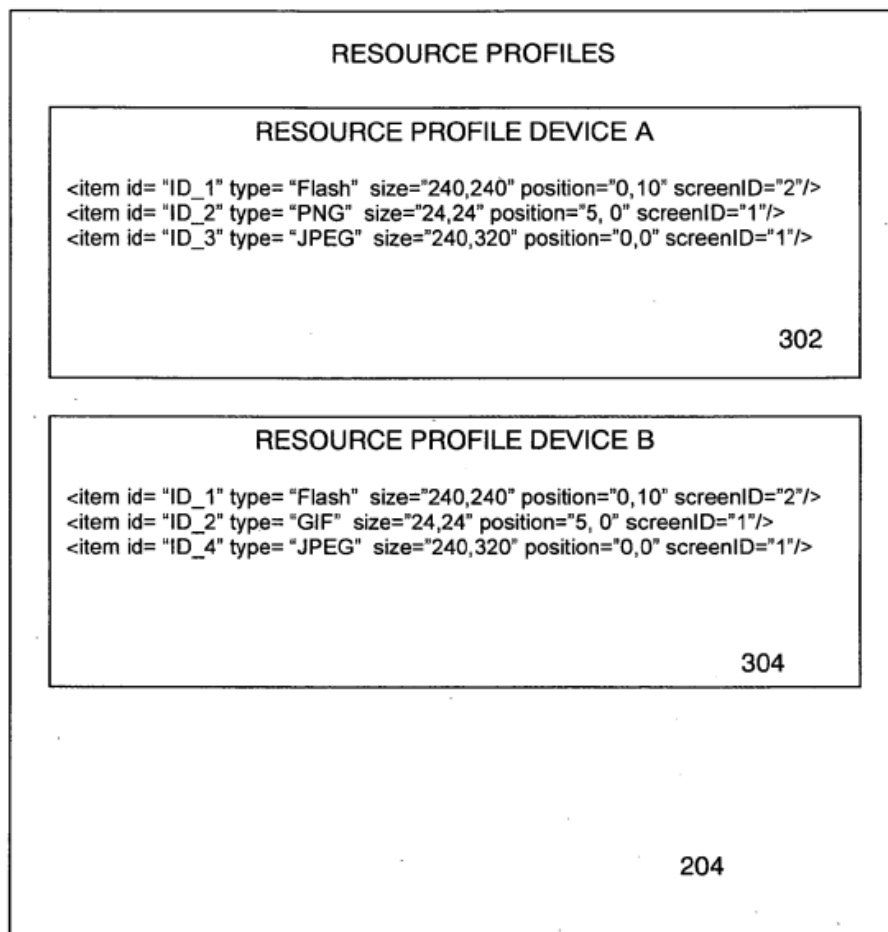
*Harris*, FIG. 2

Accordingly, the *Hariki-Harris* system tailors the HTML pages to the specific attributes (i.e., “rendering capabilities”) of the mobile device. *Bederson*, ¶170.

6. Claim 6

Hariki in combination with *Harris* discloses this feature. *Bederson* ¶¶171-176. As explained in Section VIII.A.1.b, *Hariki* explains that “UI skin package

data **specific to each mobile device** is generated from common resource data using **resource profile information for each mobile device model.**” *Hariki*, ¶29. *Hariki*’s Figure 3, below, shows an example resource profile used to generate the UI content package. *Id.*, ¶35. POSITAs would have recognized that the resource profiles include specific rendering capabilities and attributes for a specific mobile device, such as size and position. *Id.*, ¶35; *Bederson*, ¶172. As shown below, these attributes determine how the resource, for instance “JPEG” as shown for “item id= ‘ID_3’,” is to be rendered on the specific mobile device. *Id.*, ¶32; *Bederson*, ¶172. The user interface authoring tool, as described in Sections VIII.A.1.b and VIII.A.1.c, uses the resource profiles to generate a UI content package with specific resources and a descriptor file that instructs the mobile device on how to render the converted resources of the UI content package for the backgrounds, title bars, buttons, and alert sounds (i.e., “rendering blocks”) of the mobile device. *Id.*, ¶¶33, 45-46; *Bederson*, ¶172. Accordingly, the *Hariki-Harris* system discloses the claimed “rendering blocks operate[] specific to a wireless device type of said wireless device.” *Bederson*, ¶172.



Hariki, FIG. 3

Hariki's Figure 6, below, shows the UI content package sent to the mobile device. *Hariki*, ¶48. The package includes a "description file contains a number of item entries 601. Each item entry consists of certain items of information, such as identifier (ID), path (location), and file type, among other parameters, relating to a particular resource." *Id.* POSITAs would have recognized that the type of resource, e.g., Flash, PNG, JPEG, is not specific to any particular mobile device (e.g., device independent), although the resource itself is included based on the type of mobile

device. *Bederson*, ¶174. It was well-known toPOSITAs before August 2007 that a Flash, PNG, or JPEG file contained commands that could be rendered on many different device types. *Id.*, ¶174. The format of commands in these files (i.e., “syntax”) was thus device independent. *Id.*, ¶174 (explaining that “syntax” is merely the way in with code is written).

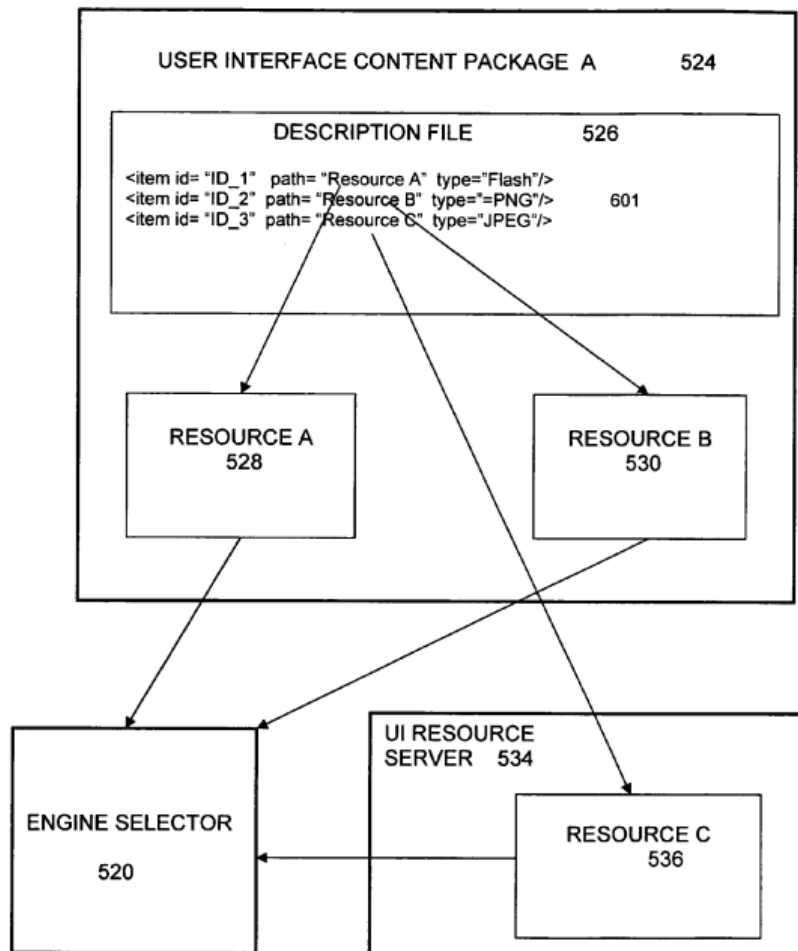


FIGURE 6

Hariki, FIG. 6

As explained in Section VIII.A.1.c, the converted resources in the UI content package (the UI skin), determine the configuration of the mobile device when used by the engines of the resource API. *Bederson*, ¶175. Consequently, POSITAs would have understood the converted resources in the UI content package instruct the mobile device how to display the contents of the converted resource. *Id.*, ¶175.

Accordingly, the resource API engines converting content use the device independent command structure of the converted resources (corresponding to the claimed “syntax that is generic to said wireless device type”) to display content on backgrounds, title bars, buttons, and alert sounds (i.e., “rendering blocks”) on the mobile device. *Bederson*, ¶176.

7. Claim 7

Hariki in combination with *Harris* discloses this feature. *Bederson* ¶177. As explained in Section VIII.A.5, the converted resources in the UI content package use a device independent command structure. Accordingly, the converted resources (i.e., “custom configuration”) “comprises a syntax that is generic regarding said wireless device type.”

8. Claim 8

Hariki in combination with *Harris* discloses this feature. *Bederson* ¶¶177-180. As explained in Section VIII.A.1.b, the “UI skin consisting of a **particular screen configuration**, size, color scheme, font, menu scheme, keypad button

assignment, ringtone selection” may be included with the mobile device. *Id.*, ¶26. *Hariki*, explains “[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 also provide the customized UI to the mobile device. *Id.* As explained above in Section VIII.A.1.h, the UI content package includes a converted resource such as a Flash, PNG, or JPEG 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. *Hariki* discloses that “custom configuration comprises configuration information...specific to said application” *Bederson*, ¶178. Additionally, as explained in Section VIII.A.1.c, *Hariki* explains “the resources referenced by the descriptor file and **utilized by the application** must be referenced in some way by the application and **then retrieved and converted to an appropriate format for use by the application.**” *Id.*, ¶39. *Hariki* further discloses “custom configuration comprises... content specific to said application.” *Bederson*, ¶179. Therefore, *Hariki* discloses the converted resource file (i.e., “custom configuration”) includes information such as size, color, position, font, menu compatible with the application (i.e., “configuration

information...specific to said application”) and resources that are formatted and utilized based on a specific application (i.e., “content specific to said application”).

9. Claim 9

Hariki in combination with *Harris* discloses this feature. *Bederson* ¶¶181-184. *Hariki* discloses “[t]he mobile device handset includes a resource API to **download...UI content that is locally stored.**” *Hariki*, ¶19. *Hariki* discloses “[t]he mobile handset 502 also includes a data storage component... **One or more user interface content packages...[are] stored in this data storage.**” *Hariki*, ¶45. *Hariki* discloses a “**UI content package comprises the appropriate converted resources.**” *Id.*, ¶33.

Additionally, as explained in Section VIII.A.1.a-VIII.A.1.b, *Hariki* discloses a description file received by the mobile device identifies resources with “certain items of information, such as identifier (ID)...relating to a particular resource.” *Id.*, ¶48. The mobile device then accesses the resource specified by the resource ID. *Id.*, ¶50.

POSITAs would have understood the ID in the server-transmitted description file identifies converted resources in the mobile device’s locally stored UI content package, which are then used by the mobile device to render content. *Bederson*, ¶183.

Accordingly, *Hariki* discloses converted resources (i.e., “custom configuration”) are stored locally in the mobile device (i.e., “one of a plurality of...wireless device”) and a server-transmitted ID (i.e., “an identifier that identifies said custom configuration”) is used to identify a specific converted resource.

10. Claim 10

The elements of this claim are related to those in claims [1a], [1b], and [9]. *Bederson* ¶¶185-186. For example, as explained in Section VIII.A.1.a, *Hariki* discloses a mobile device receives 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 correspond to the claimed “custom configuration” and “identification of [the] custom configuration,” respectively that are received by the mobile device. *Bederson*, ¶186. As explained in Section VIII.A.9, *Hariki* discloses that the mobile device includes a memory for storing received information including converted resource files. Accordingly, *Hariki* in combination with *Harris* discloses this feature as discussed in Sections VIII.A.1.b-VIII.A.1.c and VIII.A.9. *Bederson*, ¶186.

11. Claim 11

Hariki discloses this feature. *Bederson* ¶¶187-189. *Hariki* explains the UI skin consists “of a particular screen configuration, size, color scheme, font, menu

scheme, keypad button assignment, ringtone selection.” *Id.*, ¶26. *Hariki* explains “[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 the UI skin can configure “backgrounds, title bars, buttons, alert sounds, **and so on.**” *Id.*, ¶24.

As explained in Section VIII.A.1.b, POSITAs would have understood that these backgrounds, title bars, buttons, and alert sounds comprise different types of “rendering blocks” within the context of the ’245 patent.

Furthermore, POSITAs would have understood that *Hariki*’s multiple rendering blocks including keypad buttons (i.e., “button images”); ringtones and alert sounds (i.e., “rendering of audio”); and title bars (i.e., “ticker information”) cover many different customizable aspects of a mobile device, and fall squarely within the ’245 patent’s examples of displaying different types of rendering blocks on the mobile device.” *Bederson*, ¶189.

12. Claim 12

a. [12pre]

To the extent the preamble is limiting, *Hariki* discloses this feature. *Bederson* ¶¶190-192. This element is substantively identical to claim [1pre]. *Compare* [12pre] (“[a] non-transitory computer readable medium implement **a method of rendering content on a wireless device**”) *with* [1pre] (“[a] **method of**

rendering content on a wireless device”). Furthermore, *Hariki* discloses that “the various functions disclosed herein may be described using **instructions embodied in various machine-readable or computer-readable media.**” *Hariki*, ¶54. *Harris* likewise discloses “medium holding computer-executable steps for a method.” *Harris*, claim 21. Thus, POSITAs would have understood that any methods disclosed in *Hariki* and *Harris* would have been implemented on a “non-transitory computer readable medium.” *Bederson*, ¶191.

Accordingly, *Hariki* discloses this feature for substantially the reasons discussed in Section VIII.A.1.a. *Bederson*, ¶192.

b. [12a] and [12b]

Each of these elements is substantively identical to claim [1a] and [1b], respectively.

Accordingly, *Hariki* discloses these features for the reasons discussed in Sections VIII.A.1.b-VIII.A.1.c. *Bederson*, ¶¶193-194.

c. [12c], [12d], [12e], [12f]

Each of these elements is substantively identical to claim [1c], [1d], [1e], [1f], respectively.

Accordingly, *Hariki* discloses these features for the reasons discussed in Sections VIII.A.1.d-VIII.A.1.g. *Bederson*, ¶¶195-196.

d. [12g]

This element is substantively identical to claim [1g].

Accordingly, *Hariki* discloses this feature for the reasons discussed in Section VIII.A.1.h. *Bederson*, ¶¶197-198.

13. Claim 13

Claim 13 is substantively identical to claim 2. Accordingly, *Hariki* in combination with *Harris* discloses this feature for the reasons discussed in Section VIII.A.2. *Bederson*, ¶199.

14. Claim 14

Claim 14 is substantively identical to claim 3. Accordingly, *Hariki* in combination with *Harris* discloses this feature for the reasons discussed in Section VIII.A.3. *Bederson*, ¶200.

15. Claim 15

Claim 15 is substantively identical to claim 4. Accordingly, *Hariki* in combination with *Harris* discloses this feature for the same reasons discussed in Section VIII.A.4. *Bederson*, ¶201.

16. Claim 16¹⁵

Claim 16 is substantively identical to claim 5. Accordingly, *Hariki* in combination with *Harris* discloses this feature for the reasons discussed in Section VIII.A.5. *Bederson*, ¶202.

17. Claim 17

Claim 17 is substantively identical to claim 6. Accordingly, *Hariki* in combination with *Harris* discloses this feature for the reasons discussed in Section VIII.A.6. *Bederson*, ¶203.

18. Claim 18

Claim 18 is substantively identical to claim 7. Accordingly, *Hariki* in combination with *Harris* discloses this feature for the reasons discussed in Section VIII.A.7. *Bederson*, ¶204.

19. Claim 19

Claim 19 is substantively identical to claim 8. Accordingly, *Hariki* in combination with *Harris* discloses this feature for the reasons discussed in Section VIII.A.8. *Bederson*, ¶205.

¹⁵ See n.14.

20. Claim 20

Claim 20 is substantively identical to claim 9. Accordingly, *Hariki* in combination with *Harris* discloses this feature for the reasons discussed in Section VIII.A.9. *Bederson*, ¶206.

21. Claim 21

Claim 21 is substantively identical to claim 10. Accordingly, *Hariki* in combination with *Harris* discloses this feature for the same reasons discussed in Section VIII.A.10. *Bederson*, ¶207.

22. Claim 22

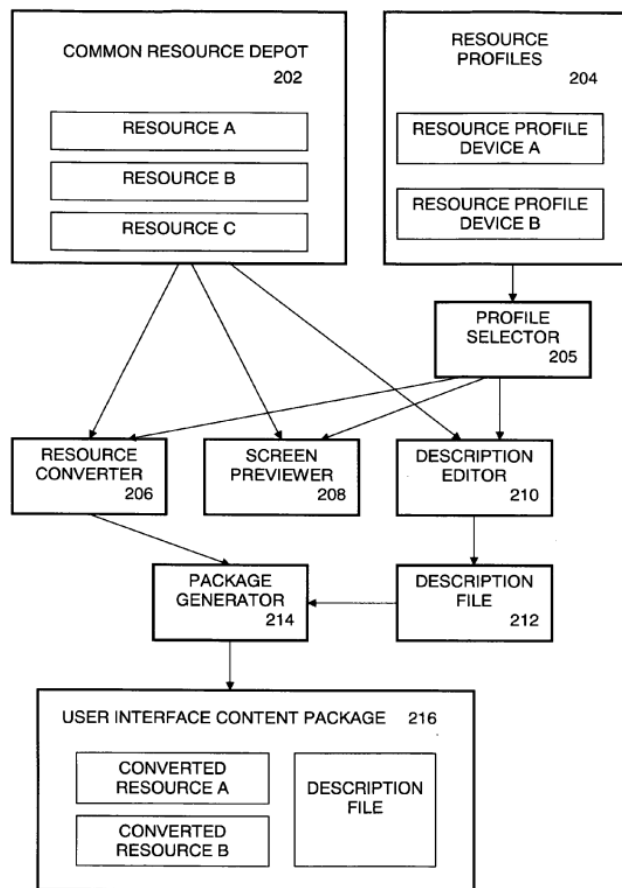
Claim 22 is substantively identical to claim 11. Accordingly, *Hariki* in combination with *Harris* discloses this feature for the reasons discussed in Section VIII.A.11. *Bederson*, ¶208.

23. Claim 23**a. [23pre]**

To the extent the preamble is limiting, *Hariki* in combination with *Harris* discloses this feature. *Bederson* ¶¶209-211. This element is substantively identical to claim [1pre]. Accordingly, *Hariki* discloses this feature for the reasons discussed in Section VIII.A.1.a.

b. [23a]

Hariki in combination with *Harris* discloses this feature for the reasons discussed in Sections VIII.A.1.a.-VIII.A.1.b. *Bederson* ¶¶212-217.

**FIGURE 2***Hariki*, FIG. 2

As discussed in VIII.A.1.a., referring to Fig. 2, *Hariki* discloses that “UI skin package data specific to each mobile device is generated from common resource data.” *Id.*, ¶29. *Hariki* explains that “[t]he UI content package comprises the appropriate converted resources and the description file.” *Id.*, ¶33.

Additionally, as discussed in VIII.A.1.b., *Hariki* explains that “the resources referenced by the descriptor file and **utilized by the application** must be

referenced in some way by the application and then retrieved and converted to an appropriate format **for use by the application.**” *Id.*, ¶39. The application may be “a web browser program to access the web pages served by server computer” *Id.*, ¶21. Accordingly, POSITAs would have understood that the converted resources in the UI package (i.e., “custom configuration”) are associated with the web browser (i.e., “an application”). *Bederson*, ¶¶214-215.

Accordingly, *Hariki* discloses a mobile device receiving (corresponding the claimed “receiving” at “wireless device”) a UI package including converted resources (i.e., the claimed “custom configuration”) that is utilized by an application such as a web browser (corresponding to “associated with an application”) from a server.

The only difference between [23a] and [1a] and [1b] is that [23a] recites “a transceiver coupled to a bus” in the mobile device. *Bederson*, ¶216. The *Hariki-Harris* system discloses this feature also. *Id.*, ¶216.

The ’245 patent discloses that “[t]he wireless device 300 further includes a transceiver ...[that] receive[s] a series of basic commands from a remote server that may be used to render application and/or content on the display.” ’245*Pat.*, 15:38-42. The ’245 patent’s Fig. 3 does not provide any additional details on the transceiver.

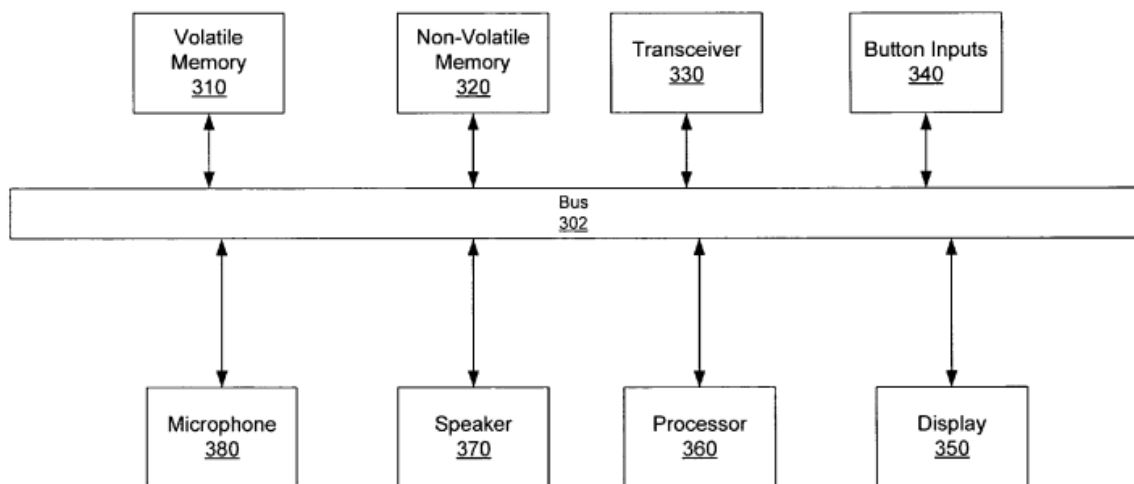


FIGURE 3

POSITAs would understand that *Hariki's* mobile devices would include a similar generic transceiver and bus for communicating with the server. Furthermore, it was well-known at the time of invention that mobile phones have transceivers and buses. For example, *Goodacre* (EX1022), shows in Fig. 1A a telephonic device 100 including an antenna 109 (corresponding to “transceiver”) and a bus 110.

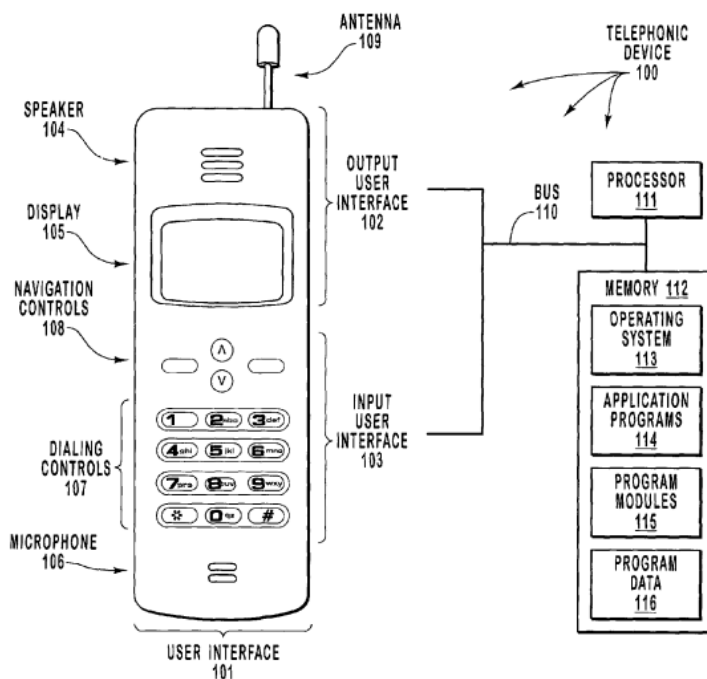


FIG. 1A

c. [23b]

Each of these elements is substantively identical to that in claim [1c].

Accordingly, *Hariki* in combination with *Harris* discloses these features for the reasons discussed in Section VIII.A.1.d. *Bederson*, ¶220.

c. [23c]

Each of these elements is substantively identical to claim [1d].

Accordingly, *Hariki* in combination with *Harris* discloses these features for the reasons discussed in Section VIII.A.1.e. *Bederson*, ¶221.

d. [23d]

Each of these elements is substantively nearly identical to claim 9.

Accordingly, *Hariki* in combination with *Harris* discloses these features for the reasons discussed in Section VIII.A.9. *Bederson*, ¶¶222-226.

The only difference between [23d] and [9] is that [23d] claims a “memory” in the mobile device. *Bederson*, ¶223. The *Hariki-Harris* system also discloses this feature. *Id.*, ¶¶223-224.

For example, *Hariki* discloses “[t]he mobile handset 502 also includes a data storage component 522, which can be implemented as any type of fixed, removable or on-board memory device.” *Hariki*, ¶45.

POSITAs would understand that any data and commands received and processed by the mobile device would have to be implemented through *Hariki*’s memory.

e. [23e]

Each of these elements is substantively nearly identical to claim [1e].

Accordingly, *Hariki* in combination with *Harris* discloses these features for the reasons discussed in Section VIII.1.f. *Bederson*, ¶¶227-228.

The only difference between [23e] and [1e] is that [23e] specifically claims a “processor” in the mobile device. *Bederson*, ¶227. The *Hariki-Harris* system also discloses this feature. *Id.*, ¶¶227-228.

For example, *Hariki* explains that “[a]spects of the UI authoring and mobile handset system described herein may be implemented as functionality programmed

into any of a variety of circuitry, including...**microcontrollers with memory (such as EEPROM), embedded microprocessors**” *Hariki*, ¶53.

f. [23f]

Each of these elements is substantively identical to claim [1b].

Accordingly, *Hariki* in combination with *Harris* discloses these features for the reasons discussed in Section VIII.A.1.c. *Bederson*, ¶229.

g. [23g]

Each of these elements is substantively nearly identical to that in claims [1e] and 2.

Accordingly, *Hariki* in combination with *Harris* discloses these features for the reasons discussed in Section VIII.A.1.b *Bederson*, ¶¶230-231.

The only difference between [23g] and [1e] is that [23g] claims an “engine for reading” in the mobile device. *Bederson*, ¶231. The *Hariki-Harris* system discloses this feature as explained in Section VIII.A.2. *Id.*, ¶231.

h. [23h]

Each of these elements is substantively nearly identical to claims [1f] and [1g].

Accordingly, *Hariki* in combination with *Harris* discloses these features for the reasons discussed in Sections VIII.A.1.g-VIII.A.1.h. *Bederson*, ¶¶232-235.

The only difference between [23h] and [1f] and [1g] is that [23f] recites “a display device coupled to said bus” in the mobile device. *Bederson*, ¶233. The *Hariki-Harris* system also discloses this feature. *Id.*, ¶¶233, 235.

Hariki discloses “certain mobile phone service and equipment providers provide user interface (UI) customization capabilities that allow users to **personalize their phones or mobile devices with...displays (wallpaper)**, menu configurations, and the like.” *Hariki*, ¶ 5. POSITAs would understand that *Hariki*’s mobile devices would include a display for displaying backgrounds and menu configurations on a mobile device. Furthermore, it was well-known at the time of invention that mobile phones have displays connected to buses. For example, *Goodacre* (EX1022), shows in Fig. 1A a telephonic device 100 including a display 105 (corresponding to “a display device”) connected to a processor 111 thorough a bus 110.

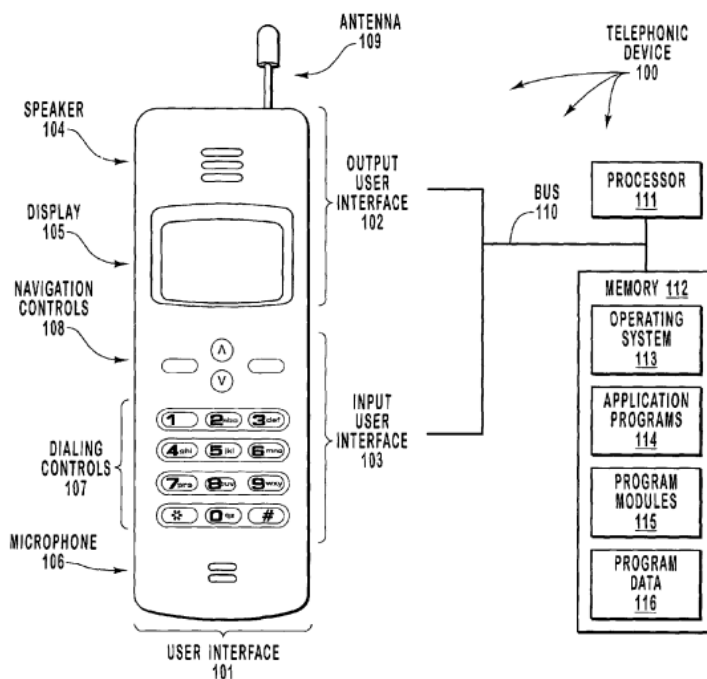


FIG. 1A

24. Claim 24

Claim 24 is substantively identical to claim 2. Accordingly, *Hariki* in combination with *Harris* discloses these features for the reasons discussed in Section VIII.A.2. *Bederson*, ¶236.

25. Claim 25

Claim 25 is substantively nearly identical to claim 3. Accordingly, *Hariki* in combination with *Harris* discloses these features for the reasons discussed in Section VIII.A.3. *Bederson*, ¶¶237-240.

The only difference between [25] and [3] is that [25] recites “an audio rendering device” in the mobile device. *Bederson*, ¶238. The *Hariki-Harris* system also discloses this feature. *Id.*, ¶240.

Hariki discloses that downloaded UI information comprises “ video clips, music or other sound clips, ringtones, ...all of which are referred to as “UI content objects”.” *Hariki*, ¶27. POSITAs would understand that *Hariki*’s mobile devices would include an audio rendering device to play the video clips, ringtones, music, and sound clips. Furthermore, it was well-known at the time of invention that mobile phones include speakers. For example, *Goodacre* (EX1022), published over three years prior to the filing of the ’245 patent, shows in Fig. 1A a telephonic device 100, which is similar to *Hariki*’s mobile device, including a speaker 104 (corresponding to “audio rendering device”).

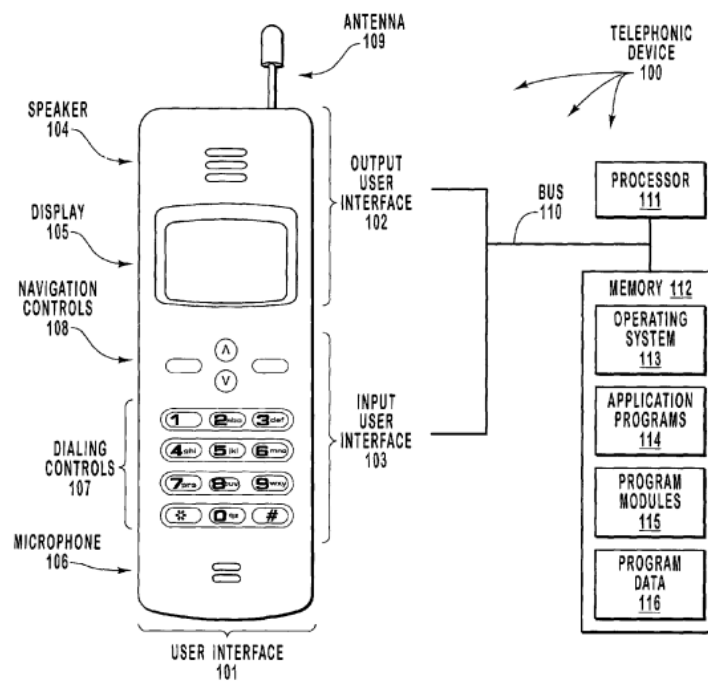


FIG. 1A

26. Claim 26

Claim 26 is substantively identical to claim 4. Accordingly, *Hariki* in combination with *Harris* discloses these features for the reasons discussed in Section VIII.A.4. *Bederson*, ¶241.

27. Claim 27¹⁶

Claim 27 is substantively identical to claim 5. Accordingly, *Hariki* in combination with *Harris* discloses these features for the reasons discussed in Section VIII.A.5. *Bederson*, ¶242.

28. Claim 28

Claim 28 is substantively identical to claim 6. Accordingly, *Hariki* in combination with *Harris* discloses these features for the reasons discussed in Section VIII.A.6. *Bederson*, ¶243.

29. Claim 29

Claim 29 is substantively identical to claim 7. Accordingly, *Hariki* in combination with *Harris* discloses these features for the reasons discussed in Section VIII.A.7. *Bederson*, ¶244.

30. Claim 30

Claim 30 is substantively identical to claim 8. Accordingly, *Hariki* in combination with *Harris* discloses these features for the reasons discussed in Section VIII.A.8. *Bederson*, ¶245.

¹⁶ See n.14.

31. Claim 31

Claim 31 is substantively identical to claim 9. Accordingly, *Hariki* in combination with *Harris* discloses these features for the reasons discussed in Section VIII.A.9. *Bederson*, ¶246.

32. Claim 32

Claim 32 is substantively identical to claim 11. Accordingly, *Hariki* in combination with *Harris* discloses these features for the reasons discussed in Section VIII.A.11. *Bederson*, ¶247.

33. Claim 33

Hariki discloses this feature. *Bederson* ¶¶248-250.

As explained in section VIII.A.1.b, POSITAs would have understood that backgrounds, title bars, buttons, and alert sounds comprise different types of “rendering blocks” within the context of the ’245 patent.

Furthermore, POSITAs would have understood that *Hariki*’s rendering blocks including title bars and display of menus (i.e., “scroll bar”); screen configuration and size (i.e., “input box on said display screen); and screen color scheme, font (i.e., “text display region on said display screen”) cover many different customizable aspects of a mobile device display, and fall squarely within the ’245 patent’s examples of displaying different types of rendering blocks on the mobile device.” *Bederson*, ¶248-250.

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*, ¶252.

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 Denials¹⁷ (“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

¹⁷ Available at https://www.uspto.gov/sites/default/files/documents/interim_proc_discretionary_denials_aia_parallel_district_court_litigation_memo_20220621_.pdf

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,¹⁸ 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 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

¹⁸ *See GoTV Streaming, LLC v. Netflix Inc.*, 2:22-CV-07556-RGK-SHK (C.D. Cal.).

district court and identifies the report providing those statistics.¹⁹ 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.²⁰ Here, the complaint in the Netflix litigation was filed on October 17, 2022, making the calculated trial date in September 2024. *See* **EXError! Reference source not found..** Although the trial date in the Netflix litigation is currently October 17, 2023, that date could get pushed back. **EXError! Reference source not found..** 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 ’245 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.

¹⁹ Available at <https://www.uscourts.gov/statistics/table/na/federal-court-managementstatistics/2022/03/31-1>.

²⁰ *See* https://www.uscourts.gov/sites/default/files/data_tables/fcms_na_distprofile1231.2022.pdf.

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 twelve claims in the district court, while this Petition challenges all twenty 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 '245 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 '245 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 '245 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..

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 1-33 of the '245 patent as unpatentable.

Respectfully submitted,

Dated: April 7, 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 13,944 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 7, 2023, I caused a true and correct copy of the foregoing materials:

- Petition for *Inter Partes* Review of U.S. Patent No. 8,478,245
- Exhibits 1001-1023
- 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 8,478,245 as follows:

Ali Alemozafar
WILSON SONSINI GOODRICH & ROSATI
650 Page Mill Road
Palo Alto, CA 94304-1050
aalemozafar@wsgr.com

Aliza George Carrano
Lead Counsel for Petitioner

