

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
PATENT 8,989,715

PATENT OWNER'S RESPONSE
37 C.F.R. § 42.120

TABLE OF CONTENTS

TABLE OF AUTHORITIES	v
CURRENT EXHIBIT LIST	vii
CLAIM LISTING	xii
I. INTRODUCTION.....	1
II. OVERVIEW OF THE PATENT-AT-ISSUE.....	2
A. Subject Matter.....	2
B. Invention as Illustrated through Embodiments.....	4
1. Applications.....	4
2. Compiled Content and Rendering Blocks.....	6
3. Custom Configuration	9
4. Client	9
III. OVERVIEW OF THE CITED REFERENCES	10
A. Hariki	10
B. Harris.....	13
IV. LEVEL OF ORDINARY SKILL IN THE ART	14
V. CLAIM CONSTRUCTION	15
A. District Court Constructions	15
B. “Customized to Said Application” Means “Created For” the Application Program.....	16
<i>1[b]/9[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</i>	<i>16</i>
<i>17[b] . . . custom configuration that configures a plurality of rendering blocks of said wireless device to render content in a manner customized to an application . . .</i>	<i>16</i>

C. All Claims Require the Recited “Application” to Run Remotely	18
<i>1[c]/9[c]/17[c] [transmitting, to said wireless device, / . . . transmits compiled content to said wireless device, said] compiled content comprising (i) first compiled content specific to a first page of said application and (ii) second compiled content specific to a second page of said application</i>	18
<i>1[d]/9[d]/17[d] wherein said compiled content is generated in part from execution of said application [by said server]</i>	18
1. The Claims Recite a Remote Application Transmitting Compiled Content Pages to a Separate Client.....	18
2. The Specification Exclusively Promotes a Remote Application Executing on a Server Such that There Is No Written Description Support for Claiming a Locally Running Application	19
3. All Parties Agree that the Claims Require the Application Be Executing Remotely.....	21
4. A POSITA Would Understand the Claimed Application is Remote from the Wireless Device	27
D. The Pages of Compiled Content Must Be in Part Generated by the Executing Remote Application.....	28
<i>1[d]/9[d]/17[d] wherein said compiled content is generated in part from execution of said application [by said server]</i>	28
E. “Layout Solver” Is Not a Means-Plus-Function Term	33
VI. HARIKI IN VIEW OF HARRIS FAILS TO RENDER ANY CLAIM OBVIOUS	35
A. No <i>Single</i> Application Petitioner Identifies Renders any Independent Claim Obvious	35
1. The UI Skin (Identified as the “Custom Configuration”) Does Not Configure a Plurality of Rendering Blocks to	

Render Content in a Manner Customized to a Web Server (Identified as the “Application”).....	36
<i>1[b]/9[b] wherein said custom configuration . . . configures said plurality of rendering blocks to render content in a manner customized to said application.....</i>	<i>36</i>
<i>17[b] . . . custom configuration that configures a plurality of rendering blocks of said wireless device to render content in a manner customized to an application . . .</i>	<i>36</i>
2. The Web Browser Does Not Transmit HTML Files to a Wireless Device.....	41
<i>1[c]/9[c]/17[c] [transmitting, to said wireless device, / . . . transmits compiled content to said wireless device, said] compiled content comprising (i) first compiled content specific to a first page of said application and (ii) second compiled content specific to a second page of said application.....</i>	<i>41</i>
3. The Web Browser Is Not Running Remotely from the Wireless Device.....	43
<i>1[c]/9[c]/17[c] [transmitting, to said wireless device, / . . . transmits compiled content to said wireless device, said] compiled content comprising (i) first compiled content specific to a first page of said application and (ii) second compiled content specific to a second page of said application.....</i>	<i>43</i>
<i>1[d]/9[d]/17[d] wherein said compiled content is generated in part from execution of said application [by said server].....</i>	<i>43</i>
4. The HTML Files are Not Generated in Part from the Execution of the Web Browser	44
<i>1[d]/9[d]/17[d] wherein said compiled content is generated in part from execution of said application [by said server].....</i>	<i>44</i>
B. Hariki’s UI Skin Is Not “Applicable” to the Received HTML Files.....	49

<i>1[f]/9[f]/17[f]</i> wherein said custom configuration is applicable to said first and second compiled content	49
C. Hariki in View of Harris Additionally Does Not Teach the “Library” and “Layout Solver” Limitations of Claims 17–20	52
<i>17[a]</i> a library of applications;	52
<i>17[b]</i> a library of custom configuration data comprising a custom configuration that configures a plurality of rendering blocks of said wireless device to render content in a manner customized to an application from said library of applications requested by said wireless device	52
D. Hariki in View of Harris Does Not Teach the “Layout Solver” Limitation.....	54
<i>17[c]</i> a layout solver that transmits compiled content to said wireless device, said compiled content comprising (i) first compiled content specific to a first page of said application and (ii) second compiled content specific to a second page of said application	54
E. Harris and Hariki Further Do not Teach Limitations in Dependent Claim 3 and 11	57
<i>3/11 [A method as described in claim 1/A non-transitory computer readable medium as described in claim 9] wherein said compiled content is partially resultant from said application operating on a remote server.....</i>	57
VII. OBJECTIVE INDICIA OF NONOBVIOUSNESS.....	57
VIII. CONCLUSION.....	59

TABLE OF AUTHORITIES

Cases

<i>Andersen Corp. v. Fiber Composites, LLC</i> , 474 F.3d 1361 (Fed. Cir. 2007)	19
<i>Apple Inc. v. Samsung Elecs. Co.</i> , 839 F.3d 1034 (Fed. Cir. 2016) (en banc)	57
<i>Curtiss-Wright Flow Control Corp. v. Velan, Inc.</i> , 438 F.3d 1374 (Fed. Cir. 2006)	22
<i>Facebook, Inc. v. Sound View Innovations, LLC</i> , IPR2017-00998-13 (PTAB Sept. 5, 2017)	26
<i>Lighting World, Inc. v. Birchwood Lighting, Inc.</i> , 382 F.3d 1354 (Fed. Cir. 2004)	34
<i>Littelfuse, Inc. v. Mersen USA EP Corp.</i> , 29 F.4th 1376 (Fed. Cir. 2022)	23
<i>Lyft, Inc. v. Rideshare Displays, Inc.</i> , IPR2021-01602-7 (PTAB Apr. 11, 2022)	29
<i>Phillips v. AWH Corp.</i> , 415 F.3d 1303 (Fed. Cir. 2005) (en banc)	31
<i>Ruckus Wireless, Inc. v. Innovative Wireless Sols., LLC</i> , 824 F.3d 999 (Fed. Cir. 2016)	21
<i>Samsung Elecs. Co. v. Smart Mobile Techs. LLC</i> , IPR2022-01248-13 (PTAB Jan. 24, 2023)	26
<i>SAS Inst., Inc. v. ComplementSoft, LLC</i> , 825 F.3d 1341 (Fed. Cir. 2016)	26
<i>SciMed Life Sys., Inc. v. Advanced Cardiovascular Sys., Inc.</i> , 242 F.3d 1337 (Fed. Cir. 2001)	20
<i>Sinorgchem Co. v. Intern. Trade Com’n</i> , 511 F.3d 1132 (Fed. Cir. 2007)	22
<i>UltimatePointer, LLC v. Nintendo Co., Ltd.</i> , 816 F.3d 816 (Fed. Cir. 2016)	20
<i>Ultratec, Inc. v. CaptionCall LLC</i> , 872 F.3d 1267 (Fed. Cir. 2017)	27

<i>Wang Labs., Inc. v. Am. Online, Inc.</i> , 197 F.3d 1377 (Fed. Cir. 1999)	21
---	----

<i>Williamson v. Citrix Online, LLC</i> , 770 F.3d 1371 (Fed. Cir. 2014) (en banc)	34
---	----

Statutes

35 U.S.C. § 112 para. 6 (pre-AIA)	33
---	----

35 U.S.C. § 311(b)	27
--------------------------	----

Rules

37 C.F.R. § 42.100(b)	28
-----------------------------	----

Other Authorities

Changes to the Claim Construction Standard, 83 Fed. Reg. 51,340 (Oct. 11, 2018)	28
--	----

Consolidated Trial Practice Guide (Nov. 2019)	28
---	----

CURRENT EXHIBIT LIST¹

No.	Brief Description
1001	U.S. Patent No. 8,989,715
1002	Expert Declaration of Ben Bederson (“Bederson”)
1003	Curriculum Vitae of Benjamin B. Bederson, Ph.D.
1004	File History of U.S. Patent No. 8,989,715
1005	File History of U.S. Patent No. 8,478,245
1006	U.S. Patent Application Publication No. 2007/0150617 (“Hariki”)
1007	U.S. Patent Application Publication No. 2003/0023755 (“Harris”)
1008	U.S. Patent No. 7,447,486 (“Tamura”)
1009	U.S. Patent No. 6,996,627 (“Carden”)
1010	U.S. Patent No. 6,669,564 (“Young”)
1011	U.S. Patent No. 6,732,183 (“Graham”)
1012	U.S. Patent No. 6,507,727 (“Henrick”)
1013	Webpage excerpt from Computer History Museum at https://www.computerhistory.org/tdih/april/6/ (last accessed March 29, 2023)
1014	https://www.webdesignerdepot.com/2009/05/the-evolution-of-cell-phone-design-between-1983-2009/
1015	Excerpts from Dan R. Olsen, Jr., Developing user interfaces (1998)
1016	Alok Sinha, Client-server computing, in Communications of the ACM, 35, 7 (1992)
1017	Tim Berners-Lee, Robert Cailliau, Ari Luotonen, Henrik Frystyk Nielsen, and Arthur Secret, The World-Wide Web, Communications
1018	Eric Kasten, HTML: A Gentle Introduction, Linux Journal (July 1995), available at https://www.linuxjournal.com/article/1081
1019	U.S. Patent Application Publication No. 2013/0124961 (“Linburn ”)

¹ Relevant portions of exhibits may be highlighted to help the Panel locate cited sections.

No.	Brief Description
1020	U.S. Patent No. 8,111,326 (“Talwar”)
1021	Order re Scheduling Conference, <i>GoTV Streaming, LLC v. Netflix, Inc.</i> , 2:22-cv-07556 (C.D. Cal. issued Feb. 13, 2023) (Doc. 61)
1022	Complaint in <i>GoTV Streaming, LLC v. Netflix, Inc.</i> , No. 2:22-cv-07556-RGK-SHK (C.D. Cal. October 17, 2022)
2001	CAMPBELL-KELLY ET AL., Mainframes to Smartphones: A History of the International Computer Industry, Ch. 11 (Harvard Univ. Press 2015)
2002	GALAZZO, Timeline from 1G to 5G: A Brief History on Cell Phones (Sept. 21, 2020), <i>available at</i> https://www.cengn.ca/information-centre/innovation/timeline-from-1g-to-5g-a-brief-history-on-cell-phones/
2003	Cingular MEdia Mall Games and Motorola Page (March 23, 2006), <i>available at</i> https://web.archive.org/web/20060323054853/http://www-xl.cingularextras.com/fuel/enduser/portal/endUserHTMLDir?c1=3&dc=0 and https://web.archive.org/web/20060323054841/http://www-xl.cingularextras.com/fuel/enduser/portal/endUserHTMLSelectPhone?makeName=motorola&dc=0
2004	Suite, Definition & Meaning - Merriam-Webster, <i>available at</i> https://www.merriam-webster.com/dictionary/suite
2005	U.S. Patent No. 7,380,205 to Bezrukov et al. (filed Oct. 28, 2003)
2006	TUTENEL ET AL., Rule-Based Layout Solving and Its Application to Procedural Interior Generation, Netherlands Organization for Scientific Research and the Netherlands ICT Research and Innovation Authority (2009)
2007	KRÖNER, Adaptive Layout of Dynamic Web Pages, Deutsches Forschungszentrum für Künstliche Intelligenz GmbH (2000)
2008	Scheduling Order, <i>GoTV Streaming, LLC v. Netflix, Inc.</i> , 2:22-cv-07556 (C.D. Cal. issued Feb 14, 2023) (Doc. 62)

No.	Brief Description
2009	Klausner Decisions re Motions to Stay Pending IPR, Docket Navigator (generated June 8, 2023)
2010	Order Granting Stipulated Stay Pending IPR, <i>Flexstent, LLC v. Abbott Labs.</i> , No. 5-18-cv-02479 (C.D. Cal. Oct. 18, 2019) (Doc.77)
2011	Order Denying Renewed Stipulated Stay Pending IPR, <i>Shenzhen Gooloo E-Commerce Co., Ltd. v. Pilot, Inc.</i> , No. 2-22-cv-02219 (C.D. Cal. Jan. 12, 2023) (Doc. 95)
2012	Renewed Joint Stipulation to Stay Case Pending Resolution of <i>Inter Partes Reviews</i> of '653 Patent, <i>Shenzhen Gooloo E-Commerce Co., Ltd. v. Pilot, Inc.</i> , No. 2-22-cv-02219 (C.D. Cal. Jan. 11, 2023) (Doc. 94)
2013	Klausner Time to Trial in Patent Cases, Docket Navigator (generated June 8, 2023)
2014	U.S. District Court—Judicial Caseload Profile for Central California from Federal Court Management Statistics—Profiles (Mar. 31, 2023), <i>available at</i> https://www.uscourts.gov/sites/default/files/data_tables/fcms_na_distprofile0331.2023.pdf
2015	Order Denying Netflix's Motions, including for Invalidity under § 101, <i>GoTV Streaming, LLC v. Netflix, Inc.</i> , 2:22-cv-07556 (C.D. Cal. issued May 24, 2023) (Doc. 109)
2016	Order re Scheduling Conference, <i>GoTV Streaming, LLC v. Netflix, Inc.</i> , 2:22-cv-07556 (C.D. Cal. issued Feb. 13, 2023) (Doc. 61)
2017	Excerpts from Redacted and De-designated Corrected Declaration of Dr. John Villasenor Regarding Invalidity of U.S. Patent Nos. 8,989,715; 8,478,245; and 8,103,865, <i>GoTV Streaming, LLC v. Netflix, Inc.</i> , 2:22-cv-07556 (C.D. Cal. served June 23, 2023)
2018	Declaration of Joshua S. Wyde regarding authentication of exhibits (Aug. 10, 2023)
2019	Excerpts from Trial Transcript Day 2, <i>GoTV Streaming, LLC v. Netflix, Inc.</i> , No. 2:22-cv-07556 (C.D. Cal. Oct. 18, 2023)

No.	Brief Description
2020	Excerpts from Trial Transcript Day 3, <i>GoTV Streaming, LLC v. Netflix, Inc.</i> , No. 2:22-cv-07556 (C.D. Cal. Oct. 19, 2023)
2021	Defendant Netflix, Inc.’s Notice of Motion and Rule 50(a) Motion for Judgment as a Matter of Law, <i>GoTV Streaming, LLC v. Netflix, Inc.</i> , No. 2:22-cv-07556 (C.D. Cal. Oct. 19, 2023) (Dkt. 389)
2022	Declaration of Mr. Stuart Lipoff
2023	Generate, Free On-Line Dictionary of Computing, <i>available at</i> https://foldoc.org/generate (last updated June 15, 1995).
2024	Output, TechTerms.com; The Computer Dictionary, <i>available at</i> https://techterms.com/definition/output (last updated December 12, 2006)
2025	From, Merriam-Webster.com Dictionary, <i>available at</i> https://www.merriam-webster.com/dictionary/from (last visited December 12, 2023)
2026	Customized, Merriam-Webster.com Dictionary, <i>available at</i> https://www.merriam-webster.com/dictionary/customized (last visited Jan. 05, 2024)
2027	Application-Specific, Wiktionary, <i>available at</i> https://en.wiktionary.org/wiki/application-specific (last visited Dec. 15, 2023)
2028	Excerpt from 10-K statement filed by Netflix with the Securities and Exchange Commission (Jan. 26, 2023)
2029	Excerpt from Netflix Women In Tech Event - UI Engineering - August 2018, Jessica Berglund, Netflix UI engineer, 22:43–23:02, <i>available at</i> https://www.youtube.com/watch?v=YpIGejJiaBo
2030	Ethan Kao Deposition Designations Video, <i>GoTV Streaming, LLC v. Netflix, Inc.</i> , No. 2:22-cv-07556 (C.D. Cal. Oct. 19, 2023) (Dkt. 413-13)

No.	Brief Description
2031	Deposition of Dr. Benjamin Bederson (Jan. 10, 2024) ²
2032	Second Declaration of Joshua S. Wyde regarding authentication of exhibits (Jan. 26, 2024)

² Per Party agreement, Dr. Bederson gave a consolidated deposition regarding U.S. Patent Number 8,989,715 in IPR No. IPR2023-00757; Patent No. 8,478,245 in IPR No. IPR2023-00758; and U.S. Patent No. 8,103,865 in IPR No. IPR2023-00759. *See* Ex. 2031, 5:21–6:11.

CLAIM LISTING

	Limitation
1[pre]	1. A method of generating content that is renderable by a wireless device, said method comprising:
1[a]	transmitting, to said wireless device, 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; and
1[c]	transmitting, to said wireless device, compiled content comprising (i) first compiled content specific to a first page of said application and (ii) second compiled content specific to a second page of said application,
1[d]	wherein said compiled content is generated in part from execution of said application,
1[e]	wherein said compiled content comprises render commands expressed in a syntax that is generic to said wireless device, and
1[f]	wherein said custom configuration is applicable to said first and second compiled content,
1[g]	wherein said compiled content and said custom configuration are usable by a graphical user interface comprising said plurality of rendering blocks to generate renderable content based on said compiled content and said custom configuration.
2	2. A method as described in claim 1 wherein said renderable content comprises audio content and display content.
3	3. A method as described in claim 1 wherein said compiled content is partially resultant from said application operating on a remote server.
4	4. A method as described in claim 1 wherein said compiled content is specific to the rendering capabilities of said wireless device.
5	5. 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.
6	6. A method as described in claim 5 wherein said custom configuration comprises a syntax that is generic regarding said wireless device type.
7	7. A method as described in claim 1 wherein said custom configuration comprises configuration information and content specific to said application.
8	8. 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 said method further comprises transmitting an identifier that identifies said custom configuration.

	Limitation
9[pre]	9. A non-transitory computer readable medium comprising instructions therein that when executed by a processor implement a method of generating content that is renderable by a wireless device, said method comprising:
9[a]	transmitting, to said wireless device, an identification of a custom configuration of a plurality of rendering blocks of said wireless device,
9[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; and
9[c]	transmitting, to said wireless device, compiled content comprising (i) first compiled content specific to a first page of said application and (ii) second compiled content specific to a second page of said application,
9[d]	wherein said compiled content is generated in part from execution of said application,
9[e]	wherein said compiled content comprises render commands expressed in a syntax that is generic to said wireless device, and
9[f]	wherein said custom configuration is applicable to said first and second compiled content,
9[g]	wherein said compiled content and said custom configuration are usable by a graphical user interface comprising said plurality of rendering blocks to generate renderable content based on said compiled content and said custom configuration.
10	10. A non-transitory computer readable medium as described in claim 9 wherein said renderable content comprises audio content and display content.
11	11. A non-transitory computer readable medium as described in claim 9 wherein said compiled content is partially resultant from said application operating on a remote server.
12	12. A non-transitory computer readable medium as described in claim 9 wherein said compiled content is specific to the rendering capabilities of said wireless device.
13	13. A non-transitory computer readable medium as described in claim 9 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.
14	14. A non-transitory computer readable medium as described in claim 13 wherein said custom configuration comprises a syntax that is generic regarding said wireless device type.
15	15. A non-transitory computer readable medium as described in claim 9 wherein said custom configuration comprises configuration information and content specific to said application.

	Limitation
16	16. A non-transitory computer readable medium as described in claim 9 wherein said method further comprises transmitting an identifier that identifies said custom configuration.
17[pre]	17. A server that is programmed to generate content that is renderable by a wireless device, comprising:
17[a]	a library of applications;
17[b]	a library of custom configuration data comprising a custom configuration that configures a plurality of rendering blocks of said wireless device to render content in a manner customized to an application from said library of applications requested by said wireless device; and
17[c]	a layout solver that transmits compiled content to said wireless device, said compiled content comprising (i) first compiled content specific to a first page of said application and (ii) second compiled content specific to a second page of said application,
17[d]	wherein said compiled content is generated in part from execution of said application by said server,
17[e]	wherein said compiled content comprises render commands expressed in a syntax that is generic to said wireless device, and
17[f]	wherein said custom configuration is applicable to said first and second compiled content,
17[g]	wherein said compiled content and said custom configuration are usable by a graphical user interface comprising said plurality of rendering blocks to generate renderable content based on said compiled content and said custom configuration.
18	18. A server as described in claim 17 wherein said renderable content comprises audio content and display content.
19	19. A server as described in claim 17 wherein said compiled content is specific to the rendering capabilities of said wireless device.
20	20. A server as described in claim 17 wherein said custom configuration comprises configuration information and content specific to said application.

I. INTRODUCTION

Petitioner identifies Hariki's "web browser" as the claimed "application" for some limitations, while Petitioner identifies a Hariki/Harris "web server" for others. However, the claims concern *a single application* with specific limitations that *neither Hariki's web browser nor Hariki/Harris' web server individually meet*.

The Hariki/Harris web server cannot be the claimed "application" because Petitioner-identified "custom configuration," a "UI skin," does not configure a plurality of rendering blocks of said wireless device to render content *in a manner customized to the web server*. Hariki's UI skin is not custom designed for a web server. The UI skin changes the appearance of a wireless device, and does not affect, and is not affected by, a disclosed web server supplying HTML. A Hariki/Harris web server sends the same HTML in response to the same request, regardless of whether the requestor is using a UI skin, and, if so, what UI skin is in use, and, vice versa, the device processes the same HTML the same way, regardless of what web server sent the HTML.

Likewise, Hariki's web browser cannot be the claimed "application" for at least three reasons. First, Hariki's web browser does not *transmit* the Petitioner-identified "compiled content"—i.e., HTML pages to a wireless device. Hariki's web browsers *receive* HTML content, they do not transmit it. Second, Hariki's web browser is not a *remote* application from the wireless device. Hariki's web browser

is a local application, running entirely on a wireless device. And third, Hariki's web browser does not *generate*—even in part—the HTML pages. The HTML files may be remotely generated with Harris' mobile content framework ("MCF"), but not a single character in the HTML file is produced by Hariki's web browser.

Moreover, the Petitioner-identified "custom configuration" (i.e., Hariki's UI skin), is not "applicable to" the Petitioner-identified "compiled content" (i.e., the HTML files from the Hariki/Harris server). Petitioner and the Board have taken the position that because a UI skin is usable by Hariki's web browser that the UI skin must be similarly usable by HTML within the web browser. However, there is *no teaching* that the HTML pages sent to the web browser use the UI skin configured "rendering blocks," and Petitioner has not shown that this ability is inherent. To the contrary, because Harris provides the option to make sure that the HTML files are "displayed correctly" remotely from the wireless device, and before the HTML files are even sent to a web browser that might access the UI skin on a wireless device, the HTML files *cannot* be using the UI skin.

II. OVERVIEW OF THE PATENT-AT-ISSUE

A. Subject Matter

Circa August 1, 2007, wireless devices generally ran "application[s] locally." Ex. 1001 2:3–5. And there were multiple types of portable wireless devices, multiple manufacturers of each type of device, multiple models from each

manufacturer, and multiple versions of each model. *Id.* 1:21–32. Each particular combination of type, manufacturer, model, and version of a device potentially had unique “device attributes” (i.e., rendering capability, display resolution, color capability, display size, etc.). *Id.* 1:32–36; 11:20–31; 11:43–56.

Therefore, developers that wished to write an application to run on multiple devices and “fully utilize” each device’s attributes had to tailor versions of the application per device, building each one “from the ground up.” *Id.* 1:37–43; 51–54. Given that providers had to pay to develop, debug, and qualify every version of an application, *id.* 1:44–47, many software developers simply chose to limit their titles to a few platforms, *id.* 1:62–67.

Additionally, maintenance of an application (i.e., upgrades, updates, and patches to the code) across target platforms similarly required a costly per-device effort. *Id.* 2:1–7. And even if the application developer did provide updates, many users failed to install the update to the locally running applications because it was a “laborious process” of “navigating through multiple pages in order to determine whether an update [was] ready” and “actively initiat[ing] the update process.” *Id.* 2:7–14.

In light of the state of application development at the time, the Patent-at-Issue contemplates a method and system for rendering centrally running remote applications on a wireless device where: (1) the application “output . . . is device

specific based on the wireless device attributes,” *id.* 2:22–26, with its own “look and feel,” *id.* 7:55–63; 10:56–58; 18:37–42; yet (2) the developers can still write a “generic application” being less concerned with “tailoring their applications for a given device type,” *id.* 2:22–26. Additionally, the claimed solution allows for developers to maintain applications centrally, “eliminating the need to update/patch each application on each wireless device.” *Id.* 2:37–43.

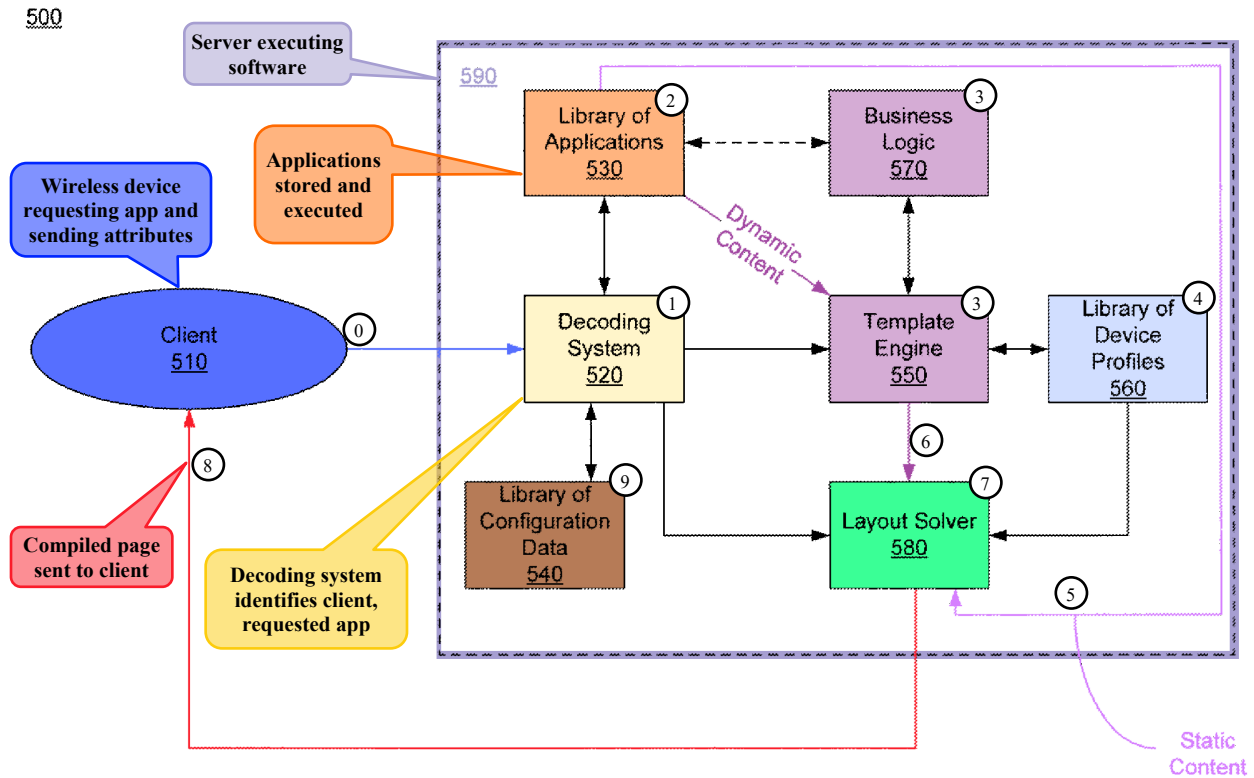
B. Invention as Illustrated through Embodiments

1. Applications

In the context of the Patent-at-Issue, an application can be “defined by an application description file and a set of static resources, e.g., images, sounds, URLs, etc. The application description file describes all the screens of the application and the custom configuration of the application.” *Id.* 16:45–49. For example, an application may be all the screens, resources, and custom configurations to implement the Texas Hold’em Poker application. *Id.* 16:39–42.

The Patent-at-Issue’s applications are “stored” and “executed” “on the remote server,” *id.* 2:37–38; 2:63–64; 3:8–9; 5:54–56; 6:6–7; 16:44–45, in contrast to traditional applications that were downloaded and locally run on wireless devices.

The applications may be stored in a library of applications 530 and are executed on a remote server 590 to generate compiled content.



Ex. 1001 Fig. 5.³ At ① a client “request[s] an application [and] identif[ies] the client’s 510 type and its capabilities.” *Id.* 16:34–37. In response, at ①, the “decoding system” on the server “locate[s] and execute[s] the requested application.” *Id.* 16:39–42.

Upon execution, at ②, the application generates the pages “for display on the wireless device.” *Id.* 16:61–62. This page can contain only static content, such as “pictures or . . . pure text.” *Id.* 16:62–66. Or this page may be a “generic template describing the screen.” *Id.* 17:3–6.

³ All color, balloon text, circle numbering, and annotations in figures are Patent Owner’s additions unless otherwise noted.

In the case of the generic template page, at ③, a “template engine,” also executing on the server, may first “merge[] screens of [the] application and dynamic data provided by the business logic” to “provide[] a high level template (e.g., extensible markup language (XML) format),” *id.* 17:6–26, and then consult the “library of device profiles,” at ④, to “eliminate content from its dynamic template output that is not supported by the client,” *id.* 17:27–39. For example, the template engine “may eliminate audio component[s] from the generic template output” when “the client 510 does not have audio capability,” or “eliminate big images” “when the client 510 is short in memory.” *Id.* 17:39–47.

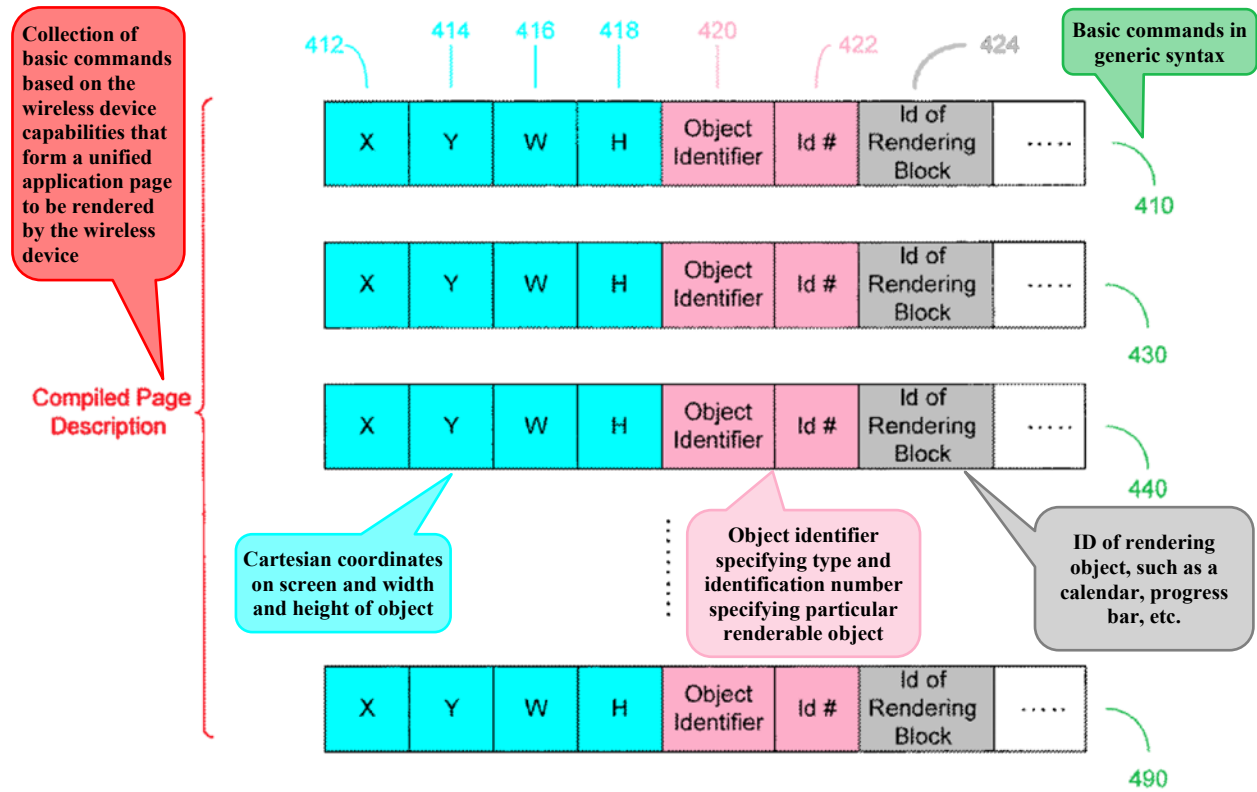
Whether the page is static (i.e., a “generic template supplied from 530,” at ⑤) or dynamic (i.e., a “template supplied from 550,” at ⑥), the generic application pages at this point “do not contain any specific or actual screen dimensions but rather only contain relative screen locations.” *Id.* 17:48–51. In this sense, they are “wireless device generic.” *Id.*

2. Compiled Content and Rendering Blocks

The Patent-at-Issue teaches that an application’s generic pages can be fed into a “layout solver,” at ⑦, that actually “translates the received information into a series of basic commands based on the client’s capabilities,” *id.* 17:53–59; 18:1–6, to create a “compiled page” that is sent to the wireless client device at ⑧, *id.* 18:11–14. Note that while the compiled page’s basic commands and their parameters are

tailored to, and based on, the wireless device capabilities, the syntax of these commands is still “device generic.” *Id.* 18:7–11.

The Patent-at-Issue illustrates what a compiled page for an application may look like:



Ex. 1001 Fig. 4. The compiled page contains a series of basic commands, including 410 to 490. Note that the compiled pages may now specify actual “Cartesian coordinates 412 and 414 of a screen region” as well as “the width 416 and the height 418 of the screen region.” *Id.* 16:2–7.

The basic commands include an “object identifier” that specifies the type of “renderable component” to be displayed (e.g., a designator for “text, image, video, audio, and the like”) and an “identification number” that specifies the particular

renderable component of the designated type to be rendered. *Id.* 13:27–29; 16:16–21; 15:60–62.

The render commands found in the exemplary device specific page description also include the identifications of the “rendering blocks.” Rendering blocks are conceptually part of the wireless device’s graphical user interface (“GUI”) that are “preconfigured to operate and store data” and “perform discrete rendering operations.” *Id.* 2:50–51; 4:7–11; 7:18–22. Examples of a GUI’s rendering blocks include text; a text box; an image; a pop-up, drop-down or tabbed menu; audio/video controls; a scrolling ticker; check box/radio buttons; a rating panel; poll results; a drawing canvas; a hierarchical tree; scroll bars; a progress bar; a table; a calendar; and the like. *Id.* 8:25–43. By referencing the rendering blocks that were customized by the custom configuration the rendering blocks are “applicable” to the compiled content.

The specification also notes that not all components of a basic command are shown in Figure 4, and “that basic commands may further include other field components for additional information[,] e.g., immediate data, such as text.” *Id.* 16:26–28.

In sum, a “collection of basic commands . . . forms a single unified page to be rendered by the wireless device.” *Id.* 15:63–65.

3. Custom Configuration

As taught by the Patent-at-Issue, if the client requests to run an application, the decoding system can also “access a library of custom configuration data 540,” shown as ⑨ above, “to customize the appearance of the application” and “send the identified custom configuration to the client.” *Id.* 16:50–60.

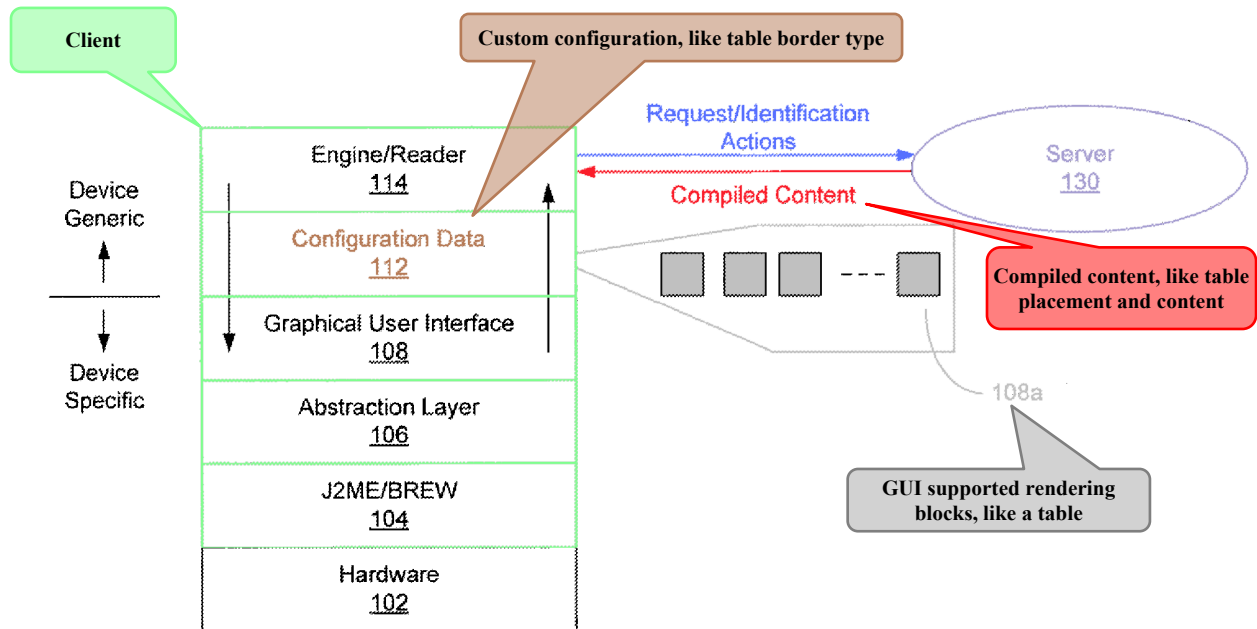
The client can use the custom configuration to configure the GUI’s rendering blocks to set how they “operate and look.” *Id.* 7:55–63; 10:56–58; 18:40–42. For example, custom configuration data may configure the GUI rendering blocks’ fonts, colors, border thickness, images, background images, icon type/image, menu styles, sound, video, etc., *id.* 8:50–10:55, such as “customiz[ing] a ‘submit’ icon to look like an airplane flying away when pressed,” *id.* 7:64–8:1.

The custom configuration may be globally applicable to pages of an application. Stated otherwise, new application compiled content can be rendered using the GUI’s rendering blocks that were previously configured without the application sending new, or resending the previous, custom configuration data. *See, e.g., id.* 14:15–43 & Fig. 2B.

4. Client

The Patent-at-Issue refers to the software on the wireless devices that interacts with the application on the server, displays application content, and detects user interaction with the application as the “client.” *Id.* 6:16–21. The client generally

comprises the “software stack” conceptualized as residing at a higher level than the wireless device hardware as shown in “Blocks 104 through 114” below. *Id.* 6:57–59.



Ex. 1001 Fig. 1B. It is the client that “draw[s] the screen according to the description as specified by the received discrete low level basic commands and in accordance with the identified custom configuration.” *Id.* 13:32–35. Specifically, the client can configure rendering blocks with the configuration data, and then an engine/reader can process the compiled content to ultimately render the remotely running application’s pages. *Id.* 7:26–31; 18:62–19:3.

III. OVERVIEW OF THE CITED REFERENCES

A. Hariki

Hariki concerns “a mobile device UI configuration system that allows a UI

skin package content provider to easily generate a new UI resource file for new or different mobile devices, and make these UI skin packages available for download and installation on user mobile devices.” Ex. 1006 [0008]. With these “UI skins,” Hariki “provide[s] user interface (UI) customization capabilities that allow users to personalize their phones or mobile devices with custom ringtones, background displays (wallpaper), menu configurations, and the like,” *id.* [0005], to replace the “default user interface for each model of device,” *id.* [0007].

Specifically, Hariki contemplates a “user interface authoring tool” that “maintains a common database of resources for each mobile device model” “defined by device manufacturers.” *Id.* [0019]. 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.* [0019]. “Thus, as shown in FIG. 2, a resource profile is provided for device A, and a resource profile is provided for device B. With reference to FIG. 1, these can correspond to resource profiles for mobile devices 108 and 109, respectively.” *Id.* [0031].

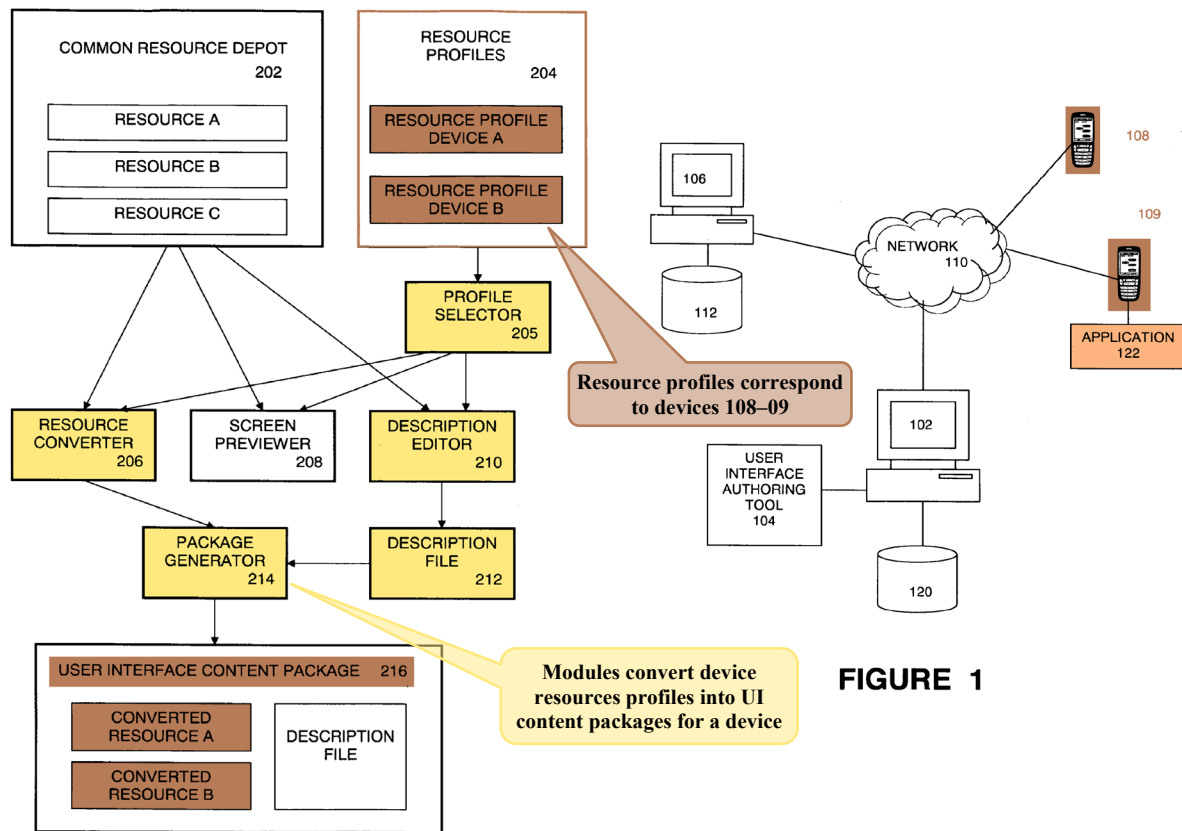


FIGURE 1

FIGURE 2

Ex. 1006 Figs. 1–2.

Hariki explains that various “modules” (205–12) use the resource profiles for the devices to create a UI content package 216 for a device. *Id.* [0032–33]. Specifically, “[t]he profile selector 205 selects a resource profile 204 depending upon the device model selected by the content provider. The corresponding resource 202 for that model is then converted by resource converter 206.” *Id.* [0032]. “A description editor component 210 produces description files 212 . . . [and t]he description file 212 and the converted resource output from resource converter 206 are processed by a package generator component 214 . . . [that] takes the resource

and description and creates a UI content package 216.” *Id.* [0033].

Once created, “[t]he actual UI content package to be downloaded is generally selected by the user of the mobile device.” *Id.* [0044]. And “the user can change the UI content by changing the [device’s settings through the] setting application 504.” *Id.* With the skin in place, “one or more application programs 122 or software modules . . . access the resources provided by the UI content package.” *Id.* [0040]. That is, any application or program “can playback or perceive the image, video, sound files, etc. of the resources provided by the UI content package.” *Id.* In this way, “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.” *Id.* [0024].

B. Harris

Harris concerns “delivery of the same content to multiple mobile devices using different device protocols.” Ex. 1007 [0002]. To do this, Harris envisions a mobile content framework (“MCF”) that uses “a generic markup language, referred to as Wireless Abstract XML (hereafter WAX).” *Id.* [0007]. A designer may author pages “created in WAX originally,” or “content [may be] first translated in to WAX from the original language of the content provider.” *Id.* The content may be stored at any “content location” interfaced with the WAX server. *Id.* [0008].

Whatever the source and wherever the content resides, the MCF may translate

the “WAX language . . . to a requesting device’s native language . . . [or] from WAX to languages like WML, HDML and HTML.” *Id.* [0018]. For example, WAX element “<wax:button>” may be sent “as a ‘soft-key’ for WAP devices, but as a ‘link’ for devices which understand only HTML.” *Id.* [0030].

However, Harris also notes that “even when two devices employ the same language, there is no guarantee that they each interpret and render content the same way. Therefore, “not only [does Harris] translate[] [WAX] into a specific language appropriate for the device, but . . . [the translation] is tailored based upon the attributes of the requesting device.” *Id.* [0018]. That is, the MCF may handle “the idiosyncrasies of each device,” such as considering the fact that Nokia’s WAP browser “does not display lists in the same fashion as the [WAP] browsers on phones from other manufacturers.” *Id.* [0017].

Harris also teaches that, in one embodiment, an automated testing environment can make sure the content will be displayed correctly before the translated device-specific files are deployed to a device. *Id.* [0035].

Ultimately, only the device-specific “customized content is . . . returned to the requesting device.” *Id.* [0024] & Fig. 3.

IV. LEVEL OF ORDINARY SKILL IN THE ART

Patent Owner generally concurs with Petitioner that a person of ordinary skill in the art (“POSITA”) would have an electrical/computer engineering degree or a

degree in a closely related field. Pet. 18. And Patent Owner’s view is that the POSITA would have a couple of years’ experience in the “field of wireless communication systems” and “[m]ore particularly, . . . method[s] and system[s] for rendering applications on a wireless device.” Ex. 1001 1:13–16. More education may substitute for experience, and vice versa. Patent Owner’s position is that the patentability analysis would not change based under either Party’s definition.

V. CLAIM CONSTRUCTION

A. District Court Constructions

The District Court has construed the following terms, Ex. 3004, p. 10:

Term	Construction
custom configuration	A configuration that determines the look and feel of the application
rendering blocks	Blocks that perform discrete rendering operations

Patent Owner uses these definitions; however, Patent Owner does not believe that these terms are in controversy in this proceeding.

Instead, Patent Owner proposes the following claim constructions that may be required for the final written decision.

B. “Customized to Said Application” Means “Created For” the Application Program

1[b]/9[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

17[b] . . . custom configuration that configures a plurality of rendering blocks of said wireless device to render content in a manner customized to an application . . .

“Customized to said application” is found in the larger limitations 1[b]/9[b]/17[b] seen above where the custom configuration configures rendering blocks in an application-customized manner. The natural reading of the limitation in context is that “customized to said application” means “tailored to an application,” “designed for the application,” or “made with the application in mind.” As Patent Owner’s expert explains, “customized” generally means that the configuration was built, fit, or altered according to individual specifications. Ex. 2022 ¶121 (citing Ex. 2026). And in the software context, customized to an application, is the equivalent of application-specific, meaning “created or written for a specific application.” *Id.* (citing Ex. 2027); *see also* Ex. 2031 84:17–85:1 (thinking of no distinction between “specific to said application” and “application-specific”).

This natural reading is confirmed by the specification that teaches the custom configuration can only be determined after the application is known. Ex. 1001 3:9–11 (“[T]he server may identify a custom configuration to be used *for the requested*

application.”); *Id.* 16:53–56 (same). Moreover, the specification teaches that the “library of custom configuration data [is] where each application may have a *corresponding* custom configuration data to customize the appearance of the application.” *Id.* 16:50–53; 19:25–29 (same). Thus, the custom configuration “corresponds” to an individual application (i.e., was consciously designed and is chosen with the application in mind) and is not merely a configuration that happens to be applied to an application. Ex. 2022 ¶122.

While Petitioner and Petitioner’s expert have not proffered an explicit definition for this term, Pet. 19; Ex. 2031 25:21–26:6, Dr. Bederson explained that the HTML Harris produces is customized content because this “HTML is an example of files that are *tailored to* a particular mobile device,” Ex. 2031 41:14–42:16 (emphasis added). I.e., Netflix’s expert has used “tailored to” as synonymous with the plain meaning of “customized to” as well.

Considering the above, “customized to said application” means “created for” the application.

C. All Claims Require the Recited “Application” to Run Remotely

1[c]/9[c]/17[c] [transmitting, to said wireless device, / . . . transmits compiled content to said wireless device, said] compiled content comprising (i) first compiled content specific to a first page of said application and (ii) second compiled content specific to a second page of said application

1[d]/9[d]/17[d] wherein said compiled content is generated in part from execution of said application [by said server]

The claims, specification, and both parties agree that the recited “application” must be executing remotely from the wireless device.

1. The Claims Recite a Remote Application Transmitting Compiled Content Pages to a Separate Client

The claims require the “application” to run remotely from the wireless device. All claims recite “transmit[ing] compiled content to said wireless device, said compiled content comprising . . . page[s] of said application” or “transmitting, to said wireless device, compiled content comprising . . . page[s] of said application.” *See* limitation 1[c], 9[c], and 17[c]. These limitations require the application content pages initially be remote, because, if the application content pages were already on the wireless device, it would make no sense to “transmit” them to where the pages were already located. Ex. 2022 ¶98. And all independent claims recite “said compiled content is generated in part from execution of said application,” *see* limitation 1[d], 9[d], and 17[d], so the application generating, at least in part, the remote pages that are transmitted to the wireless device must also be remote. Ex. 2022 ¶¶99–100 (explaining what generated from execution means).

And claim 17 has an additional limitation that compiled content is not only generated remotely and transmitted to the wireless device, but that the remotely generated content is also generated in part from execution of said application “by said server,” leaving no doubt that the application is operating on the server, not the separately recited “wireless device.”⁴ *See* limitation 17[d]; *see also* Ex. 2022 ¶101.

2. The Specification Exclusively Promotes a Remote Application Executing on a Server Such that There Is No Written Description Support for Claiming a Locally Running Application

In line with the claims, the specification pervasively advocates only one invention where the application is remotely running on a server. Ex. 2022 ¶104. From the very beginning, the abstract, meant to give a concise understanding of the invention, teaches that “[t]he wireless device receives compiled content *from the*

⁴ Note that, in the same way Petitioner’s superseded claim differentiation argument that dependent claims 3 and 11 reciting the “application operating on a remote server” means independent claims 1 and 9 must not be limited to a remote application fails, an argument that claim 17’s limitation “by said server” means claims 3 and 11 do not have to be remote also fails. *See* explanation in Section V.C.3 beginning on page 21 below; *see also* Ex. 2022 ¶¶102–03, 110 (explaining that “remote” and “by a server” are not equivalent). Additionally, claim differentiation does not apply to Claim 17 that has narrower “library” limitations, because the limited claim construction tool does not apply where there are “other differences varying the scope of the claimed subject matter.” *Andersen Corp. v. Fiber Composites, LLC*, 474 F.3d 1361, 1370 (Fed. Cir. 2007).

application executing on the server.” Ex. 1001 abstract (emphasis added); *see also SciMed Life Sys., Inc. v. Advanced Cardiovascular Sys., Inc.*, 242 F.3d 1337, 1342 (Fed. Cir. 2001) (restricting limitation in part in light of the abstract, because “from the outset the specification” characterized the limitation as having the restriction).

The Patent-at-Issue also sets forth the problems that arise when “a wireless device runs the application locally,” Ex. 1001 2:3–14, and teaches a solution that only comes about “since applications and basic commands for rendering applications *are performed and generated on the server,*” *id.* 4:21–25 (emphasis added). *See also id.* 2:38–41 (“[T]he present invention execute[s] the requested application on the server. Thus providing software updates and patches for a generic application can be done by patching/updating the generic application on the server, thereby eliminating the need to update/patch each application on each wireless device.”); 5:56–62 (same); 6:34–42 (same); 20:21–25 (same); *see also UltimatePointer, LLC v. Nintendo Co., Ltd.*, 816 F.3d 816, 823–24 (Fed. Cir. 2016) (finding extolling virtues of one feature, and criticism of its opposite, to limit the claims to the feature, even if another construction was plausible if the claims were divorced from the specification); *SciMed Life*, 242 F.3d at 1342–43 (same).

Indeed, the Patent-at-Issue *exclusively* notes the embodied “application is executed on the server.” Ex. 1001 3:8–9; 5:54–56; 6:6–7; 6:29–31; 6:38–39; 13:11–13; 16:44–45; 18:56–58; 19:38–39. And all relevant figures show the application is

executing on a server and the application is transmitting compiled content pages to the remote client. *Id.* Figs. 1B, 2A–B, & 5. The flow charts—echoing the claims—show the wireless device is “[r]eceiving compiled content generated in part *from execution of the application by the server.*” *Id.* Fig. 6 (item 640) (emphasis added); *see also* Fig. 7. Therefore, a construction that encompasses an application executed on the wireless device would be invalid for lack of written description support as a POSITA would not understand how a local application would fit within the invention, Ex. 2022 ¶¶104–05, and “claims are not properly construed to have a meaning or scope that would lead to their invalidity for failure to satisfy the requirements of patentability.” *Wang Labs., Inc. v. Am. Online, Inc.*, 197 F.3d 1377, 1382–83 (Fed. Cir. 1999); *see also Ruckus Wireless, Inc. v. Innovative Wireless Sols., LLC*, 824 F.3d 999, 1004 (Fed. Cir. 2016) (holding that “canon favoring constructions that preserve claim validity . . . counsels against” “construing the instant claims to encompass . . . subject matter [that] would likely render the claims invalid for lack of written description”).

3. All Parties Agree that the Claims Require the Application Be Executing Remotely

Initially, Petitioner implicitly invoked claim differentiation, arguing that “claim 3 . . . further limits the ‘application’ in claim 1 to ‘operat[e] on a remote server’ . . . Thus, POSITAs would have understood that the claimed application in claim 1 could reside anywhere, for example on a client or on a server.” Pet. 36 n.10.

First, this claim differentiation argument does not work. Claim 1 and 9 do not require operation on a server, and only require a remote application. Dependent claims 3 and 11, further limits the remote application to one “operating on a . . . server.” “Remote operation” and “execution on a server” are not identical limitations. Ex. 2022 ¶¶101–02, 110. Therefore, construing claims 1 and 9 to require a “remote” application would not render the “server” language of claims 3 and 11 superfluous.⁵ In a case like this, where the claims are not “otherwise identical” but for the “remote” limitation, one should not apply the doctrine of claim differentiation. *Sinorgchem Co. v. Intern. Trade Com’n*, 511 F.3d 1132, 1140 (Fed. Cir. 2007) (noting that when a dependent claim is “narrower in scope, the claims are not rendered identical and present no claim differentiation problem”).

Second, even if Petitioner was correct—which it is not—claim differentiation is a “limited tool of claim construction.” *Curtiss-Wright Flow Control Corp. v. Velan, Inc.*, 438 F.3d 1374, 1378 (Fed. Cir. 2006). And because the “presumption of differentiation in claim scope is not a hard and fast rule,” “any presumption

⁵ This also applies to claim 17 and its “by said server” limitation. *Curtiss-Wright Flow Control Corp. v. Velan, Inc.*, 438 F.3d 1374, 1381 (Fed. Cir. 2006) (“[C]laim differentiation takes on relevance in the context of a claim construction that would render additional, or different, language in another independent claim superfluous.”).

created by the doctrine of claim differentiation *will be overcome by a contrary construction dictated by the written description,*” such as the lack of written description support as just noted. *Littelfuse, Inc. v. Mersen USA EP Corp.*, 29 F.4th 1376, 1380 (Fed. Cir. 2022) (emphasis added, internal quotes removed, and citations omitted); *Curtiss-Wright*, 438 F.3d at 1381 (explaining that a court may not “rely[] on the claim differentiation presumption” if the presumption leads to a construction “not consistent with the overall context of this invention and this field of art as described in the specification.”).

Third, in the trial before the District Court, Petitioner and its trial expert made it clear that Petitioner has materially changed its claim construction position, and that the claims do require the application to execute remotely “on the server” (though, as discussed, “on a server” is necessarily remote from the wireless device, but not all remote applications are on a server).⁶ Ex. 2019 (Petitioner Expert’s Testimony), 209:12–14 (“The claimed application—it’s very clear in the claims in the patent—the application has to be executing on the server.”); *id.* 216:8–10 (“So the patents, the asserted claims require an application that’s executing on a server.”); *id.* 222:10–12 (“So it’s important to emphasize that the claims require they recite an

⁶ Petitioner was discussing claim 4, dependent on claim 1 of the ’715 Patent, and claim 16, dependent on claim 12, in the ’715 Patent’s Parent, U.S. Patent No. 8,478,245.

application, and that application has to be executing on the server.”); *id.* 225:11–12 (“Well, according to the patent, the application is—needs to be executing on the server.”); *id.* 228:6–9 (“Yeah. The—it has to be for an application, the claimed application. And the claimed application in the patents has to be executing on the server.”); *id.* 229:4–6 (“And that application has to be the same application that I discussed before, the one that’s executing on the server.”); Ex. 2020 (Closing Argument by Petitioner), 42:12–13 (“[R]emember, the application in the claims is what’s on the server.”); *id.* 46:10–17 (“[F]ocus[ing] on where that application lives . . . both [experts] agree that the application lives on the server, on the back end. No dispute about that.”).

Netflix and its expert explicitly differentiated the claimed application from anything that could be called an application on the wireless device:

Q. The device has its own software application on it, correct?

A. It has software on it.

Q. Is that software not an application?

A. Well, I just want to be careful to make sure that the word application is not misused. It is not the claimed application . . .

Q. So you have an application, claimed application on the server, correct?

A. There must be an application on the server. That’s right.

Q. Then the app—a different application on the wireless device renders a user display, correct?

A. Well, it renders a display that shows the application that’s running on the server, yeah.

Ex. 2019, 243:4–24; *see also* Ex. 2020, 39:7–13 (“So remember, I showed you the evidence that both experts agree that the application that’s in the claim is a back-end application. In fact, you remember I asked [Patent Owner’s Expert]: If I download the Netflix app on my phone, is that the application in the claim? He said no. He agreed with me. Both experts agree. The application must be on the back end.”).

Petitioner’s position was not conditional. Ex. 2022 ¶¶106–07. That is, Petitioner did not say that “under GoTV’s construction where the application is remote” we would not infringe. *Id.* Petitioner unreservedly adopted this position as its own, and simply noted that Patent Owner’s position was the same. *Id.*

Consistent with its trial position, Petitioner filed a Motion for Judgment as a Matter of Law after the trial that there is no infringement of the ’715 Patent because “[t]he ‘application’ required by the claims is on the server.” Ex. 2021, p. 6, ll. 2–3 (citing Ex. 2019, 209:12–14, 216:8–9, 222:7–16).

Patent Owner characterizes this construction as Petitioner’s new position

because Petitioner should not be able to take irreconcilable positions.⁷ The Federal Circuit prohibits “simultaneously argu[ing]” “incongru[ous] . . . claim[] requirements for infringement versus invalidity” in a District Court case. *CommScope Techs. LLC v. Dali Wireless Inc.*, 10 F.4th 1289, 1299 (Fed. Cir. 2021). This prohibition should be no different for a party before the PTAB and the District Court, because “[t]he AIA [IPR] proceeding is structured as a complete alternative to litigation of . . . issues” dealing with unpatentability over printed publications and patents. *SAS Inst., Inc. v. ComplementSoft, LLC*, 825 F.3d 1341, 1354

⁷ The Institution Decision cited a routine decision, *Samsung Elecs. Co. v. Smart Mobile Techs. LLC*, IPR2022-01248-13, at 58 (PTAB Jan. 24, 2023), for the proposition that a Petitioner can advance “different constructions” in District Court and before the Board. Inst. Dec. 105. *Samsung* merely permitted a petitioner to use “no formal claim construction” versus some undisclosed specific construction, not polar opposites. *Id.* A petitioner cannot state its belief that a claim means one thing in District Court and simultaneously the reverse in the PTAB, because this necessitates that petitioner is untruthful or advancing a position with which it expressly disagrees. *Facebook, Inc. v. Sound View Innovations, LLC*, IPR2017-00998-13, at 17–18 (PTAB Sept. 5, 2017) (“Petitioner’s failure to inform [the PTAB] of its differing claim construction arguments before the district court raises the specter of lack of candor.”); *Hologic, Inc. v. Enzo Life Scis., Inc.*, IPR2018-00019-21, at 2 (PTAB Nov. 28, 2018) (finding a Petitioner may not “satisfy its obligations under our rules by conducting unpatentability analysis only under a claim construction with which it expressly disagrees”).

(Fed. Cir. 2016) (citing 35 U.S.C. § 311(b)). To the extent this Panel allows Petitioner to take inconsistent positions, the Panel should at least weigh the irreconcilable positions as favoring the application must be remote, because the Federal Circuit has said that “no reasonable fact finder would refuse to consider evidence of inconsistent sworn testimony” and “any such inconsistencies would likely bear on . . . overall credibility.” *Ultratec, Inc. v. CaptionCall LLC*, 872 F.3d 1267, 1273 (Fed. Cir. 2017) (discussing an expert’s credibility, but the statement equally applies to a party).

In contrast, Patent Owner and its expert has maintained that the claims require the recited application to run remotely, usually on a server as required by claims 3, 11, and 17. Ex. 2022 ¶108.

4. A POSITA Would Understand the Claimed Application is Remote from the Wireless Device

Petitioner’s expert admits that the ’715 Patent concerns remotely running applications when discussing a POSITA’s level of skill, explaining that lack of experience or education could be made up through additional study or design of “*remote server applications*.” Ex. 1002 ¶38 (emphasis added); *see also id.* ¶¶46, 52 (providing a “background for the element-by-element analysis of the claims” by discussing where an “*application ran on other computers* and sent the graphics commands to a different computer to be rendered”); *see also* Ex. 2022 ¶109.

Indeed, a POSITA would recognize that the claimed application is remote, as

Patent Owner’s expert explains. Ex. 2022 ¶¶109–11.

* * * * *

The PTAB uses the same claim construction standard as in District Court specifically to avoid construing a patent’s limitations inconsistently across fora. TPG 45–46 (citing 37 C.F.R. § 42.100(b)); Changes to the Claim Construction Standard, 83 Fed. Reg. 51,340, 51,342 (Oct. 11, 2018). Given the claims, specification, party agreement, and expert testimony—and to protect “the integrity of the patent system,” *see* TPG 5—the Board should find that limitations 1[c]–[d], 9[c]–[d], and 17[c]–[d] limit the application to one remote from the wireless device.

D. The Pages of Compiled Content Must Be in Part Generated by the Executing Remote Application

1[d]/9[d]/17[d] wherein said compiled content is generated in part from execution of said application [by said server]
--

Petitioner has proposed that the limitation “said compiled content is generated in part from execution of said application” may be met by a first application (a web browser) executing that causes a second application (a web server) to actually generate compiled content. Pet. 43 (arguing that a browser generates HTML, because in some instances Harris’s MCF is prompted to generate custom HTML upon a browser’s HTTP request). The Board initially agreed. Inst. Dec. 79. This is not correct. Ex. 2022 ¶¶112–20.

A plain and ordinary reading of this limitation is that, when the generating

application is “executing,” the application “produces” or “outputs” (i.e., is the source of), at least in part, the compiled content. Ex. 2022 ¶113. As Patent Owner’s expert explains, a POSITA understands that the term “generate” when discussing an application means: that which is *produced* during execution according to an algorithm, program, or set of rules, Ex. 2022 ¶116 (citing Ex. 2023), and that this data “generated” or produced at a software level is also referred to as *output*, *id.* (citing Ex. 2024). *See also Lyft, Inc. v. Rideshare Displays, Inc.*, IPR2021-01602-7, at 21 (PTAB Apr. 11, 2022) (construing “generated” in accordance with its plain and ordinary meaning of “originated or produced”) (citing <https://www.merriamwebster.com/dictionary/generate>).

The Institution Decision explains Petitioner’s position where “in part from,” in the larger phrase “generated in part from,” is implicitly defined as “upon,” “caused,” or “the basis for,” saying “compiled content is generated *upon* execution of the web browser to make the HTML request,” so “a request from the web browser *caused* the MCF to generate the translation into the ‘device-specific language’ for the requesting device.” Inst. Dec. 79 (emphasis added). While “from” can mean “the starting point,” “the cause of,” or “basis for,” Ex. 2025 (Merriam-Webster.com Dictionary), the claims and specification teach generation “from” execution is referring to another meaning: “the source of,” *see id.* *See also* Ex. 2022 ¶117.

The specification shows “from” means “the source of.” Ex. 2022 ¶114. The

specification says that “[d]uring the execution of an application, the server determines whether the *executed application produces* dynamic or static pages,” Ex. 1001 3:29–31 (emphasis added) and “[t]he server may then determine whether the *executed application produces* dynamic or static content,” *id.* 19:39–40 (emphasis added). This “production” (i.e., sourcing or origination) of pages “during” the executing application’s “execution” on the server corresponds to the limitation language.

Likewise, the “said” in “said compiled content” of limitations 1[d], 9[d], and 17[d] refers to prior limitations where this “compiled content” is pages of the remote application. *See* limitations 1[c], 9[c], and 17[c]. And the specification teaches that the “server tailors the *output of a generic application* based on the wireless device capability,” *id.* 5:53–54 (emphasis added), “the *output is generated from* a generic application,” *id.* 2:25–26 (emphasis added), “[t]he server will eventually translate the *output of the application* to a device specific set of commands *for transmission* to the device,” *id.* 6:9–11 (emphasis added), and the wireless device “[r]eceive[s] compiled content *generated in part from execution of the application by the server*,” *id.* Fig. 6 (item 640). Again, the specification mirrors the claims where the “generated” compiled content is the “output” of (read: “produced by”) the application executing on the server that is transmitted to, and received by, the wireless device. Ex. 2022 ¶¶115, 117.

And even though the specification teaches that client software “executing” on a wireless device “request[s] an application,” Ex. 1001 16:34–37, and, in response, the server “locate[s] and execute[s] the requested application,” *id.* 16:39–42, the specification does not teach the client software “generates” the compiled content as an initiating cause. Ex. 2022 ¶114. Likewise, even though the requesting device type influences what the server application generates (i.e., the server application may tailor the compiled content based upon the wireless device capability that is executing the client software, Ex. 1001 3:42–57), the specification does not teach the client software “generates” the compiled content as an initiating cause. *Id.* Rather, “applications and basic commands for rendering applications *are performed and generated on the server*,” Ex. 1001 4:21–25 (emphasis added), it is still “*the server 130 [that] generates a device specific result from the generic application*,” *id.* 6:34–35 (emphasis added), and it is “*[d]uring application execution pages are generated*,” *id.* 16:61–62 (emphasis added).

Therefore, under the *Phillips* standard, “from” should be understood to indicate the “source,” not “starting time for,” “cause of,” or “basis for” the generated compiled content. *Phillips v. AWH Corp.*, 415 F.3d 1303, 1316 (Fed. Cir. 2005) (en banc) (“[I]nterpretation to be given a term can only be determined and confirmed with a full understanding of what the inventors actually invented and intended to envelop with the claim.”) (quotation and citation omitted). The only construction of

“compiled content is generated in part from execution of said application [by said server]” that “stays true to the claim language and most naturally aligns with the patent’s description of the invention” is that the identified executing application produces/outputs/sources part of the compiled content. *See id.* at 1316 (citation omitted); *see also* Ex. 2022 ¶¶113–17.

Finally, the “in part” language does not change what “generate . . . from” requires. *Id.* 118–19. Patent Owner acknowledges that an application’s generation only need be “in part from execution” to meet this limitation. For instance, the Patent-at-Issue explains that in the case where the application uses dynamic data, part of the compiled content is generated from the template engine:

During the execution of an application, the server determines whether the executed application produces dynamic or static pages. If dynamic data is produced, *a template engine of the server merges dynamic data provided by a business logic unit* and template screens of the requested application provided by executing the application on the server.

Ex. 1001 3:29–34 (emphasis added). That the remote application on the server only generated the compiled content “in part,” as some content was provided by the business logic unit, does not take the application outside the claim scope, because the *executing application also produced some part of the compiled content*. Ex. 2022 ¶120.

To further explain by analogy, Petitioner’s attorneys did not “generate from” their “execution” this Response, even “in part,” because not a single character in this Response was produced by Petitioner’s attorneys. That Petitioner’s attorneys’ execution generated the Petition that *prompted* this Response, or the fact that the content in this Response is *tailored to address the arguments* that were put in the Petition through Petitioner attorneys’ execution, does not change the outcome. In like manner, only an application that actually generates from its execution some compiled content meets the claim limitation, while applications that (1) merely prompt the generating application to produce output during the execution of the generating application, or (2) merely affect the compiled content the generating application produces or outputs during the execution of the generating application, do not.

Per the plain meaning of the claims, as explained by Patent Owner’s expert, and confirmed by the specification, to meet this claim limitation the identified application must produce/output *some part* of the identified compiled content.

E. “Layout Solver” Is Not a Means-Plus-Function Term

In a footnote, Petitioner argues that “layout solver” is a means-plus-function limitation subject to pre-AIA 35 U.S.C. § 112 para. 6. Pet. 69 n.15. Patent Owner disagrees; A POSITA understands that the claim term is used in the pertinent field to designate the class of structures. Ex. 2022 ¶¶28–31.

The absence of the word “means” creates a presumption that the term is not a means-plus-function limitation. *Williamson v. Citrix Online, LLC*, 770 F.3d 1371, 1378 (Fed. Cir. 2014) (en banc). Moreover, “it is sufficient if the claim term is used in common parlance or by persons of skill in the pertinent art to designate structure, even if the term covers a broad class of structures and even if the term identifies the structures by their function” to avoid application of section 112 paragraph 6. *Lighting World, Inc. v. Birchwood Lighting, Inc.*, 382 F.3d 1354, 1359–60 (Fed. Cir. 2004). And “layout solver” is used by POSITAs to designate specific software/hardware. Ex. 2022 ¶125 (citing Ex. 2006, Rule-Based Layout Solving and Its Application to Procedural Interior Generation, 5 (2009) (“The job of the *layout solver*, discussed in the previous section, is to place an object in a layout, making sure the rules defined for that object in the class description hold.”) and Ex. 2007 Adaptive Layout of Dynamic Web Pages, 10, 12–13 (2000) (discussing the difficulty of a general “layout solver,” and previewing the role of a specific web page “layout generator”)). Here, the Patent-at-Issue describes a layout solver as the software or hardware of the server that tailors and translates a template and/or static data representing the visual arrangement of an application based on the device profile and capability. Ex. 1001 3:50–4:1; 16:62–67; 17:54–65; 18:2–6; 19:55–62; Figs. 5 & 7; *see also* Ex. 2022 ¶124.

VI. HARIKI IN VIEW OF HARRIS FAILS TO RENDER ANY CLAIM OBVIOUS

A. No *Single* Application Petitioner Identifies Renders any Independent Claim Obvious

Independent claims recite “an application” once, and every subsequent reference in every claim is to “said application,” making clear there is only one “application.” Ex. 2022 ¶137. But Petitioner points to *two* separate applications as teaching the claimed “application”: a “web server” and a “web client,” because neither individually meets all the claim requirements. And these are two different applications. *Id.* ¶141; *see also* Ex. 2031 21:1–4 (“[M]y general understanding is that web servers and web browsers are commonly implemented as separate components or separate applications.”).

The “web server” does not meet claim limitations 1[b]/9[b]/17[b] because the custom configuration does not configure a plurality of rendering blocks *customized for a* web server. *Id.* ¶142. Similarly, the “web browser” cannot be the claimed “application” because it does not meet limitations 1[b]–[c]/9[b]–[c]/17[b]–[c] as the web browser does not *transmit* the identified compiled content. *Id.* The web browser additionally does not meet these claim limitations, because the web browser is not *remote* from the wireless device and does not *generate* the identified compiled content, HTML pages, even in part. *Id.*

Because Petitioner identifies no single application which meets all the claim

limitations of any independent claim, even if Hariki and Harris are combined as Petitioner alleges, Petitioner fails to show any claim is obvious. *Id.* ¶143.

1. The UI Skin (Identified as the “Custom Configuration”) Does Not Configure a Plurality of Rendering Blocks to Render Content in a Manner Customized to a Web Server (Identified as the “Application”)

1[b]/9[b] wherein said custom configuration . . . configures said plurality of rendering blocks to render content in a manner customized to said application

17[b] . . . custom configuration that configures a plurality of rendering blocks of said wireless device to render content in a manner customized to an application . . .

Petitioner identifies the “converted resource files . . . in Hariki’s UI content package” as “the claimed ‘custom configuration.’” Pet. 30. Petitioner identifies the application as “the web server process [on server 102] that serves HTML files to the mobile device.” *Id.* 36–37. So, assuming, *arguendo*, that the UI Skin of Hariki is analogous to the claimed “custom configuration” and the web server of Hariki is analogous to the claimed “application program” (which Patent Owner does not concede), as set forth in the claim construction section, this means the UI content package must be created for a web server process. *See* Section V.B beginning on 16. But these UI skin resource files are not designed with a web server process in mind. Ex. 2022 ¶¶141–42.

Indeed, the process of creating a UI content package starts with the selection and retrieval of a “resource profile.” *See* Ex. 1006 Fig. 4 (item 401). Hariki teaches that “[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*.” *Id.* [0019] (emphasis added). That is, the “resource files [are] *for the different mobile devices*.” Ex. 1006 [0025] (emphasis added). And that it is “an application program *executed on the mobile handset* . . . [that] access[es] the resources,” *id.*[0019] (emphasis added). Therefore, the UI skin is—if anything—written and created for applications running on mobile devices, not a web server. *See also* Ex. 1002 ¶135 (“Hariki . . . discloses sending content ***specific to a mobile device***.”) (different emphasis).

Hariki does not contemplate that the web server application is running or rendering *anything* on the wireless devices, so there are no rendering blocks on the wireless device that the UI skin can configure in a manner customized to the web server. *Id.* ¶151. A web server sends the same HTML in response to the same request, regardless of whether the requestor is using a UI skin, and, if so, what UI skin is in use, and, vice versa, the device processes the same HTML the same way, regardless of what web server sent the HTML. Ex. 2022 ¶146. Hariki and Petitioner do not say otherwise. In this case, as Netflix’s expert noted, “standard resources that are the same across everywhere, that perhaps that wouldn’t be considered to be customized.” Ex. 2031 31:21–31:2. This is why Petitioner conflates the server with the browser, referring to the “web server-web browser *interaction*,” Pet. 37

(emphasis added), trying to impute the browser's rendering to the server.

Indeed, Petitioner's expert was able to identify items that he believed were aware of and customized to a web browser in Petitioner identified compiled content, Ex. 2031 68:7–69:4, items that he believed were aware of and customized to a web browser in Petitioner identified custom configuration, *id.* 69:3–70:15, but was unable to point to any item that was aware of and customized to a web server, *id.* 70:17–71:6.

Instead, Petitioner argues that the web server meets this limitation because “POSITAs would have recognized the user interface authoring tool and web server process were part of the same suite of programs because they [1] reside on the server together and [2] that the authoring tool generated the custom UI for mobile devices accessing the web server process.” Pet. 37.

For [1], not all programs residing on the same computer have to be part of a suite, i.e., “a set of computer programs designed to work together and usually sold as a single unit.” Ex. 2022 ¶144 (citing Ex. 2004 (<https://www.merriam-webster.com/-dictionary/-suite>)). Indeed, Petitioner does not argue that the web server and the user interface authoring tool interact in any way. And even a cursory review of Hariki's Figure 2, the “functional diagram of a user interface authoring tool,” Ex. 1006 [0012], [0028], shows no such interaction. Further, Petitioner's expert admits “there's a content server and a web server, and they might be separate”

though they “could run together on the same hardware.” Ex. 2031 64:1–5.

Hariki merely says that 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,” Ex. 1006 [0027]. Hariki’s teaching simply means multiple programs or circuitry may perform the user interface authoring tool functionality, Ex. 2022 ¶145, not that a web server is part of a user interface authoring tool “suite.”

For [2], Petitioner does not explain why, even if the authoring tool provided a “UI content package” to a remote device, and that remote device had a web browser that interfaced with the UI content package and displayed content from a web server running on the same server as the authoring tool, the UI content package would customize rendering blocks for, this particular web server. Hariki says that “the mobile devices run[] a web browser program to access the web pages [not only] served by server computer 102 [but] any other available content provider or supplemental server, such as computer 106.” *Id.* [0021]. Indeed, while the content provider knows what mobile device a content package is destined for, so may custom the UI skin for that device, Ex. 2031 62:6–11 (“Well, when a content provider is creating content packages embodying UI skins, I think they would at least need to know information about the characteristics of a device or device type that they were creating a UI skin for . . .”), the provider has no idea what web servers the user will

visit, so there is no way to design a UI skin for “characteristics” of unknown web servers. Ex. 2022 ¶146. In fact, Petitioner’s expert explained that he did not consider how a general website would interact with a UI skin. Ex. 2031 64:14–65:8 (saying that “[i]f a person using a mobile device visits some other server that is not part of the system”—i.e., any web server not specifically listed in Hariki (such as yahoo.com)—these web servers are likely not “relevant” as they are “just using a browser and a server . . . They’re not using the Harris-Hariki system that I’m describing.”). There is simply nothing that makes downloading web content from a web server that happens to be co-resident on the server running the authoring tools special such that the “UI content package” on the wireless device is made with this web server in mind any more than any other web server serving the same or different content. Ex. 2022 ¶146.

In sum, Hariki’s “UI skin” on a wireless device (i.e., the Petitioner-identified “custom configuration”) does not configure a plurality of rendering blocks in a manner customized for the “web server” (the Petitioner-identified “application”). As such, Hariki’s “web server” does not meet limitations 1[b], 9[b], or 17[b].

While the Board has issued a preliminary decision that Hariki’s “web browser” does meet these claim limitations that are missing with respect to the “web server,” Inst. Dec. 60–65, the web browser application is missing at least three other limitations as now discussed.

2. The Web Browser Does Not Transmit HTML Files to a Wireless Device

1[c]/9[c]/17[c] [transmitting, to said wireless device, / . . . transmits compiled content to said wireless device, said] compiled content comprising (i) first compiled content specific to a first page of said application and (ii) second compiled content specific to a second page of said application

For this limitation, Petitioner identifies a second claimed application, “the web browser” running on the wireless devices, Pet. 36; however, the web browser does not “transmit[] compiled content to said wireless device.” Petitioner identifies the “compiled content” as “HTML files” from “World-Wide Web (WWW) server [102].” Pet. 38–39 (quoting Ex. 1006 [0021]). Hariki teaches it is the “*server . . . [that] **transmits** these pages as Hypertext Markup Language (HTML) files over the Internet 110 **to** the mobile devices 108 and 109.*” Ex. 1006 [0021]. Petitioner has the same understanding, stating that the “HTML files [are] *sent by a web server.*” Pet. 38 (emphasis added). HTML files are not sent by the web browser which is “a client [that] *receives* multiple HTML pages *from* a server.” *Id.* 39; *accord* Ex. 2022 ¶¶147–49; *see also* Pet. 39 (“*[T]he server transmits . . . HTML files (corresponding to the claimed ‘compiled content’).*”) (emphasis added); *id.* 40 (“*Hariki discloses sending HTML files **to** a mobile device.*”) (emphasis added). Indeed, Petitioner’s expert could only say “that it is *conceivably possible* that an HTTP request could include some HTML,” Ex. 2031 22:5–7 (emphasis added), not that Hariki’s browser ever transmits two HTML pages, *id.* 41:8–13 (testifying that

he is unable to recall any reliance upon “Hariki’s browser sending or transmitting HTML files to a web server”). Therefore, Hariki’s “web browser” does not transmit the identified compiled content.

Harris fails to disclose that which Hariki lacks. Specifically, with respect to Harris, Petitioner still maintains that “POSITAs would have been motivated to find ways of generating and *sending* information *from applications* such as a *web server process* . . . and looked to *Harris* because, . . . *Hariki* discloses *sending* HTML files *to* a mobile device . . .” *Id.* (emphasis added); *see also id.* 42 (“Harris contemplates serving multiple pages (e.g., HTML files) *to* the mobile client.”) (emphasis added). Therefore, even with Harris, the “web browser” still does not transmit the identified compiled content. *See also* Ex. 2031 41:3–7 (“Q: So after combining Harris with Hariki, in your combination, it’s still a server that transmits HTML files to a mobile device; is that correct? A: I think that’s right.”); 43:1–5 (same); 44:13–17 (“Q: And you haven’t identified the wireless device transmitting HTML files to the server; is that correct? A: I think that’s right. I did not point to that specifically.”)

The Institution Decision only serves to highlight that the web browser may “request” compiled content, but it is another application, the web server, that “transmits” the requested compiled content. Inst. Dec. 72–73 (“Regarding ‘transmitting’ according to limitation 1c, when a *web browser on a mobile device requests* a first web page, *Hariki’s server responds by transmitting* a first HTML

file corresponding to ‘first compiled content specific to a first page of said application.’”). The Board wrote that “[a]n HTML file *transmitted by a server to a web browser* on a requesting device is ‘specific to’ a ‘page’ of the web browser,” *Id.* 73, but that does not change that the HTML file is not transmitted by the web browser as required (if the web browser is identified as the claimed application). Ex. 2022 ¶151.

This is a clear missing limitation: the web browser does not transmit a first and second HTML page.

3. The Web Browser Is Not Running Remotely from the Wireless Device

1[c]/9[c]/17[c] [transmitting, to said wireless device, / . . . transmits compiled content to said wireless device, said] compiled content comprising (i) first compiled content specific to a first page of said application and (ii) second compiled content specific to a second page of said application

1[d]/9[d]/17[d] wherein said compiled content is generated in part from execution of said application [by said server]

For these limitations, Petitioner again exhorts that the claimed application is “the web browser,” Pet. 36; however, the web browser resides on the wireless device. Therefore, while it does not transmit HTML files at all, as just described, it doubly does not transmit HTML *remotely* from the wireless device as required. *See* Section V.B beginning on page 15. Hariki’s “web browser” is a *locally* running application on the wireless device. Ex. 2022 ¶152; Ex. 1002 ¶60 (“Web browsers are client applications.”); Ex. 2031 16:15–19 (“So, in this case, I think that web

browser running on a mobile device, in this case, the one disclosed in Hariki, might be considered by a POSITA to be a client -- client application.”); 18:1–4 (“[I]f you have a standalone web browser running on a client, then it probably—I think it would be reasonable to consider it a client application.”).

Indeed, Petitioner implicitly admits that if the claims require the application be remote from the wireless device—as it does—that the web browser is insufficient. *See also* Ex. 2031 60:18–22 (“If the browser is on the mobile—if it’s running on the mobile device, then I think it probably is not remote to the mobile device.”). The Petitioner says in this scenario it relies upon the web server application. Pet. 36–37 (“To the extent PO contends that the ‘associated’ application must be remote to the mobile device . . . the server also provides content data such as HTML files through a *web server process*.”) (emphasis added); *see also* Pet. 66 (arguing only that the “web server” and “authoring tool,” not a “web browser,” meet limitation 17[pre], requiring “a server . . . generate content”).

Therefore, in addition to not transmitting HTML, a “web browser” cannot be the claimed application because it is not remote. Ex. 2022 ¶152.

4. The HTML Files are Not Generated in Part from the Execution of the Web Browser

1[d]/9[d]/17[d] wherein said compiled content is generated in part from execution of said application [by said server]
--

For this limitation, Petitioner also advances that the claimed application is

“the web browser” running on the wireless devices, Pet. 36, but the web browser does not—even in part—“generate . . . from [its] execution” the HTML files that Petitioner identified as the Claims’ “compiled content.” As explained in the claim construction section, see Section V.D beginning on page 28, this limitation requires the identified application to “produce” or “output” the compiled content during its execution. Yet, Petitioner concedes that “*Hariki* . . . does not expressly disclose how the HTML files are generated.” Pet. 40; *see also* Ex. 2031 45:7–11 (“So it’s true, then, that *Hariki* doesn’t expressly disclose how HTML files are generated, correct? A: That’s what I wrote. I think that’s right.”). And Petitioner agrees that with the *Hariki*-Harris combination, it is only Harris’ MCF that “generates” HTML files from WAX.⁸ Pet. 40 (“POSITAs would have been motivated to find ways of *generating* and sending information from applications *such as a web server process*.”) (emphasis added); Ex. 2022 ¶¶153–54; *see also* Ex. 2031 45:12–14 (“Q: And does the Harris reference disclose that browsers generate HTML? A: I don’t recall it doing so.”); 58:7–9 (“Q: Do you remember Harris discussing a browser producing

⁸ For claim 17[c], dealing with a layout solver transmitting compiled content, Petitioner said that “the browser of *Hariki* processes the HTML to *generate* and layout the graphical user interface that is created for display,” Pet. 71–72 (emphasis added), but this is generation of graphics on a GUI, not the generation *of the HTML* as the limitations require. Ex. 2022 ¶163.

HTML? A: No, I do not.”).

As Harris explains, the “Web Server Accepts [a] Device Request” for content, the “Content [is] **Generated**” on the MCF, and the generated “Content [is] Returned to [the] Device.” Ex. 1007, Fig. 3 (especially items 60, 68, & 76) (emphasis added). Therefore, the HTML files are *completely* produced and output by the MCF, and the MCF transmits the finished generated HTML to the wireless devices. *Accord* Pet. 42 (“Harris explains that ‘the framework [MCF] translates the WAX into a device-specific markup language (WML, HTML, etc.).’”) (citing Ex. 1007 [0027]).

Not a single character in the HTML files is produced or output by the web browser (the Petitioner-identified “application”), whose role is limited to *requesting* and *displaying* content. Ex. 2022 ¶153. As such, the web browsers on the wireless devices of Hariki-Harris do not generate (under a meaning that stays true to the Patent-at-Issue’s teaching) *any* of the identified compiled content, i.e., the Petitioner-identified HTML files. *Id.* ¶155.

Petitioner’s argument that a browser supposedly “generates” HTML “in part from [its] execution,” because in some instances Harris’s MCF may be prompted to translate WAX to HTML based upon the browser type found in the browser’s “HTTP request,” Pet. 43, relies upon the construction of “from” as “being caused by,” “starting at,” or “as a function of.” But as already explained, such a claim construction is incorrect under *Phillips* in light of the specification that teaches

“from” means “the source of” (i.e., what application produced or output the compiled content during the identified application’s execution). *See* Section V.D beginning on page 28; *see also* Ex. 2022 ¶154.

It is solely the MCF that generates—produces or outputs—the HTML, during the MCF application’s execution, regardless of what prompted the MCF to action, or what HTML the MCF generates. Pet. 43–44 (“[T]he **[mobile content] framework** ‘translates the WAX into a device-specific markup language (. . . HTML . . .)’”) (citing Ex. 1007 [0027]) (emphasis added); *see also* Ex. 2031 50:14–18 (“So do you stand by that statement that **the template generation process begins** when the web server accepts the HTTP request from a mobile device? A: I think that’s right.”) (emphasis added); 51:14–18 (“[T]he MCF **does not start generating** until it receives the request? A: I think that’s primarily what Harris discloses.”); 51:19–53:17 (explaining that generally that it “a web server is running some code to dynamically generate a response such as HTML” and “generation would happen with [web server] code and that storage would happen with [web server] code” before the web server “code would be used to transmit that stored data such as HTML, to the requester”).

Hariki-Harris’ web browser is at best analogous to the “client” in the Patent-at-Issue. Specifically, the web browser no more “generates” HTML than the client in the Patent-at-Issue that also “requests an application,” Ex. 1001 2:66–3:1, that

prompts “[t]he requested application [to be] executed on the server” “[i]n response,” *id.* 3:4–9, whose device on which the client is executing influences the compiled content generated, *id.* 18:9–11 (“[T]he parameters of the basic commands are tailored based on the wireless device capabilities”), but *does not* “generate” the application—it is still the server that “produces dynamic or static pages” “during execution of an application [on] the server,” *id.* 3:29–31.

The HTML files are simply *not* generated in part from execution of the web browser, so the web browser does not meet this claim limitation for this third reason.

* * * * *

Given that the UI skin (Petitioner identifies as the “custom configuration”) does not configure a plurality of rendering blocks to render content in a manner customized to, a web server (Petitioner identifies as the “application”), ***the web server application does not meet the claim requirements found in every claim.*** And because the web browser (Petitioner alternatively identifies as the “application”) is not running remotely and does not transmit or generate, even in part, HTML pages (Petitioner identified “compiled content”), ***the web browser application also does not meet the claim requirements found in every claim.*** Therefore, the Petitioner has not identified a single application meeting all the claim requirements, and, for this reason alone, has not shown any claim of the ’715 Patent unpatentable by a preponderance of the evidence.

B. Hariki's UI Skin Is Not "Applicable" to the Received HTML Files

Separate from the fact that there is no one application that meets all the claim limitations, the claims also require the identified "custom configuration" (i.e., Hariki's UI skin) be "applicable" to the identified "compiled content" (i.e., HTML files sent by Harris' MCF). As now described, Hariki's UI skin is not applicable to the HTML files.

1[f]/9[f]/17[f] wherein said custom configuration is applicable to said first and second compiled content

Petitioner identifies the "converted resource files . . . in Hariki's UI content package" as "the claimed 'custom configuration,'" Pet. 30, and says the UI skin operates on "rendering blocks" such as the device's "backgrounds, title bars, buttons, and alert sounds," *id.* 47 (quoting Ex. 1006 [0022]). Petitioner identifies the "compiled content" as "HTML files" from "World-Wide Web (WWW) server [102]." Pet. 38–39 (quoting Ex. 1006 [0021]). However, the UI skin is not "applicable to" the HTML files through the UI skin configured backgrounds, title bars, buttons, and alert sounds as required.

Petitioner only argues that the UI skin may be used by a browser, and then makes the logical leap that the browser's use of 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 the

application.” Pet. 47–48 (citing Ex. 1002 ¶148). The Board likewise followed this unsupported position, that “customization to the web browser program” (i.e., “application-level customization”) is necessarily “applicable to” HTML rendered in a browser. Inst. Dec. 85–86. However, just because a *browser* may access resource files does not mean that resource files are applicable to *the HTML files* within the browser.

Indeed, as explained by Patent Owner’s expert, HTML is a standard that specifies how to display, but has no provision for interfacing with the UI skin accessed through Hariki-specific local API. Ex. 2022 ¶¶156–57.

The suggestion that the HTML uses the UI skin is untenable given Harris’ disclosure that “an automated testing environment will make sure content will display[] correctly *before* they are deployed to a device.”⁹ Ex. 1007 [0035] (emphasis added). If the HTML content displays correctly in the testing environment on the MCF, where there are no UI skin resources—that are on remote

⁹ There appears to have been some miscommunication, evidenced by the Board saying that “Patent Owner mischaracterizes Harris’s disclosure” because “Harris does not indicate that automated testing involves deployment to a device where ‘the UI skin resources are accessed through the resource API.’” Inst. Dec. 90 (citation omitted). Patent Owner did not, and does not, take the stated position. Harris teaches the opposite: that the automated testing is done *remotely and before* deploying the HTML to the device.

devices—this means the Hariki-Harris HTML does *not* access, reference, or use any UI skin resources. The Board correctly notes that this testing is optional, Inst. Dec. 90, but this does not change the logic: if such “correct display” *can be* done correctly without access to UI skin resources, this means the HTML files do not use the UI skin resources. Put another way, if the HTML files *did* rely upon the UI skin resources to present properly, the remote test environment *could not—even optionally*—“correctly display” the content prior to deployment. Ex. 2022 ¶158.

Indeed, as Patent Owner’s expert explains, it is HTML that sets how the page is displayed, not UI skin resources. Ex. 2022 ¶156. Petitioner acknowledges the same. Pet. 11 (explaining that it is HTML that “describ[es] the visual presentation”), 38 (“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* [sic].”) (emphasis added).

Given that the HTML may optionally be properly displayed prior to deployment, that the HTML determines the look and feel of the web pages, and that Petitioner only points to Hariki’s disclosures that the UI skin is “applicable to” the web browser, not that the UI skin is applicable to HTML within the browser, Petitioner fails to show Hariki, alone or in combination with Harris, meet these limitations.

Separately from the fact that there is no one application that meets all the claim limitations, the Hariki–Harris combination also does not teach that the UI skin

(Petitioner identified custom configuration) is applicable to the HTML pages (Petitioner identified first and second compiled content). As this limitation is found in all the claims, Petitioner again has not shown any claim unpatentable by a preponderance of the evidence.

C. Hariki in View of Harris Additionally Does Not Teach the “Library” and “Layout Solver” Limitations of Claims 17–20

17[a] a library of applications;

17[b] a library of custom configuration data comprising a custom configuration that configures a plurality of rendering blocks of said wireless device to render content in a manner customized to an application from said library of applications requested by said wireless device

For limitation 17[b], Petitioner says “[t]he only difference between [17b] and [1a] and [1b] is that [17b] recites ‘a library of custom configuration data’ on the server,” Pet. 68, and then identifies configuration data from the resource profile database 204 and common resource depot 202 that are used to create the UI skins that is deployed to the wireless devices as “a library of custom configuration data,” *id.* 68–69. But the library of custom configuration data is not the only difference; Limitation 17[b] also requires that the “custom configuration data” must configure rendering blocks on the wireless device to customize an application *from the previously recited “library of applications”* in limitation 17[a].

And for limitation 17[a] Petitioner identified “the user interface authoring tool 104” as the “library of applications,” “because the user interface authoring tool

would be a “suite of programs” and that a “suite of programs [is] synonymous with a ‘library of applications.’” Pet. 67 (citing Ex. 1002 ¶94). Petitioner also previously argued that the “web server” was part of the alleged “suite of programs,” Pet. 37, so too would be an application in this newly identified library.¹⁰

Patent Owner has already shown why the identified custom configuration data (i.e., the resource files in the UI skin) do not configure a plurality of rendering blocks of said wireless device to render content in a manner customized to the identified application (i.e., a web server) and that, therefore, this limitation is not met. *See* Section VI.A.1 beginning on page 36; *see also* Ex. 2022 ¶160. Additionally, Patent Owner has already explained that the web server is not an application in a “library” or “suite” of programs just because (1) the web server and authoring tool reside on the same computer, or (2) that the mobile device, using the custom UI developed with the authoring tool, accesses the web server. *Id.* Thus, the web server does not meet the limitation for this additional reason.

To the extent that Petitioner is arguing the resource files in the UI skins on the

¹⁰ In contrast, Petitioner did not argue that a “web browser” was part of the authoring tool “suite of programs” for 1[b], so is not an “application” at issue for claim 17. Even so, as shown in Sections VI.A.2–4, the web browser is not *remote*—much less run on a server—and does not *transmit* or *generate* any part of the identified compiled content, HTML files, as also required by claim 17. Ex. 2022 ¶161.

wireless devices configures rendering blocks *for the user interface authoring tool*, this is incorrect. Petitioner previously admitted that the identified custom configuration (i.e., the “UI skin”) configured the “background, title bars, buttons, and alert sounds that comprise ‘rendering blocks’” *on the mobile device*. Pet. 31. But, like the web server, the user interface authoring tool does not run on the mobile device where these “rendering blocks of said wireless device to render content” reside, see Ex. 1006, Fig. 1, so Petitioner’s identified rendering blocks are not, and could not, be used to render the UI authoring tool 104 executing on separate server 102. Ex. 2022 ¶159.

Therefore, Petitioner has not shown the common resource depot 202 (Petitioner identified “library of custom configuration data comprising a custom configuration”) configures a plurality of rendering blocks of said wireless device to render content in a manner customized to a web server or UI skin authoring tool (Petitioner identified applications from said library of applications requested by said wireless device).

D. Hariki in View of Harris Does Not Teach the “Layout Solver” Limitation

17[c] a layout solver that transmits compiled content to said wireless device, said compiled content comprising (i) first compiled content specific to a first page of said application and (ii) second compiled content specific to a second page of said application

Petitioner initially argues that Hariki in combination with Harris discloses this

“feature” for the same reasons discussed in limitation 1c.¹¹ *Id.* Pet. 68–69.

Because Petitioner points to a different “application” for Claim 17[a]–[b] not discussed in limitation 1[c], the user interface authoring tool, Pet. 67, Petitioner cannot simply refer to limitation 1[c]. In any case, Hariki teaches that “the user interface authoring tool 104 [is] executed by server computer 102,” Ex. 1006 [0027], and Hariki does not indicate, and Petitioner does not argue, that (1) any UI authoring tool pages are transmitted, or (2) that the UI content packages sent to the wireless devices are “applications.” So, Hariki does not transmit anything that could be called pages of the authoring tool. Ex. 2022 ¶162.

Then Petitioner concedes that limitation 17[c] is different than 1[c], because 17[c] requires compiled content pages must be sent by a “layout solver,” and Petitioner appears to identify the web browser as a layout solver, because, according to Petitioner, it is “the browser of Hariki [that] processes the HTML to generate and *layout* the graphical user interface that is created for display.” Pet. 71–72 (emphasis added). Regardless of any “laying out” the browser performs, as explained in Section VI.A.2 beginning on page 41, the limitation requires the layout solver to *transmit* the identified compiled content (i.e., HTML files) to the wireless

¹¹ Petitioner says “Sections VIII(A)(1)(b)–(c)” referring to limitations 1[a]–[b], Pet. 70, but this appears to be an error. To the extent it is not, limitations 1[a]–[b] only invoke Hariki, so cannot support 17[c] where Petitioner also relies upon Harris.

devices, and the browser does not transmit HTML anywhere; the browser receives it.¹² Ex. 2022 ¶163.

The Petitioner concludes by saying that “*Hariki* in combination with *Harris* discloses a ‘layout solver’ for substantially the same reasons discussed in Sections VIII.A.1.c and VIII.A.14,” referring to limitations 1b and claim 14. *id.* 72. This logic fails, as limitation 1b only refers to *Hariki*, and both limitation 1b and claim 14 concern the custom configuration of rendering blocks, and the syntax of the commands therefor, that do not appear relevant.

In sum, Petitioner has additionally not shown the *Hariki-Harris* combination discloses a web browser (Petitioner identified “layout solver”) transmits compiled content to said wireless device, said compiled content comprising pages of HTML (Petitioner identified first and second compiled content specific to a first and second page of said application).

¹² Petitioner’s contention that the browser “processes . . . to generate the graphical user interface”—i.e., causes rendering on the screen, is equally irrelevant, given that the preamble requires “[a] server that is programmed to *generate*,” not an application running on the wireless devices like the web browser “to generate.” The preamble also requires that the server generate “content that *is* renderable” (i.e., generation *of* the identified HTML) not merely rendering *of* renderable content by a browser (i.e., generation of images on a GUI by processing the identified HTML).

E. Harris and Hariki Further Do not Teach Limitations in Dependent Claim 3 and 11

3/11 [A method as described in claim 1/A non-transitory computer readable medium as described in claim 9] wherein said compiled content is partially resultant from said application operating on a remote server.

Claims 3 and 11 further limit the independent claims 1 and 9, such that the application executing remotely “must operate on a remote *server*.” As discussed for independent claims 1 and 9 above, see Section VI.A.3 beginning on page 43, the web browser does not run remotely, much less specifically on a server. Ex. 2022 ¶164.

Therefore, Petitioner has additionally not shown HTML files (Petitioner identified “compiled content”) is partially resultant from a web browser (Petitioner alternatively identified “application”) operating on a remote server.

VII. OBJECTIVE INDICIA OF NONOBVIOUSNESS

This is a case where the Board must consider the fourth Ghramm factor, the so called “secondary considerations.” *Apple Inc. v. Samsung Elecs. Co.*, 839 F.3d 1034, 1048 (Fed. Cir. 2016) (en banc) (“Objective indicia of nonobviousness must be considered in every case where present.”). Here the patented invention, as implemented in the Netflix system, has been a commercial success due in part to the “essential” patented technology.

A jury has determined that the accused Netflix system infringes claim 4 of the ’715 Patent, and by necessity, claim 1. Ex. 3003. And claim 9 is the “non-transitory

computer readable medium comprising instructions therein that when executed by a processor implement a method” claimed in claim 1. Therefore, the Netflix system practices the ’715 Patent.

And, as demonstrated in the January 26, 2023 10-K statement filed by Netflix with the Securities and Exchange Commission, the accused Netflix system has been very successful, with over 200 million subscribers paying an average of over \$11 per month, bringing in streaming revenues of over \$30 billion worldwide:

Results of Operations				
The following represents our consolidated performance highlights:				
	As of/ Year Ended December 31,			Change
	2022	2021	2020	2022 vs. 2021
(in thousands, except revenue per membership and percentages)				
Financial Results:				
Streaming revenues	\$ 31,469,852	\$ 29,515,496	\$ 24,756,675	7 %
DVD revenues	145,698	182,348	239,381	(20) %
Total revenues	\$ 31,615,550	\$ 29,697,844	\$ 24,996,056	6 %
Operating income	\$ 5,632,831	\$ 6,194,509	\$ 4,585,289	(9) %
Operating margin	18 %	21 %	18 %	
Global Streaming Memberships:				
Paid net membership additions	8,903	18,181	36,573	(51) %
Paid memberships at end of period	230,747	221,844	203,663	4 %
Average paying memberships	222,924	210,784	189,083	6 %
Average monthly revenue per paying membership	\$ 11.76	\$ 11.67	\$ 10.91	1 %

Ex. 2028, at 3.

This commercial success of the Netflix system has a nexus to benefits conferred by the claimed features of the ’715 Patent. As confirmed by Netflix’s employees, Netflix’s commercial success is derived from the Netflix application’s use of the patented invention to selectively, based on which type of device the user is using, “deliver only the data needed to render the view in as few requests as possible, which is essential for a quality mobile experience.” Ex. 2029 (Berglund, Netflix UI Engineer); *see also* Ex. 2030, (Kao, Netflix UI Engineer, Dep. Tr. p.3 at

125:2–8 (agreeing with Berglund)).

Given that the '715 Patented technology is “essential” to Netflix’s undisputed commercial success there is objective indicia, with a nexus to the claimed invention, that additionally supports the non-obviousness of the claims of the '715 Patent.

VIII. CONCLUSION

Hariki’s resource files in the “UI skin” that Petitioner labels the claimed “custom configuration” are resident on wireless devices and affect applications running on the wireless devices, so the UI skin does not configure a plurality of rendering blocks of said wireless device to render content in a manner customized *to a web server or user interface authoring tool* running remotely. And Hariki’s *web browser* running on the wireless device does not transmit HTML to the wireless device, is not running remotely, and does not generate any of the HTML. As neither application meets all the application related claim requirements, even when considering the prior art references as Petitioner combines them, Petitioner has not shown that any of the independent claims are rendered obvious. The dependent claims are not obvious for at least the same reasons.

Petitioner also has not shown that the UI skin is *applicable to* the Petitioner identified compiled content (i.e., HTML files) in the Hariki-Harris combination. This limitation is also found in all the claims, so, all the claims are not rendered obvious for this additional reason.

Petitioner has failed to show Hariki-Harris discloses the “library” and “layout solver” limitations of claim 17, so Petitioner fails to show claim 17 and its dependent claims 18–20, are obvious for these further reasons.

Finally, Petitioner has not shown a web browser runs on a server as required by dependent claims 3 and 11, so not shown these dependent claims obvious for this additional reason.

Patent Owner requests that the Board issue a final written decision finding Petitioner has failed to show any claim obvious by a preponderance of the evidence.

Dated: January 30, 2024

Respectfully submitted,

/Joshua S. Wyde/

Joshua S. Wyde (Reg. No. 57,698)

ALAVI & ANAIPAKOS PLLC
609 Main Street, Suite 3200
Houston, Texas 77002
Tel: (713) 751–2365
Fax: (713) 751–2341
jwyde@aatriallaw.com

Counsel for Patent Owner,
GOTV STREAMING, LLC

CERTIFICATE OF WORD COUNT

Per § 42.24(a)–(b), the undersigned certifies that the number of words in the paper, PATENT OWNER’S RESPONSE, is:

13,161

The word count does not include the table of contents, table of authorities, certificate of service, word count, exhibits list, or claim listing.

To arrive at this count, the undersigned has relied upon the word count feature of the word-processing system used to prepare the paper.

Dated: January 30, 2024

/Joshua S. Wyde/
Joshua S. Wyde (Reg. No. 57,698)

CERTIFICATE OF SERVICE

The undersigned certifies that on the thirtieth day of January, 2024, a complete and entire copy of the foregoing “PATENT OWNER’S RESPONSE,” including exhibits, if any, was served on the date below on the following counsel of record shown below via email per Petitioner’s consent to electronic service, see Paper 2 at 79:

Petitioner’s distribution list
Aliza George Carrano
Indranil Mukerji
J. Christopher Moulder

Netflix-GTS-WFG@willkie.com
acarrano@willkie.com
imukerji@willkie.com
cmoulder@willkie.com

Dated: January 30, 2024

/Joshua S. Wyde/
Joshua S. Wyde (Reg. No. 57,698)

Counsel for Patent Owner,
GOTV STREAMING, LLC