

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

ATLASSIAN CORP. PLC and ATLASSIAN, INC.,
Petitioner,

v.

EXPRESS MOBILE, INC.,
Patent Owner.

IPR2021-01471
Patent 9,063,755 B2

Before JEFFREY S. SMITH, AARON W. MOORE, and
RUSSELL E. CASS, *Administrative Patent Judges*.

MOORE, *Administrative Patent Judge*.

DECISION
Denying Institution of *Inter Partes* Review
35 U.S.C. § 314

I. INTRODUCTION

Atlassian Corp. PLC and Atlassian, Inc. (collectively, “Petitioner”) filed a Petition (Paper 1, “Pet.”) requesting *inter partes* review of claims 23–25 of U.S. Patent No. 9,063,755 B2 (the “’755 patent,” Ex. 1001). Express Mobile, Inc. (“Patent Owner”) filed a Preliminary Response (Paper 6, “Prelim. Resp.”). Under 35 U.S.C. § 314 and 37 C.F.R. § 42.4(a), we have authority to determine whether to institute review.

Institution of an *inter partes* review is authorized when “the information presented in the petition . . . and any response . . . shows that there is a reasonable likelihood that the petitioner would prevail with respect to at least 1 of the claims challenged in the petition.” 35 U.S.C. § 314(a).

Having considered the Petition and the evidence of record, we conclude there is not a reasonable likelihood that Petitioner will prevail in establishing the unpatentability of at least one claim of the ’755 patent and, therefore, do not institute *inter partes* review.

A. *Related Proceedings*

The parties state that the ’755 patent is the subject of several district court proceedings. *See* Pet. 1–3; Paper 4, 1–2.

We denied petitions for *inter partes* review of the ’755 patent in IPR2021-00709, IPR2021-01144, and IPR2021-01228, and instituted an *inter partes* review in IPR2021-01455. Another petition for *inter partes* review of this patent, IPR2022-00249, is pending.

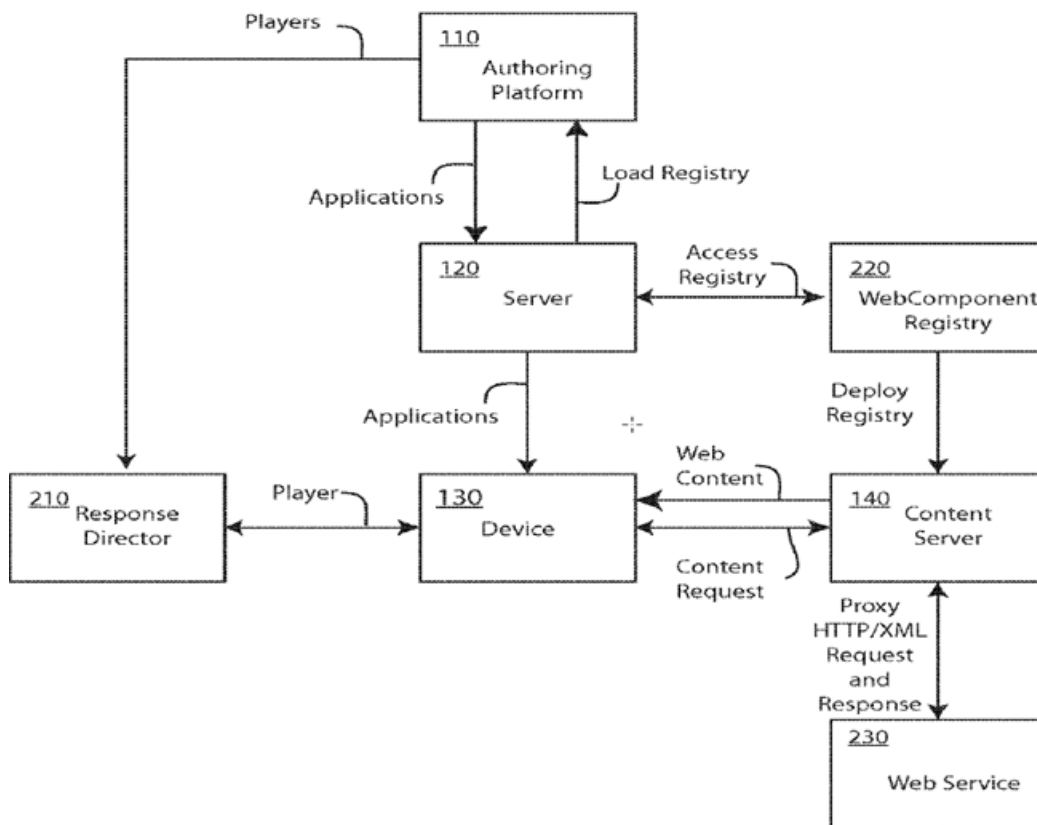
B. *The ’755 Patent*

The patent relates to a platform for authoring software code for mobile devices. *See* Ex. 1001, 1:8–10. A user of the authoring platform provides instructions for a mobile device in the form of device-specific

instructions, referred to as a Player, and device-independent instructions, referred to as an Application. *See id.* at 5:8–15. The authoring tool can produce a plurality of Players for different devices, and a plurality of Applications for displaying pages on the devices. *See id.* at 5:42–49.

The Player transforms device-independent instructions of the Application into device-specific instructions that are executable by the device. *See Ex. 1001*, 5:56–58. Thus, the authoring tool can be used to design device-independent Applications that specific devices then use to generate displays from the Applications. *See id.* at 6:12–17.

Figure 2A is reproduced below.



“FIG. 2A is a schematic of an embodiment of [a] system illustrating the communications between different system components.” *Ex. 1001*, 2:21–23.

Authoring platform 110 generates one or more Players that are provided to response director 210. *See* Ex. 1001, 8:7–9. Device 130 requests a Player from response director 210, and receives and installs the Player. *See id.* at 8:9–11. Web service 230 is a plurality of services obtainable over the Internet. *See id.* at 8:18–19. Each web service is identified in an entry in web component registry 220. *See id.* at 8:19–22. Web component registry 220 is provided through server 120 to authoring platform 110 so that a user of the authoring platform may bind web services 230 to elements to be displayed on device 130. *See id.* at 8:22–26. A web component registry 220 for each registered web service 230 is loaded into authoring platform 110. *See id.* at 8:54–56. The user of the authoring platform can then assign components of any web service 230 to an Application without any need to write code. *See id.* at 8:56–58:8.

C. Representative Claim

The only independent claim that is challenged in this proceeding is claim 23, which is reproduced below.

23. A method of providing information to a device having a display from a web component of a web service to a device on a network, said method comprising:

- accepting, on the device, a first code over the network, where said first code is device-dependent;

- accepting, on the device, a second code over the network, where said second code is device-independent and includes a plurality of symbolic names of inputs and outputs associated with the web service; and

- executing said first code on the device,

where the symbolic names are provided from a registry of one or more web components related to inputs and outputs of a web service obtainable over a network,

where the web service requires both an input symbolic name and one or more associated input values and returns one or more output values having an associated output symbolic name, and

where the registry includes

- a) symbolic names required for evoking one or more web components each related to a set of inputs and outputs of a web service obtainable over a network, where the symbolic names are character strings that do not contain either a persistent address or pointer to an output value accessible to the web service, and
- b) the address of the web service;

where said executing includes:

processing said symbolic names of the second code on the device,

transmitting processed instructions from the device to the web service, and

accepting a third code on the device over the network, where said third code is a device-independent third code including the output of the web component provided by the web service over the network and in response to the second code.

D. Asserted Grounds

The Petition contends that the challenged claims are unpatentable on the following grounds:

Claims	35 U.S.C. §	Reference(s)
23–25	103	Arner ¹
23–25	103	Huang ² and Shenfield ³

¹ U.S. Patent No. 8,341,595 (Ex. 1004).

² U.S. Patent Appl. Pub. No. 2007/0118844 (Ex. 1008).

³ U.S. Patent Appl. Pub. No. 2006/0200749 (Ex. 1012).

Petitioner relies on a Declaration of Kevin Jeffay, Ph.D., filed as Exhibit 1003. Patent Owner relies on a Declaration of Kevin C. Almeroth, Ph.D., filed as Exhibit 2001.

II. ANALYSIS

A. *Level of Skill in the Art*

The level of skill in the art is a factual determination that provides a primary guarantee of objectivity in an obviousness analysis. *See Al-Site Corp. v. VSI Int'l Inc.*, 174 F.3d 1308, 1323 (Fed. Cir. 1999) (citing *Graham v. John Deere Co.*, 383 U.S. 1, 17–18 (1966)). The level of skill in the art also informs the claim construction analysis. *See Teva Pharms. USA, Inc. v. Sandoz, Inc.*, 135 S. Ct. 831, 841 (2015) (explaining that claim construction seeks the meaning “a skilled artisan would ascribe” to the claim term “in the context of the specific patent claim”).

Petitioner asserts that a person of ordinary skill in the art “would have been a person with a Bachelor of Science in Computer Science and three years of experience as a web developer,” “would [have been] well versed in HTML, JavaScript, and CSS,” and “would [have been] familiar with server-side scripting and developing programs using higher level languages such as Java, C++, or Visual Basic.” Pet. 8 (citing Ex. 1003 ¶¶ 43–44). Patent Owner does not address this issue in the Preliminary Response.

As Patent Owner does not dispute Petitioner’s characterization of the level of skill in the art, and because we find it generally consistent with the disclosures of the ’755 patent and the prior art, we adopt it for purposes of this analysis.

B. Claim Construction

We construe claims using the same claim construction standard that would be used in a civil action under 35 U.S.C. § 282(b), as enunciated in *Phillips v. AWH Corp.*, 415 F.3d 1303 (Fed. Cir. 2005) (en banc), and subsequent cases. *See* 37 C.F.R. § 42.100(b).

Petitioner asserts that “[f]or purposes of this Petition,” it “applies the following constructions from parallel litigation relating to the ’755 patent”:

Claim Term	Court’s Construction(s)
“device-dependent code”	“code that is specific to the operating system, programming language, or platform of a device”
“device-independent code”	“code that is not specific to the operating system, programming language, or platform of a device”
“web component”	“software objects that provide functionalities of a web service”

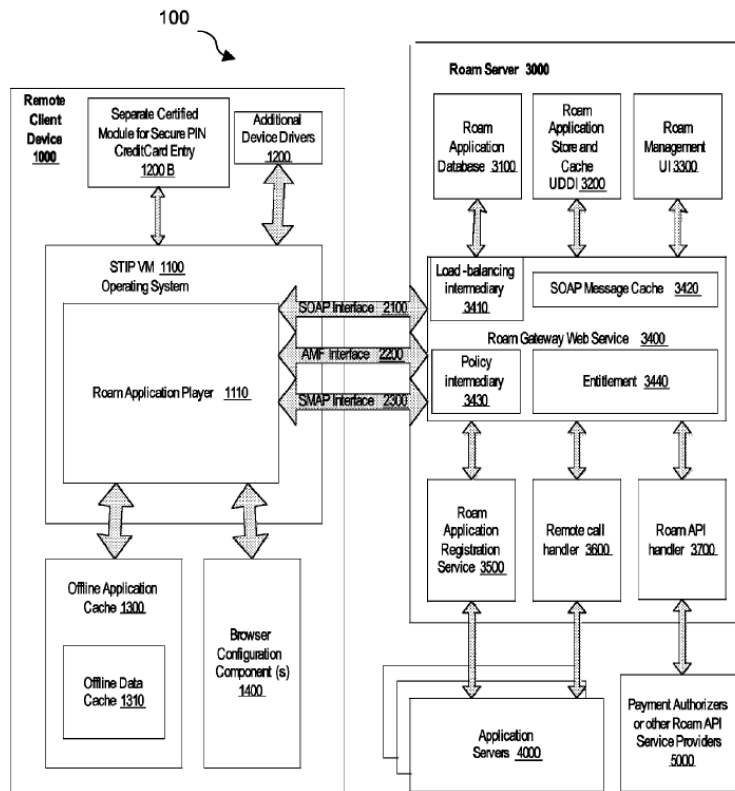
Pet. 9 (citing Ex. 1023, 10, 19–20). Petitioner also states that “one district court has construed ‘registry’ to mean ‘a database that is used for computing functionality,’” that Petitioner “disagrees with [that] construction,” but that “[r]egardless of which construction of ‘registry’ is applied, the Challenged Claims are obvious.” *Id.* at 9–10.

Patent Owner agrees with the construction listed above for “device-dependent code” and disagrees with the above constructions of “device-independent code” and “web component,” but asserts that the “ultimate construction” of these terms “does not impact this proceeding.” Prelim. Resp. 4–5.

We conclude that we need not engage in any formal claim construction to resolve the issues addressed in this Decision. *See Nidec Motor Corp. v. Zhongshan Broad Ocean Motor Co.*, 868 F.3d 1013, 1017 (Fed. Cir. 2017).

C. *Obviousness in View of Arner*

Arner is a U.S. Patent that concerns “Rich-Internet-Applications,” referred to as “RIAs,” that consist of XML code. *See* Ex. 1004, 5:14–16. It describes a “Roam Application Player” that interprets the XML and displays the RIAs. *See id.* at 9:63-64. The Player “requests functionality” for the RIA by communicating with a “server-side intermediary 3400 through XML messages embedded in Simple Object Access Protocol (SOAP) requests.” *Id.* at 7:10–12. This is illustrated in Figure 2:



“FIG. 2 is a detailed diagram of [Arner’s] system architecture.” Ex. 1004, 31–32.

The “intermediary server accesses . . . web services” and provides the requested functionality “via a single, secure and access-controlled network connection.” Ex. 1004, 4:19–23. It “receives the requested functionality/data” from the web service, and then provides a response to the Player on the device. *Id.* at Fig. 4. Arner explains that “[c]hanneling all communication with the network through a single, secure, certified server intermediary” provides critical “security,” “access control,” and “auditing” benefits. *Id.* at 5:4–10.

Arner also describes an integrated development environment (“IDE”) for creating and publishing the RIAs. See *id.* at 1:24–27. Arner describes how “[b]efore saving the application,” the IDE “converts the WSDL calls to GUIDs,” which are globally unique identifiers, and “replaces all web-service calls” in the RIA “with GUIDs.” *Id.* at 15:27–33. Then, when the RIA requires functionality from a web service, it sends the intermediary server—the “server web service”—a “SOAP request that contains a GUID” and the intermediary server “consults [a] database to determine what remote web service to call and sends a SOAP request” to the “remote web service.” *Id.* at 10:17–30. Additionally, “[f]or each supported target device and platform, the server runs an XSLT on the application to get another file . . . stripped down to only the relevant information for that device” and, when a device “request[s] a new application[,] . . . the server sends the device- and platform-appropriate version of the application to the device where it is then executed by the player.” *Id.* at 15:35–39, 44–48.

Petitioner contends that claims 23–25 of the ’755 patent are unpatentable under 35 U.S.C. § 103(a) as obvious in view of Arner alone.

See Pet. 10–38. Patent Owner argues that Arner fails to teach or suggest several limitations of claim 23. See Prelim. Resp. 12–24.

*“second code” that is “device-independent”
and “includes a plurality of symbolic names of inputs and outputs”*

Petitioner asserts that “Arner describes developing applications, i.e., RIAs, which correspond to the claimed second code” and that one of ordinary skill in the art “would understand that XML files are code and Arner itself expressly refers to ‘the application code.’” Pet. 19 (citing Ex. 1004, 1:25; 4:43–44; 12:36; Ex. 1003 ¶ 141). Petitioner argues that “Arner’s RIAs are device-independent code because they are XML files and XML is not specific to the operating system, programming language, or platform of a device,” and that Arner explains how “the use of an application player rather than attempting to run RIAs directly on the client device provides platform independence for the RIA.” Pet. 20 (citing Ex. 1003 ¶ 143 & quoting Ex. 1005, 9:27–28). Petitioner further argues that Arner’s RIAs include “a plurality of symbolic names of inputs and outputs associated with the web service” because it “discloses that web service functionality is integrated into an application during the software development process using a ‘WSDL Integrator’ tool, which is part of Arner’s [IDE].” Pet. 20 (citing Ex. 1003 ¶¶ 145–159; Ex. 1004, 4:6–10; Ex. 1005, 64).

Patent Owner responds that the RIAs are not “device-independent” and do not “include a plurality of symbolic names.” Prelim. Resp. 18. First, Patent Owner argues that Arner describes how the “development server” performs an “XSLT transformation of [the] application specific to the mobile device,” and that, as such, the RIA “is not device-independent” *Id.*

at 18–19 (citing Ex. 2001 ¶¶ 49–50; Ex. 1004, 15:35–40; 15:45–47, Fig. 8). Second, Patent Owner argues that Arner’s RIAs “do not include ‘symbolic names’ because “the IDE ‘replaces all web-service calls with GUIDs prior to publishing an application to the ‘deployment server.’” Pet. 19–20 (citing Ex. 2001 ¶ 50; Ex. 1004, 15:30–47, 15:33–38).

We agree with Patent Owner. Arner explains how the server “converts the WSDL calls to GUIDs and stores them in the database” and that “[b]efore saving the application to local data store, the server replaces all web-service calls with GUIDs.” Ex. 1004, 15:29–32. The RIAs thus do not include the WSDL symbolic names.

Claim 23 recites “[a] method of providing information to a device having a display from a web component of a web service to a device on a network.” It also recites that the “symbolic names” are those “required for evoking one or more web components each related to a set of inputs and outputs of a web service obtainable over a network.” This means that the “plurality of symbolic names of inputs and outputs associated with the web service” are those of the web service that is supplying information to the device, not some other identifier, such as the GUIDs.

Arner additionally explains that “[f]or each supported target device and platform, the server runs an XSLT on the application to get another file . . . stripped down to only the relevant information for that device” and that, when the devices request an application, “the server sends the device- and platform-appropriate version of the application to the device where it is then executed by the player.” *Id.* at 15:35–48. Arner thus describes how the RIAs are customized to particular devices, and not “device-independent.” Petitioner does not account for this disclosure. See Pet. 19–20. It may be

that use of Arner's player provides some degree of device or platform independence, but Arner describes how the RIAs include at least some customization for different platforms, meaning that they are not fully "device independent."

"processing said symbolic names . . . on the device" and "transmitting processed instructions from the device to the web service"

Petitioner argues that "Arner discloses that executing the first code (Arner's player) includes the step of processing said symbolic names of the second code (Arner's RIA) on the device." Pet. 32 (citing Ex. 1003 ¶¶ 192–194). According to Petitioner, one of skill in the art would have recognized that "execution of the RIA involves processing the WSDL-derived symbolic names in the application because Arner discloses that 'the player requests server-side functionality by sending the server web service a SOAP request that contains a GUID and relevant parameters (408)' to a server, which then 'determine[s] what remote web service to call and sends a SOAP request containing parameters to the specified remote web service or services.'" *Id.* at 33 (citing Ex. 1004, 10:19–26; Ex. 1005, 16:31–17:6; Ex. 1003 ¶ 193). Petitioner asserts that the artisan "would recognize that the appropriate parameters can be sent to the web service only if the player processes the symbolic names in the RIA." *Id.* (citing Ex. 1003 ¶ 194).

Patent Owner responds that "symbolic names" refers to the "plurality of symbolic names of inputs and outputs associated with the web service" and that "Arner fails to disclose or suggest that limitation" because "Arner's devices do not invoke web services and thus do not process the symbolic names associated with inputs and outputs of web services." Prelim. Resp. 12–13. Patent Owner argues that "[s]uch processing occurs on Arner's

intermediary server, not on the device.” *Id.* at 13. Patent Owner further argues that “devices do not transmit processed instructions—i.e., the instructions that invoke the web service—to the web service” because “[s]uch transmission is . . . performed by the intermediary server.” *Id.* (citing Ex. 2001 ¶¶ 52–58). Patent Owner asserts that, in Arner, “the intermediary server . . . incorporates the WSDL-derived code—which includes the parameter names Petitioner identifies as the symbolic names” and that, “[i]nsofar as Petitioner alleges that the symbolic names are incorporated into WSDL-generated code ‘for calling one or more web components,’ only the intermediary server performs that function, not the device.” *Id.* at 14 (citing Ex. 1004, 10:17–30; Ex. 2001 ¶¶ 52, 57). Patent Owner also cites Arner’s disclosure that “[f]or security reasons, the player always refers to web-service calls by their GUIDs and the server translates those GUIDs back into the actual calls at run-time.” *Id.* at 15 (citing Ex. 1004, 15:30–35). Patent Owner additionally argues that Petitioner fails to show that the “‘parameters’ contained in the application’s request to the intermediary server” are “the ‘WSDL-derived’ parameter names that it identifies as the symbolic names.” *Id.* at 16.

We agree with Patent Owner. As explained above, Arner’s IDE “converts the WSDL calls to GUIDs” and “replaces all web-service calls” in the RIA with GUIDs. Ex. 1004, 15:27–33. When the RIA runs, it sends the intermediary server a “SOAP request that contains a GUID” and the intermediary “server consults [a] database to determine what remote web service to call and sends a SOAP request” to the web service. *Id.* at 10:17–30. Arner therefore describes a system that does not use WSDL symbolic

names in the RIAs running on the player, and thus does not “process[] said symbolic names of the second code on the device,” as claimed.

Petitioner’s assertion that the SOAP request from the player includes “relevant parameters” (*see, e.g.*, Pet. 33) is insufficient to show processing of the symbolic names because Arner is clear that it is not using WSDL symbolic names in the RIAs. We see no basis to conclude that the “parameters” include symbolic names, and also see no reason why they would, because Arner is using the GUID to identify the web service.

We also agree with Patent Owner that Arner does not transmit “processed instructions from the device to the web service,” because it describes transmitting one instruction—the SOAP request containing the GUID—from the device to the intermediary, and a *different* instruction—the SOAP request to the web service—from the intermediary to the web service. *See* Ex. 1004, 10:17–26; *id.* at 15 (“For security reasons, the player always refers to web-service calls by their GUIDs and the server translates those GUIDs back into the actual calls at run-time.”). Petitioner does not account for the fact that the instructions are different, for example by seeking a claim construction of “instruction” that would cover both.

Petitioner also argues that “Patent Owner’s argument is inconsistent with the ’755 patent itself, which describes a web service architecture that uses an intermediary server, as in Arner.” Pet. 34. To support this argument, Petitioner compares Figure 2A of the ’755 patent with Figure 1 of Arner. *See id.* at 35. We agree that Figure 2A of the ’755 patent and Figure 1 of Arner have some similarities. However, we find this argument unpersuasive because Petitioner does not point us to any description in the ’755 patent in which the application is using an identifier other than a web

service symbolic name. The '755 patent describes how content server 140 “handle[s] all communication between device 130 and the web service 230,” but it does not describe converting the instructions, and Petitioner does not explain why that would be required. See Ex. 1001, 8:67–9:9 (explaining that the content server 230 simply “proxies the request [from device 130] to web service 230 and returns the requested information to device 130”).

Petitioner further argues that “[e]ven if the Challenged Claims were incorrectly interpreted to require ‘direct communication’ between the web service and the user’s device, such an architecture is obvious in view of Arner.” Pet. 35 (citing Ex. 1003 ¶¶ 205–206). Petitioner contends that “Arner describes the intermediary Roam Server itself as web service” and that “[i]f this server is viewed as the web service,” the artisan “would recognize that execution of the application by the player would involving processing and transmitting instructions from the second code to a web service server, without any intermediary server.” *Id.* (citing Ex. 1004, 10:7; Ex. 1003 ¶ 204; Ex. 1005, 16:23).

We do not agree that Arner’s description of server 3000 as a “web service” is a disclosure that the server 3000 is operating as a repository of the web services that may be called by the RIA. Arner distinguishes between the server 3000, which act as a “web service” or “server web service” to provide the user with RIAs, and “*remote web services*,” which are the source of the web services that would be called by the RIA. See Ex. 1004, 10:2–28.

Petitioner also contends that “Arner expressly discloses devices that directly connect to web service servers that are distinct from the ROAM Server.” Pet. at 36 (citing Ex. 1004, 1:32–57; Ex. 1005, 5:10–26).

Petitioner asserts that “while Arner prefers indirect communication for enhanced security, in some implementations this may not be a priority,” that one of skill in the art “may prefer speed instead, which would make direct communication with the web service more preferred,” and that the selection of the architecture “is simply a design choice,” and “both would be straightforward to . . . implement.” *Id.* (citing Ex. 1003 ¶¶ 205–206).

We find this cursory argument insufficient to establish a motivation to make what would amount to a wholesale modification of Arner. The cited portion of Arner is simply a paragraph of the Background section that generally describes how “smart client” applications “combine the best of the desktop applications and web-based applications.” Ex. 1004, 1:44–46. We fail to see how this suggests removing Arner’s server 3000, which is an important component of Arner’s system. *See id.* at 6:52–60. Moreover, Petitioner offers no evidence to establish that removal of Arner’s server 3000 would actually result in any noticeable increase in speed, much less an increase that might outweigh the loss of security that Arner’s server 3000 provides. In the absence of such evidence, we conclude that, on this record at least, the proposed modification is an impermissible hindsight reconstruction. *See KSR Int’l Co. v. Teleflex Inc.*, 550 U.S. 398, 421 (2007) (cautioning that “[a] factfinder should be aware . . . of the distortion caused by hindsight bias and must be cautious of arguments reliant upon *ex post* reasoning”).

* * *

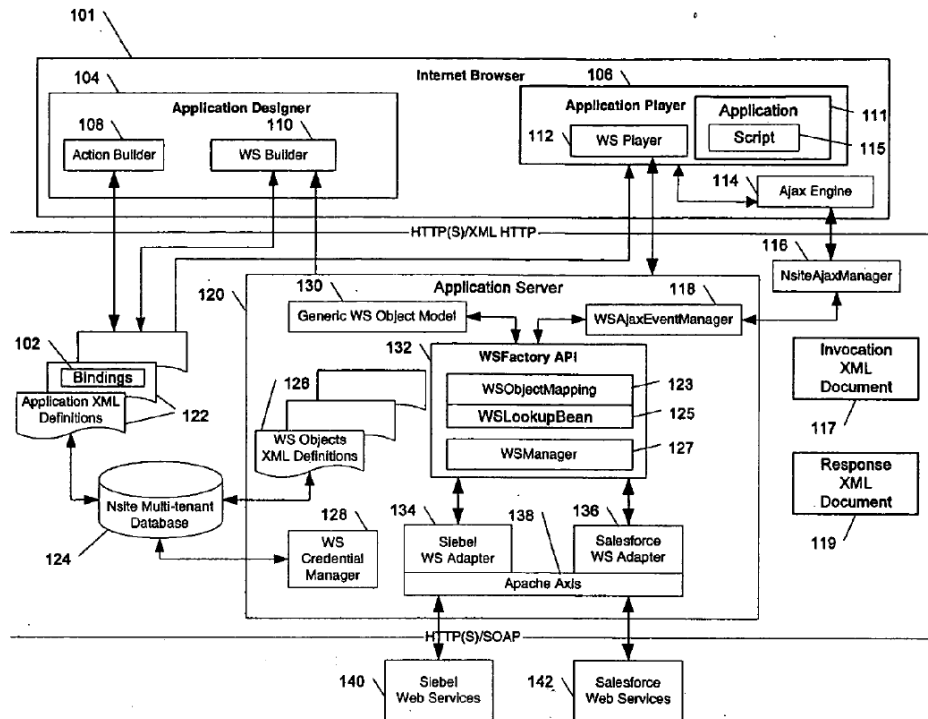
For these reasons, we agree with Patent Owner that Petitioner has not shown Arner to teach or suggest (a) “second code” that is “device-independent” and “includes a plurality of symbolic names of inputs and outputs” or (b) “processing said symbolic names . . . on the device” and “transmitting processed instructions from the device to the web service,” as recited in claim 23.

Petitioner thus has not established a reasonable likelihood of proving that claim 23 would have been obvious in view of Arner and, for the same reasons, Petitioner has not shown a reasonable likelihood of proving that dependent claims 24 or 25 would have been obvious in view of Arner.⁴ See *In re Fine*, 837 F.2d 1071, 1076 (Fed. Cir. 1988) (“Dependent claims are nonobvious under section 103 if the independent claims from which they depend are nonobvious.”).

⁴ Because these issues are dispositive of this ground, we do not reach Petitioner’s other arguments regarding the prior art.

D. Obviousness in View of Huang and Shenfield

Huang discloses a system for using web services as a data source for a software application. *See* Ex. 1008, Abstract. Figure 1A, reproduced below, shows an example of the system.



“FIG. 1A is an illustrative drawing of a system for developing and executing browser-based applications.” Ex. 1008 ¶ 9.

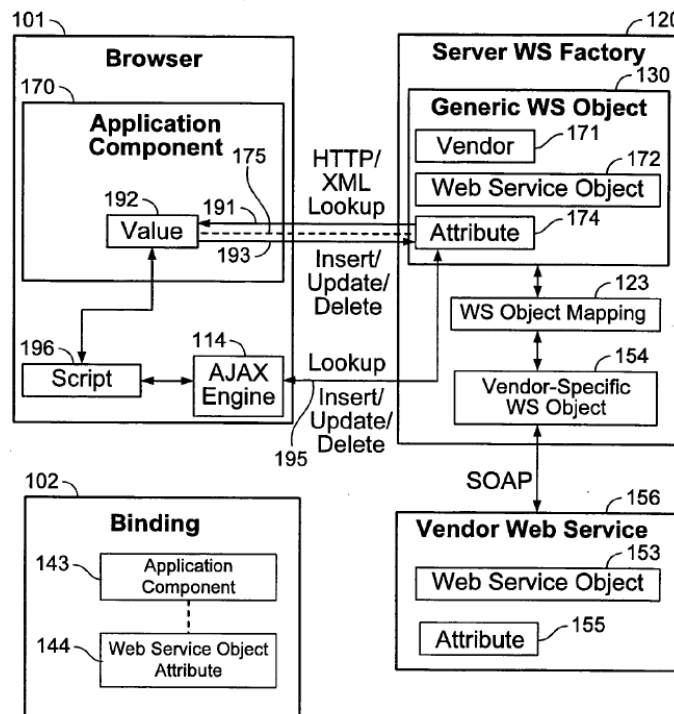
Huang’s system includes Application Designer 104 and Application Player 106. *See* Ex. 1008 ¶ 40. Application Designer 104 is a computer program with a user interface that allows a user to create application 111. *See id.* Application Player 106 is a computer program that executes application 111, receives values from web services, sends values to web services, and invokes other operations provided by web services. *See id.*

The components of server 120 include generic web service object model 130 (“WSObject”) and web service factory application programming

interface 132 (“WSFactory API”). Ex. 1001 ¶ 44. WSFactory API includes WSOBJECTMapping 123, which maps, or converts, the generic WSOBJECT to vendor-specific adapters, such as Siebel Web service adapter 134 for interacting with Siebel web service 140. *Id.* The mapping between generic WSOBJECT and the vendor-specific adapters is represented as WS Object XML definition 126, also referred to as an XML mapping file. *Id.*

Huang further describes how a user may “define bindings 102 between components of the application, e.g. buttons and tables, and web service attributes, e.g. values that can be stored in or retrieved from web services.” Ex. 1008 ¶ 55.

Figure 1B shows the various components of the system, including application component 170 running in a browser 101, the generic WSOBJECT 130, vendor web service 156, and a binding 193:



“FIG. 1B is an illustrative drawing of a binding between an application component and a web service.” Ex. 1008 ¶ 10.

Petitioner contends that claims 23–25 of the ’755 patent are unpatentable under 35 U.S.C. § 103(a) as obvious in view of Huang and Shenfield. *See* Pet. 39–72. Essentially, Petitioner relies on Huang, except that it cites Shenfield to support its argument that it would have obvious that the “player” be written in “device-dependent” code. *See* Pet. 44. Patent Owner argues that Huang fails to teach or suggest several limitations of claim 23. Prelim. Resp. 25–39.

“accepting, on the device, a second code” that “includes a plurality of symbolic names of inputs and outputs associated with the web service”; “symbolic names are provided from a registry of one or more web components related to inputs and outputs of a web service obtainable over a network”

Petitioner contends that “Huang describes an ‘application,’ which is a second code accepted on the device over the network,” and “discloses that its applications are XML-based code and are received by the internet browser on the user’s device ‘from a server.’” Pet. 51 (citing Ex. 1008 ¶ 42; Ex. 1010 ¶¶ 15, 18). Petitioner further argues that “Huang’s application (the second code) includes the claimed symbolic names as part of what Huang calls the ‘bindings’ included in the application code.” *Id.* (citing Ex. 1003 ¶¶ 258–277). Petitioner cites Huang’s explanation that the binding is “a two-way mapping which specifies that a particular web service attribute 155 can be set to the value 192 of the application component 170, and, conversely, that the value of the application component 170 can be set to the value of the attribute 155.” *Id.* at 51–52 (quoting Ex. 1008 ¶ 56). Petitioner further argues that “[b]ased on Huang’s discussion of WSDL and SOAP,” one of skill in the art “would further recognize that the bindings include symbolic names of inputs and outputs of the web service.” *Id.* at 52 (citing

Ex. 1003 ¶¶ 261–264). Petitioner argues that because Huang “discloses that SOAP messages are used to send and receive information from WSDL-based web services,” one “would recognize that such protocols necessitate the use of input and output symbolic names associated with the web service and defined in the associated WSDL.” *Id.* (citing Ex. 1003 ¶ 262).

Patent Owner argues that “Huang involves two different sets of symbolic names—those associated with a ‘generic web services object,’ in the intermediary server, and those associated with the ‘vendor web service.’” Prelim. Resp. 26 (citing Ex. 1008 ¶¶ 55, 74). Patent Owner asserts that “[t]he application . . . uses symbolic names for the generic web service object” and that “[t]hose symbolic names . . . are not symbolic names ‘associated with the web service’ because “[a]s Huang explains, if the application itself contained the symbolic names associated with the web service, that would ‘tie[] the application to a particular web services vendor.’” *Id.* at 27 (citing Ex. 2001 ¶ 64; Ex. 1008 ¶ 74).

We agree with Patent Owner. As described above, Huang is directed to a system that purposefully avoids using symbolic names of web services in the application in order to allow the application easy access to different web services, such as Siebel and Salesforce. The symbolic names associated with the generic object are then mapped, at the server level, to vendor specific objects. See Ex. 1008 ¶ 44 (“WSFactory API 132 . . . maps, i.e., converts, [WSObject] to vendor-specific [APIs], such as a Siebel Web Service Adapter 134 for interacting with the Siebel Web Service 140, and a Sales force Web Service Adapter 136 for interacting with the Salesforce Web Service 142.”); see *id.* ¶ 74 (“Applications interact with the generic web services object and need not contain hard-coded dependencies on

vendor specific objects.”). We thus conclude that Huang does not disclose “a second code” that “includes . . . symbolic names of inputs and outputs associated with the web service” that are “provided from a registry of one or more web components related to inputs and outputs of a web service.”

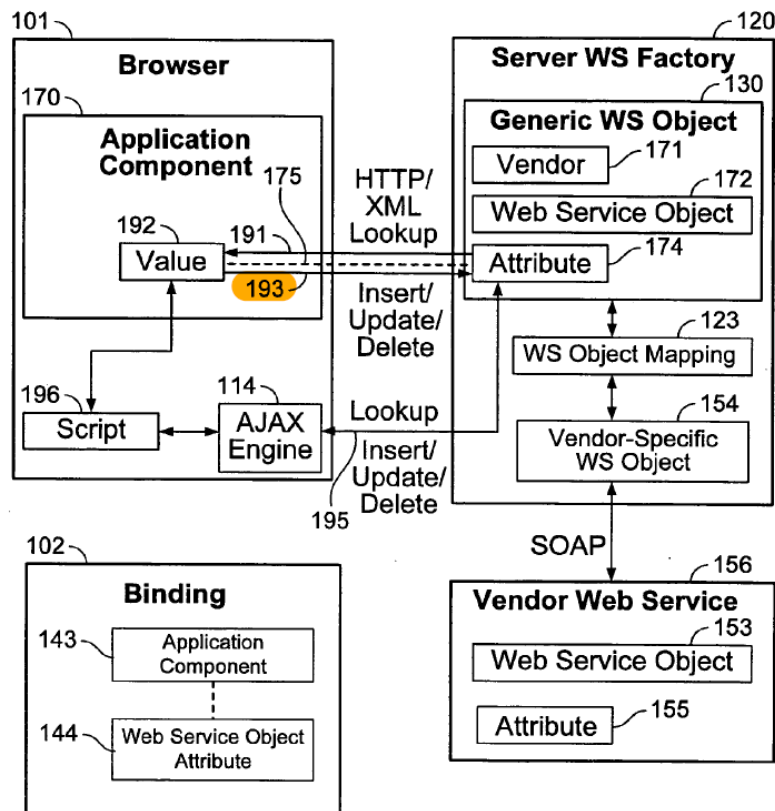
Petitioner alternatively argues that “Huang also discloses . . . that instead of using generic symbolic names in the application and mapping these generic names to vendor-specific web services, the device could call vendor-specific web service objects directly.” Pet. 54 (citing Ex. 1008 ¶ 74).

We do not agree with Petitioner’s reliance on Huang’s paragraph 74, which simply makes note of a problem with prior art systems that Huang sought to improve. We have previously found this passage to teach away from this modification:

In this paragraph, Huang is discussing the problem with calling a vendor-specific object directly from an application, and describing a solution of mapping each vendor-specific object to a generic web services object. Huang teaches that with this solution, applications need not contain hard-coded dependencies on vendor specific objects. Huang teaches that this solution allows the designer and the player to be vendor-independent and work with multiple web services without the need for vendor-specific code in the designer, player, or internet browser. A person of ordinary skill, upon reading Huang’s teaching that mapping each vendor-specific object to a generic web services object allows the designer and player to be vendor-independent, whereas calling vendor-specific objects without such mapping does not, would be led in a direction divergent from the path of “the device provides . . . [the] input symbolic name to the web service” that was taken by the applicant.

IPR2021-01144, Paper9, at 19–20 (PTAB January 18, 2022) (internal citations omitted; quoting *In re Fulton*, 391 F.3d 1195, 1201 (Fed. Cir. 2004)).

Petitioner also asserts that “Huang discloses that the symbolic names in the application will be generic (so the application can interact with web services from multiple vendors), but the bindings will include a direct mapping to the vendor-specific web service attribute.” Pet. 53 (citing Ex. 1008 ¶¶ 8, 60; Ex. 1003 ¶¶ 265–266). We do not agree. As shown in Figure 1B, reproduced below, highlighted binding 193 associates a value 192 of an application component 170 with an attribute 155 of a web service 156, but does so *through* Generic WS Object 130:



“FIG. 1B is an illustrative drawing of a binding between an application component and a web service.” Ex. 1008 ¶ 10.

Petitioner does not adequately explain why a system that binds application components to a generic object, as in Huang, would need to include the vendor-specific symbolic names in the applications. With respect to Huang's paragraph 60, Petitioner has not sufficiently shown that the "web service attributes" are symbolic names of web services, and not generic names that the Generic WS Object maps to different web services.

Petitioner's expert asserts that "[t]o the extent Patent Owner argues that the application contains a different symbolic name than the one specifically defined in the WSDL document describing the service (such as, for example, a generic symbolic name that maps to the web service attribute), that is a distinction without a difference." Ex. 1003 ¶ 265. We do not agree. The difference is that the use of the generic WSOBJ allows the application to access to multiple vendors, as described in Huang.

Petitioner concludes that "[m]ost importantly . . . nothing in the language of [claim 23] requires vendor-specific symbolic names to be included the claimed 'second code.'" *Id.* at 55. We again disagree. As noted above in connection with the discussion of Arner, claim 23 recites "[a] method of providing information to a device having a display from a web component of a web service to a device on a network" and that "the 'symbolic names' are those 'required for evoking one or more web components each related to a set of inputs and outputs of a web service obtainable over a network.'" The "plurality of symbolic names of inputs and outputs associated with the web service" are of the web service that is supplying information to the device.

“processing said symbolic names . . . on the device” and “transmitting processed instructions from the device to the web service”

With respect to processing the symbolic names on the device, Patent Owner argues that “[f]or example, Huang discloses that a ‘QUERY’ instruction may be sent from the device to the web service in response to a user action” and that one of skill in the art “would recognize that such instructions correspond to ‘processed instructions’ because they include data and values obtained by the Player from executing the application.” Pet. 66 (citing Ex. 1008 ¶ 69; Ex. 1003 ¶ 317; Ex. 1008 ¶¶ 40, 54). Patent Owner further argues that “[t]o the extent Patent Owner argues that Huang does not teach this claim limitation because Huang teaches an intermediary ‘Application Server’ located between the device and the web service,” the argument “should be rejected” because (1) the challenged claims “do not require a direct connection between the device and the web service,” (2) one of skill in the art “would recognize that the instructions transmitted from the user device are functionally the same as the instructions received by the web service,” and (3) one of skill in the art “would readily envision such an alternative design and its implementation would be well within [their] skill.” *Id.* at 67 (citing Ex. 1003 ¶¶ 319–326).

We find these three arguments unpersuasive. First, the fact that the claims do not recite a “direct” connection is immaterial because they *do* recite “transmitting the processed instructions from the device to the web service.” The issue is whether the “instructions” go “from the device to the web service.” Second, even if the instructions from the user device are “functionally the same” as the instructions received by the web service, they nevertheless are not the *same* “instructions,” and, as noted above, Petitioner

has not sought a construction of “instructions” that would encompass different instructions. Finally, Petitioner’s arguments that changing to a “direct connection” would be “readily envision[ed]” by one of skill in the art, “well within” the “skill” or “knowledge” of one of skill in the art, and “straightforward to implement” are, we conclude, insufficient to establish a motivation to modify Huang in the way Petitioner suggests. *See KSR*, 550 U.S. at 421 (explaining that “a patent composed of several elements is not proved obvious merely by demonstrating that each of its elements was, independently, known in the prior art”).

* * *

For these reasons, we agree with Patent Owner that Petitioner has not sufficiently shown Huang to teach or suggest “accepting, on the device, a second code” that “includes a plurality of symbolic names of inputs and outputs associated with the web service,” where the “symbolic names are provided from a registry of one or more web components related to inputs and outputs of a web service obtainable over a network,” as recited in claim 23. Petitioner also has not sufficiently shown Huang to teach or suggest “processing . . . symbolic names . . . on the device” and “transmitting processed instructions from the device to the web service.”

Petitioner thus has not established a reasonable likelihood of proving that claim 23 would have been obvious in view of Huang, and, for the same reasons, Petitioner has not shown a reasonable likelihood of proving that dependent claims 24 or 25 would have been obvious on view of Huang.⁵ *See In re Fine*, 837 F.2d at 1076.

⁵ Because these issues are dispositive of this ground, we do not reach Petitioner’s other arguments regarding the prior art.

E. Discretion

Because the above analysis is dispositive, we do not reach Patent Owner's arguments for discretionary denial under 35 U.S.C. §§ 314(a) & 325(d).

III. CONCLUSION

Petitioner has not shown a reasonable likelihood that it will prevail in showing that any of claims 23–25 of the '755 patent are unpatentable.

IV. ORDER

In consideration of the foregoing, it is hereby:

ORDERED that, pursuant to 35 U.S.C. § 314(a), an *inter partes* review is not instituted.

IPR2021-01471
Patent 9,063,755 B2

FOR PETITIONER:

Adam Brausa
DURIE TANGRI LLP
abrausa@durietangri.com

FOR PATENT OWNER:

Benoit Quarmby
Kenneth J. Weatherwax
Parham Hendifar
MOLOLAMKEN LLP
bquarmby@mololamken.com
weatherwax@lowensteinweatherwax.com
hendifar@lowensteinweatherwax.com