

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

ADOBE INC.,

Petitioner

v.

EXPRESS MOBILE, INC.,

Patent Owner

Case IPR2021-_____

U.S. Patent No. 9,063,755

Petition for *Inter Partes* Review of

U.S. Patent No. 9,063,755

TABLE OF CONTENTS

I.	INTRODUCTION	1
II.	GROUND FOR STANDING (37 C.F.R. § 42.104(A))	1
III.	PAYMENT OF FEES	1
IV.	STATEMENT OF PRECISE RELIEF REQUESTED FOR EACH CLAIM CHALLENGED	2
	A. Claims for Which Review is Requested.....	2
	B. Statutory Grounds of Challenge.....	2
V.	LEVEL OF ORDINARY SKILL IN THE ART.....	3
VI.	OVERVIEW OF THE '755 PATENT	3
	A. The '755 Patent	3
	B. Prosecution History of the '755 patent.....	5
VII.	OVERVIEW OF THE PRIOR ART	6
	A. Huang	6
	B. Shenfield.....	8
VIII.	CLAIM CONSTRUCTION	10
IX.	DETAILED EXPLANATION OF GROUNDS.....	13
	A. Ground 1: <i>Huang</i> in View of <i>Shenfield</i> Renders Obvious Claims 1, 3, 5-7, 11-12, 14, 16-18, and 22	13
	1. Claim 1	14
	2. Claim 3	67
	3. Claim 5	68
	4. Claim 6	69

5.	Claim 7	70
6.	Claim 11	70
7.	Claim 12	73
8.	Claim 14	75
9.	Claim 16	75
10.	Claim 17	76
11.	Claim 18	76
12.	Claim 22	76
X.	THE DISCRETIONARY FACTORS FAVOR INSTITUTING TRIAL	77
A.	This Petition Presents New, Non-Cumulative Prior Art That Has Not Previously Been Considered	77
B.	The <i>Fintiv</i> Factors Favor Institution	77
C.	The <i>General Plastic</i> Factors Favor Institution.....	81
XI.	MANDATORY NOTICES UNDER 37 C.F.R. § 42.8.....	86
A.	Real Parties-in-Interest	86
B.	Related Matters.....	86
C.	Lead and Backup Counsel.....	89
D.	Service Information	89

TABLE OF AUTHORITIES

	<u>Page(s)</u>
<u>Cases</u>	
<i>Apple, Inc. v. Fintiv, Inc.</i> , IPR2020-00019, Paper 11 (PTAB Mar. 20, 2020)	<i>passim</i>
<i>Apple, Inc. v. UUSI, LLC d/b/a Naraton</i> , IPR2019-00358, Paper 12 (PTAB Aug. 5, 2019)	83
<i>Brinkmann Corp. v. A&J Mfg., LLC</i> , IPR2015-00056, Paper 10 (PTAB Mar. 23, 2015)	1
<i>Callaway Golf Co. v. Acushnet Co.</i> , 576 F.3d 1331 (Fed. Cir. 2009)	8
<i>Cisco Sys., Inc. v. Ramot at Tel Aviv Univ. Ltd.</i> , IPR 2020-00122, Paper 15 (PTAB May 15, 2020)	80
<i>Core Wireless Licensing S.A.R.L. v. Apple Inc.</i> , No. 15-cv-05007-RS (N.D. Cal. June 15, 2016)	78
<i>DISH Network L.L.C. v. Broadband iTV, Inc.</i> , IPR2020-01359, Paper No. 15 (PTAB Feb. 12, 2021)	80
<i>Ericsson Inc. v. Intellectual Ventures I LLC</i> , IPR2018-01318, Paper 6 (PTAB Jan. 15, 2019)	85
<i>Fitbit, Inc. v. Valencell, Inc.</i> , 964 F.3d 1112 (Fed. Cir. 2020)	11
<i>General Plastic Industrial Co., Ltd. v. Canon Kabushiki Kaisha</i> , IPR2016-01357, Paper 19 (PTAB Sept. 6, 2017)	81, 82, 83, 84
<i>Harari v. Lee</i> , 656 F.3d 1331 (Fed. Cir. 2011)	8
<i>Karl Storz Endoscopy-Am., Inc. v. Stryker Corp.</i> , No. 14-cv-00876-RS, 2015 WL 13727876 (N.D. Cal. Mar. 30, 2015)	78

<i>KSR Int’l Co. v. Teleflex Inc.</i> , 550 U.S. 398 (2007).....	37
<i>Netflix, Inc. v. Realtime Adaptive Streaming LLC</i> , IPR2018-01630 Paper 13 (PTAB Apr. 19, 2019)	84, 85
<i>Nevro Corp. v. Boston Sci. Neuromodulation Corp.</i> , IPR2017-01920, Paper 12 (PTAB Feb. 5, 2018).....	83
<i>Novo Indus., L.P. v. Micro Molds Corp.</i> , 350 F.3d 1348 (Fed. Cir. 2003)	11
<i>OrthoAccel Techs., Inc. v. Propel Orthodontics, LLC</i> , No. 17-cv-03801-RS (N.D. Cal. Aug. 10, 2018).....	78
<i>Phillips v. AWH Corp.</i> , 415 F.3d 1303 (Fed. Cir. 2005) (<i>en banc</i>)	10
<i>Samsung Elecs. Am., Inc. v. Uniloc Luxembourg S.A.</i> , IPR2017-01797, Paper 8 (PTAB Feb. 6, 2018).....	82
<i>Samsung Elecs. Co., Ltd. v. Iron Oak Techs., LLC</i> , IPR2018-01554, Paper 9 (PTAB Feb. 13, 2019).....	82
<i>Signify Holding B.V. v. Lighting Sci. Grp. Corp.</i> , IPR2020-00753, Paper 16 (PTAB Sept. 16, 2020).....	83
<i>Taiwan Semiconductor Mfg. Co., Ltd. v. Fraunhofer Gesellschaft Zur Förderung Der Angewandten Forschung EV</i> , IPR2020-01669, Paper No. 13 (PTAB Apr. 15, 2021)	79
<i>Toyota Motor Corp. v. Cellport Sys., Inc.</i> , IPR2015-00633, Paper No. 11 (Aug. 14, 2015)	10
<i>Traverse LLC v. Amprius</i> , No. 20-cv-00484-RS, 2020 WL 6820942 (N.D. Cal. Nov. 6, 2020)	78
<i>United Fire Protection Corp. v. Eng’red Corrosion Solutions, LLC</i> , IPR2018-00991, Paper 10 (PTAB Nov. 15, 2018).....	84
<i>Vivid Techs., Inc. v. Am. Sci. & Eng’g, Inc.</i> , 200 F.3d 795 (Fed. Cir. 1999)	10

<i>Volkswagen Grp. of Am., Inc. v. Carucel Investments, L.P.</i> , IPR2019-01103, Paper 7 (PTAB Nov. 12, 2019).....	82, 83, 84, 85
<i>Volkswagen Grp. of Am., Inc. v. Carucel Investments, L.P.</i> , IPR2019-01573, Paper 7 (PTAB Jan. 22, 2020)	85
<i>X Commerce, Inc. v. Express Mobile, Inc.</i> , No. 17-cv-02605-RS (N.D. Cal. Mar. 5, 2021)	78

Statutes

37 C.F.R. § 42.8	86
37 C.F.R. § 42.24(a).....	91
35 U.S.C. §§ 102(a)	3
35 U.S.C. § 102(b)	3
35 U.S.C. § 103	2
35 U.S.C. § 311	2
35 U.S.C. §314(a)	77, 81
35 U.S.C. § 315(b)	1
37 C.F.R. § 42.104(A)	1
America Invents Act	11
Fed. R. Civ. P. 4(d)(4).....	1

Other Authorities

83 Fed. Reg. 51,340-51,359 (Oct. 11, 2018)	10
--	----

LIST OF EXHIBITS

Ex-1001	U.S. Patent No. 9,063,755 to Rempell (“the ’755 patent”)
Ex-1002	Declaration of Ian Cullimore, Ph.D.
Ex-1003	Curriculum Vitae of Ian Cullimore, Ph.D.
Ex-1004	Prosecution History of U.S. Patent No. 9,063,755
Ex-1005	U.S. Patent Application Publication No. 2007/0118844 (“ <i>Huang</i> ”)
Ex-1006	U.S. Patent Application Publication No. 2007/0079282 (“ <i>Nachnani</i> ”)
Ex-1007	U.S. Patent Application Publication No. 2006/0200749 (“ <i>Shenfield</i> ”)
Ex-1008	RESERVED
Ex-1009	RESERVED
Ex-1010	RESERVED
Ex-1011	Memorandum Opinion (Dkt. 121), <i>Express Mobile, Inc. v. GoDaddy.com, LLC</i> , C.A. No. 19-1937-RGA (D. Del. June 1, 2021)
Ex-1012	Memorandum Order (Dkt. 137), <i>Shopify Inc. et al. v. Express Mobile, Inc.</i> , C.A. No. 19-439-RGA (D. Del. June 23, 2020)
Ex-1013	RESERVED
Ex-1014	RESERVED
Ex-1015	Joint Claim Construction Brief (Dkt. 117), <i>Shopify Inc. et al. v. Express Mobile, Inc.</i> , C.A. No. 19-439-RGA (lead case) (D. Del. May 8, 2020)

Ex-1016	Alan Grosskurth and Michael W. Godfrey, A Reference Architecture for Web Browsers, Journal of Software Maintenance & Evolution: Research & Practices 2006; 00:1-7 (2006)
Ex-1017	Hypertext Markup Language - 2.0, IETF (Nov. 1997), <i>available at</i> https://datatracker.ietf.org/doc/html/rfc1866
Ex-1018	Francisco Curbera et al., Unraveling the Web Services Web, IEEE Internet Computing Vol. 7, Issue 2 86-93 (Apr. 2002)
Ex-1019	U.S. Patent Application Publication No. 2007/0067421
Ex-1020	Google Search WSDL Document, <i>available at</i> https://cs.au.dk/~amoeller/WWW/webservices/GoogleSearch.wsdl (Rev. Aug. 16, 2002)
Ex-1021	U.S. Patent Application Publication No. 2003/0009523
Ex-1022	U.S. Patent Application Publication No. 2006/0034202
Ex-1023	U.S. Patent Application Publication No. 2004/0186888
Ex-1024	U.S. Patent Application Publication No. 2006/0221190
Ex-1025	U.S. Patent Application Publication No. 2007/0100836
Ex-1026	U.S. Patent No. 7,191,160
Ex-1027	Steven J. Vaughan-Nichols, Web Services: Beyond the Hype, Computer 35(2):18-21 (Feb. 2002)
Ex-1028	U.S. Patent No. 7,873,625
Ex-1029	U.S. Patent Application Publication No. 2006/0230462
Ex-1030	RESERVED

Ex-1031	Order Granting Stay (Dkt. 105), <i>OrthoAccel Techs., Inc. v. Propel Orthodontics, LLC</i> , No. 17-cv-03801-RS (N.D. Cal. Aug. 10, 2018)
Ex-1032	Order Staying Action (Dkt. 205), <i>Core Wireless Licensing S.A.R.L. v. Apple Inc.</i> , No. 15-cv-05007-RS (N.D. Cal. June 15, 2016)
Ex-1033	Order Granting Stay and Denying Motion to Consolidate (Dkt. 254), <i>X Commerce, Inc. v. Express Mobile, Inc.</i> , No. 17-cv-02605-RS (lead case) (N.D. Cal. Mar. 5, 2021)
Ex-1034	RESERVED
Ex-1035	RESERVED
Ex-1036	Proposed Schedules (Dkt. 50-1), <i>Express Mobile, Inc. v. Adobe Inc. et al.</i> , No. 20-cv-08297 (N.D. Cal. Apr. 23, 2021)
Ex-1037	Order re Claim Construction Schedule (Dkt. 51), <i>Express Mobile, Inc. v. Adobe Inc. et al.</i> , No. 20-cv-08297 (N.D. Cal. Apr. 26, 2021)

I. INTRODUCTION

Petitioner Adobe Inc. (“Adobe”) requests *inter partes* review (“IPR”) of claims 1, 3, 5-7, 11-12, 14, 16-18, and 22 of U.S. Patent No. 9,063,755 (“the ’755 patent”).

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

Adobe certifies that the ’755 patent is available for IPR and that Adobe is not barred or estopped from requesting IPR to challenge the claims on the grounds herein. Adobe filed this Petition within one year of the filing of a waiver of service of Patent Owner’s (“PO”) first complaint against Adobe in district court. *See* Dkt. 27, *Express Mobile, Inc. v. Adobe Inc. et al.*, Case No. 20-cv-08297-RS (N.D. Cal.) (“ExMo-Adobe Litigation”); 35 U.S.C. § 315(b); Fed. R. Civ. P. 4(d)(4); *Brinkmann Corp. v. A&J Mfg., LLC*, IPR2015-00056, Paper 10 at 6-7 (PTAB Mar. 23, 2015) (filing of waiver of service represents date of service).

III. PAYMENT OF FEES

The U.S. Patent and Trademark Office (“USPTO”) is authorized to charge any fees due during this proceeding to Deposit Account No. 502387.

IV. STATEMENT OF PRECISE RELIEF REQUESTED FOR EACH CLAIM CHALLENGED

A. Claims for Which Review is Requested

Adobe respectfully requests review of claims 1, 3, 5-7, 11-12, 14, 16-18, and 22 (“challenged claims”) of the ’755 patent and cancellation of these claims under 35 U.S.C. § 311 as unpatentable.

B. Statutory Grounds of Challenge

The challenged claims should be cancelled as unpatentable on the following ground:

Ground 1: Claims 1, 3, 5-7, 11-12, 14, 16-18, and 22 are unpatentable under pre-AIA 35 U.S.C. § 103 as obvious over U.S. Patent Application Publication No. 2007/0118844 (“*Huang*”), titled “Designer and Player for Web Services Applications” (Ex-1005, Cover), in view of U.S. Patent Application Publication No. 2006/0200749 (“*Shenfield*”), titled “System and Method for Conversion of Generic Services’ Applications into Component Based Applications for Devices” (Ex-1007, Cover).

The ’755 patent issued from U.S. Patent Application No. 12/936,395, which is the national stage of PCT/US2009/039695, filed on April 6, 2009, and claims priority to, among others, U.S. Provisional Patent Application No. 61/123,438, filed on April 7, 2008. (Ex-1001, Cover.) For this proceeding only, Petitioner assumes the earliest effective filing date of the ’755 patent is April 7, 2008.

Huang published on May 24, 2007, from U.S. Patent Application No. 11/286,267, filed on November 23, 2005. Accordingly, *Huang* qualifies as prior art to the '755 patent under pre-AIA 35 U.S.C. §§ 102(a) and/or (e).

Shenfield published on September 7, 2006, from U.S. Patent Application No. 11/069,971, filed on March 3, 2005. Accordingly, *Shenfield* qualifies as prior art to the '755 patent under pre-AIA 35 U.S.C. § 102(b).

The USPTO did not consider *Huang* and *Shenfield* during prosecution of the '755 patent. (Ex-1001, Cover (“References Cited”).)

V. LEVEL OF ORDINARY SKILL IN THE ART

A person of ordinary skill in the art at the time of the alleged invention of the '755 patent (“POSITA”), which for purposes of this proceeding is April 7, 2008, would have had a B.S. in Computer Science, Computer Engineering, or equivalent, plus two years working experience. (Ex-1002, ¶15.) More education can supplement practical experience and vice-versa. (*Id.*)

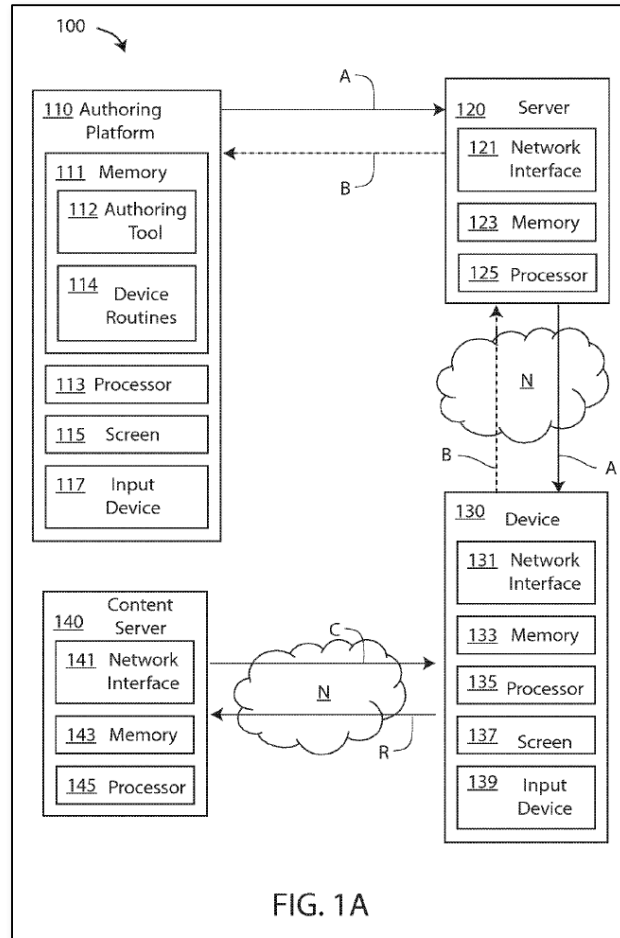
VI. OVERVIEW OF THE '755 PATENT

A. The '755 Patent

The '755 patent “generally relates to providing software for mobile devices, and more particularly to a method and system for authoring Applications for devices.” (Ex-1001, 1:8-10; *see also* Ex-1004, 1209-16 (Appeal Brief summarizing '755 patent).) The '755 patent describes a “system” with “a database of web services

obtainable over a network and an authoring tool” that “generate[s] code to provide content on a display of a platform.” (Ex-1001, 1:34-37.) “The authoring tool is configured to define an object for presentation on the display, select a component of a web service included in said database, associate said object with said selected component, and produce code that, when executed on the platform, provides said selected component on the display of the platform.” (*Id.*, 1:37-42; *see also id.*, 4:13-16.) The ’755 patent states that, “[i]n one embodiment, authoring tool 112 is a graphical system for designing the layout of features as a display that is to appear on screen 137.” (*Id.*, 3:47-49.) One “mechanism for binding the outputs of the web service to the UI components is through symbolic references that matches each output to the symbolic name of the UI component.” (*Id.*, 9:11-14.) The generated code includes a device-independent “Application” and a device-dependent “Player.” (*See id.*, Abstract, 1:59-2:3.)

Figure 1A of the ’755 patent illustrates “a system including an authoring platform and a server for providing programming instructions to a device over a network.” (Ex-1001, 2:14-17.)



(Ex-1001, Fig. 1A; *see also id.*, 2:65-4:41.)

B. Prosecution History of the '755 patent

During prosecution, the applicant admitted, among other things, that the prior art disclosed registries (“Benedetti teaches a registry of web services”), device-dependent players (“since the code is generated to run on a specific device, such as server 34, McCain is teaching the generation of a Player”), device-independent applications (identifying “device-independent rule set 150” as an “application”), accessing web services, and symbolic names with associated data-format types (the

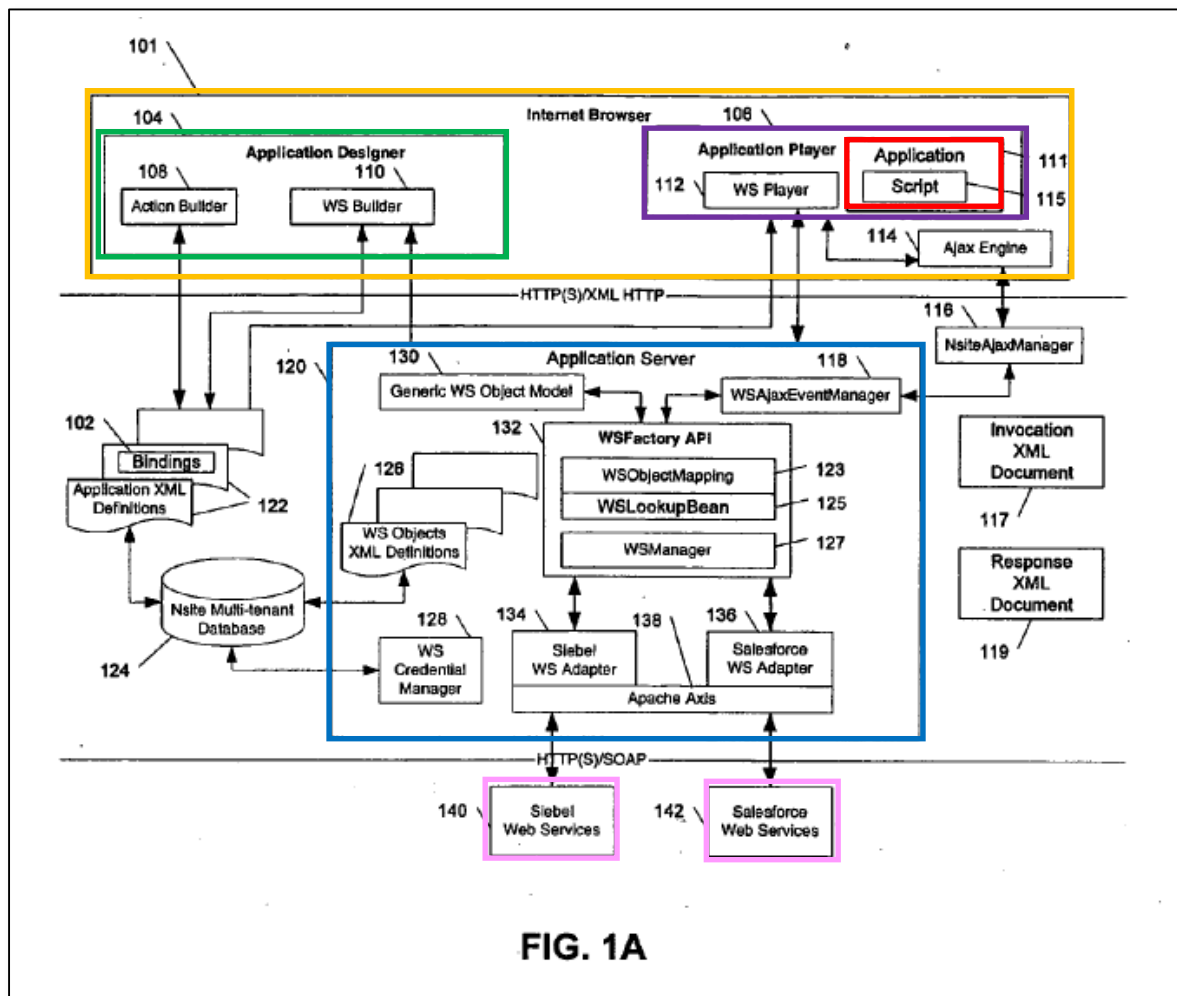
names and data structures in a WSDL document). (*See, e.g.*, Ex-1004, 1068-69, 1216-22.) In the applicant’s Appeal Brief challenging a final Office Action rejecting all pending claims over prior art, the applicant argued, that “the combination of references, taken individually or in combination, does not teach or suggest providing both device-independent code and device-dependent code to a device that, when executed on the device, exchanges web service input and output names and values, and displays output on such device.” (Ex-1004, 1223.) The examiner ultimately found that the prior art did not render the pending claims obvious (*id.*, 1255-59), but the examiner did not consider references that disclosed providing both device-independent and device-dependent code to a device to output a display, such as *Huang* or *Shenfield* discussed herein. (*See generally*, Ex-1004.)

VII. OVERVIEW OF THE PRIOR ART

A. Huang

Huang seeks to promote interoperability of computer functionality. *Huang* identifies the problem of integration of systems that utilize different operating systems. (Ex-1005, ¶[0003].) To that end, *Huang* discloses the use of software applications that execute in web browsers to allow for “access to web services supported by each of a number of different organizations each having a different type of computer system and/or database.” (*Id.*, ¶[0007].)

Huang's system creates "composite software applications which draw objects from several different organizations." (Ex-1005, ¶[0008].) *Huang*'s system includes a **browser 101** running a **designer 104**, which generates an **application 111** that will be executed on **player 106**, and communicates with **web services 140/142** via **server 120**:¹



(Ex-1005, Fig. 1A; *see also id.*, ¶¶[0040]-[0043].)

¹ All emphases and color annotations to figures added unless otherwise noted.

Huang's "designer module ... is the builder of the application" and its "player module ... run[s] the application." (Ex-1005, ¶[0008].) The designer module "creates the binding or mapping of meta-information" such that the objects in an "application" can access web services from other organizations. (*See id.* at ¶¶[0007]-[0008].) *Huang* incorporates by reference *Nachnani* (Ex-1006) in its entirety, which provides additional details regarding *Huang*'s system, including *Huang*'s designer, players, applications, and how applications are built. (Ex-1005, ¶¶[0006], [0040], [0042], [0133]; Ex-1002, ¶78.) *Harari v. Lee*, 656 F.3d 1331, 1335 (Fed. Cir. 2011) (finding statement "[t]he disclosure so the two applications are hereby incorporate[d] by reference" was "unequivocal" and incorporated entirely); *Callaway Golf Co. v. Acushnet Co.*, 576 F.3d 1331, 1346-47 (Fed. Cir. 2009) (holding phrase "[r]eference is made to [Molitor '637] which describes a number of foamable compositions of a character which may be employed for one or both layers ... for the golf ball of this invention" incorporated "the entire list of foamable compounds ... disclosed by Molitor '637").

B. Shenfield

Shenfield also addresses interoperability, focusing on the "need for application programs ... that can be run on client devices having a wide variety of runtime environments." (Ex-1007, ¶[0005].) To that end, *Shenfield* discloses a system that generates and provides over a network "component applications" that

include both “platform-neutral” and “platform-specific” components. (*Id.*, ¶¶[0029], [0044]-[0045], [0057].) Specifically, *Shenfield* contemplates “platform-neutral data 400 [and] message 404” components that “define data entities,” including those that are “dynamically generated according to Web Service(s) 106” and “define the format of messages ... to communicate with external systems such as the web service 106.” (*Id.*, ¶¶[0041]-[0042], [0044].) These “platform-neutral” components run with “platform-specific presentation component 402 descriptors [that define the appearance and behavior of the component applications 105 as it is displayed by the user interface 202] for various pre-defined client runtimes” and similar workflow components that are “a series of instructions in [a] selected programming/scripting language” and “define differing work flows based upon different supported capabilities or feature of particular devices.” (*Id.*, ¶¶[0043]-[0045].)

Like *Huang*, *Shenfield*’s component applications may utilize web services. (Ex-1007, ¶[0024].) *Shenfield* explains that web services have interfaces described in “Web Services Description Language (WSDL)” that can be obtained from “a Universal Discovery Description and Integration (UDDI) services registry.” (Ex-1007, ¶[0025].)

VIII. CLAIM CONSTRUCTION

For IPR proceedings, the Board construes claim terms only when necessary to resolve the underlying controversy. *Toyota Motor Corp. v. Cellport Sys., Inc.*, IPR2015-00633, Paper No. 11 at 16 (Aug. 14, 2015) (citing *Vivid Techs., Inc. v. Am. Sci. & Eng’g, Inc.*, 200 F.3d 795, 803 (Fed. Cir. 1999)). When construing claims, the Board applies the standard in *Phillips v. AWH Corp.*, 415 F.3d 1303, 1312-16 (Fed. Cir. 2005) (*en banc*): claim terms are typically given their ordinary and customary meanings, as understood by a POSITA at the time of the invention, after consider the claim language, the specification, and the prosecution history. *See* 83 Fed. Reg. 51,340-51,359 (Oct. 11, 2018).

Express Mobile has stipulated to the following constructions, which district courts have adopted:

Term	Stipulated Construction
“where said Application is a device-dependent code”	“where said Application is a device-independent code” (Ex-1011, 10.) ²

² The correction to this obvious error is not subject to reasonable debate (Ex-1015, 2 (Express Mobile agreeing to construction); Ex-1011, 10 (same); Ex-1001,

Term	Stipulated Construction
“authoring tool” / “authoring tool configured to”	“a system, with a graphical interface, for generating code to display content on a device screen” (Ex-1011, 10.)
“device-dependent code”	“code that is specific to the operating system, programming language, or platform of a device” (Ex- 1011, 10.)
“device-independent code”	“code that is not specific to the operating system, programming language, or platform of a device” (Ex- 1011, 10.)
“for evoking one or more web components”	“for calling up one or more web components” (Ex-1011, 10.)

Abstract, 2:1-2, 5:13-15, 5:57-59, 6:4-5, 6:13-14, 13:49 (specification describing “Application” as “device-independent”)) and the prosecution history does not suggest a different interpretation (Ex-1002, ¶44). *Novo Indus., L.P. v. Micro Molds Corp.*, 350 F.3d 1348, 1354 (Fed. Cir. 2003)); *see also Fitbit, Inc. v. Valencell, Inc.*, 964 F.3d 1112, 1119-20 (Fed. Cir. 2020) (“there is foundation for such authority [of the Board to correct errors] in the America Invents Act”).

District courts have construed the following terms in the challenged claims:

Term	Courts' Construction	Exemplary Intrinsic Evidence
“registry”	“a database that is used for computing functionality” (Ex-1011, 17-19.)	Ex-1001, 1:34-42, 1:51-58, 7:1-3, 7:63-66, 8:19-22, 22:26-29; Ex-1004, 1069.
“web component”	“software objects that provide functionalities of a web service” (Ex-1011, 19-20.)	Ex-1001, 8:18-22, 8:40-42, Fig. 3F.
“web service”	“a software system that supports interaction between devices over a network” (Ex-1011, 20-22.)	Ex-1001, 8:18-19; Ex-1004, 1068.
“Application” / “application”	“device-independent code containing instructions for a device and which is separate from the Player” Ex-1011, 23-25.)	Ex-1001, 1:59-2:3, 5:8-15; Ex-1004, 1214, 1216.

Term	Courts' Construction	Exemplary Intrinsic Evidence
"Player" / "player"	"device-specific code which contains instructions for a device and which is separate from the Application" (Ex-1012, 9-14)	Ex-1001, 1:59-2:3, 5:8-15; Ex-1004, 1214, 1216.

In this Petition, Adobe applies the district court construction of the terms in the table above. The district courts' constructions are stipulated or consistent with the intrinsic record. Even if the Board were to apply the plain meanings instead of the district courts' constructions, the result would not change as the plain meaning would be consistent with or subsume the scope of the construed claims above. (Ex-1002, ¶46.)

IX. DETAILED EXPLANATION OF GROUNDS

A. Ground 1: *Huang* in View of *Shenfield* Renders Obvious Claims 1, 3, 5-7, 11-12, 14, 16-18, and 22

As set forth below, a POSITA would have found the challenged claims obvious over *Huang* in view of *Shenfield*.

1. Claim 1

- a. [1.pre] “A system for generating code to provide content on a display of a device, said system comprising:”

Huang discloses element 1.pre. *Huang* discloses a “computer enabled *system* for *developing* and executing *browser-based software applications*.” (Ex-1005, ¶[0040].) *Huang* discloses a system that includes **designer 104** that *generates code*, such as **application 111**, which *provides content on a display of a device* when executed in **player 106**, with both **designer 104** and **player 106** executing within one or more **web browser[s] 101**:

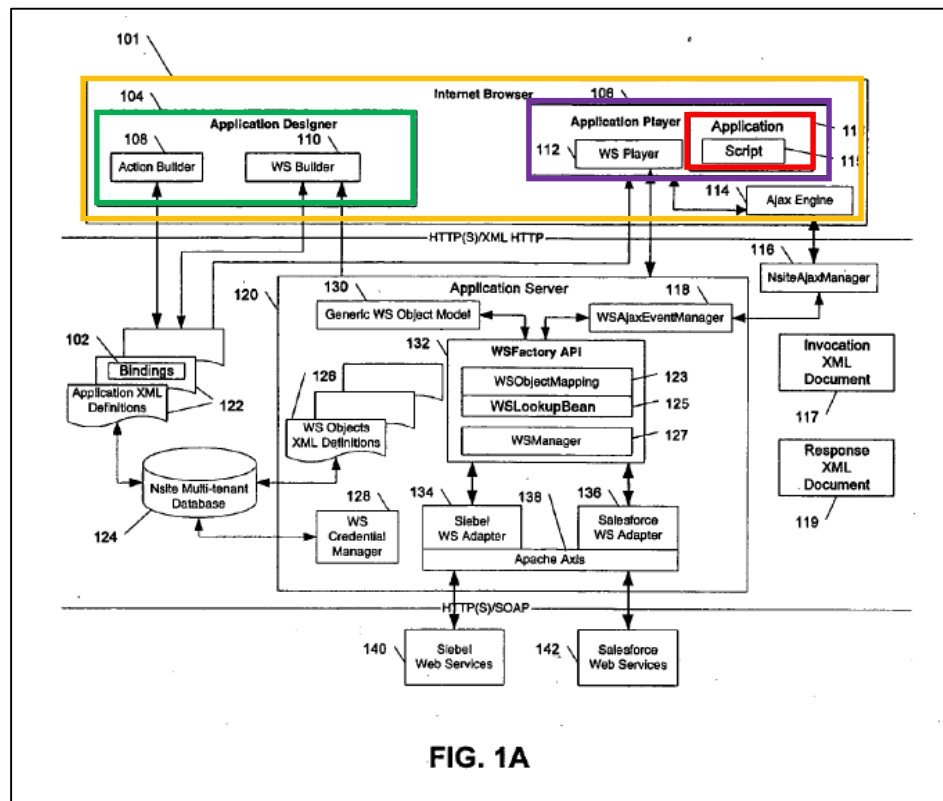


FIG. 1A

(Ex-1005, Fig. 1A; see also *id.*, ¶¶[0040]-[0043].)

Huang's system generates code to provide content on a display of a device.

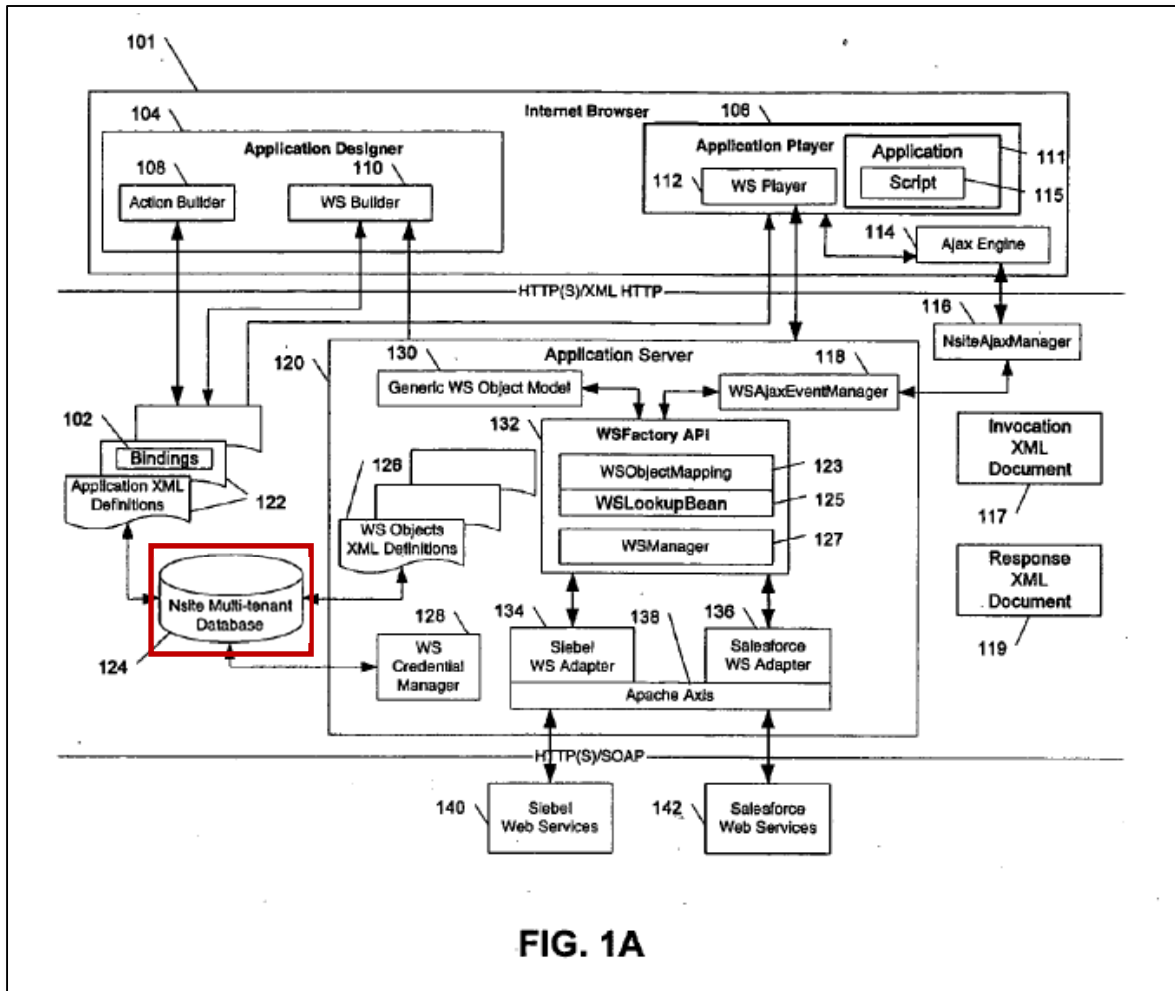
(Ex-1002, ¶¶92-99.) *Huang's* designer 104 *generates* applications (Ex-1005, ¶[0008]), such as application 111, which a POSITA would have understood is code because it is a “software computer program” (*id.*, ¶[0040]) and is “stored in Extensible Markup Language (XML) format as XML definitions 122” (*id.*, [0042]). (Ex-1002, ¶¶94-95.) A POSITA would have further understood that *Huang's* applications *display content* through player 106 running in web browser 101 (Ex-1005, ¶[0042] & Fig. 1A) because “Player 106 ... provides a user interface which executes a browser based application 111” (*id.*, ¶[0040]). (Ex-1002, ¶97.) A POSITA would have understood that browsers visually display HTML pages and would have understood that the “user interface ... [of] a browser based application 111” would likewise be displayed. (*Id.*, ¶98.) *Nachnani*, which *Huang* incorporates by reference (Ex-1005, ¶¶[0006], [0040]), confirms that the “Player UI” operates on a “Client Computer” (e.g., *device*) and “executes an application 135” to “present[] the application 135 to a user 130 via the web browser,” including “user interface portions” and “data associated with the application.” (Ex-1006, ¶¶[0040], [0097] & Fig. 8; Ex-1002, ¶99.)

- b. [1.a] “computer memory storing a registry of: 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;”

Huang in view of *Shenfield* renders this limitation obvious under the district court’s construction of the terms “registry” (“a database that is used for computing functionality”), “web component[]” (“software objects that provide functionalities of a web service”), “web service” (“a software system that supports interaction between devices over a network”), and “for evoking one or more web components” (“for calling up one or more web components”). (*Supra* Section VIII.)

- (i) “computer memory storing a registry of”

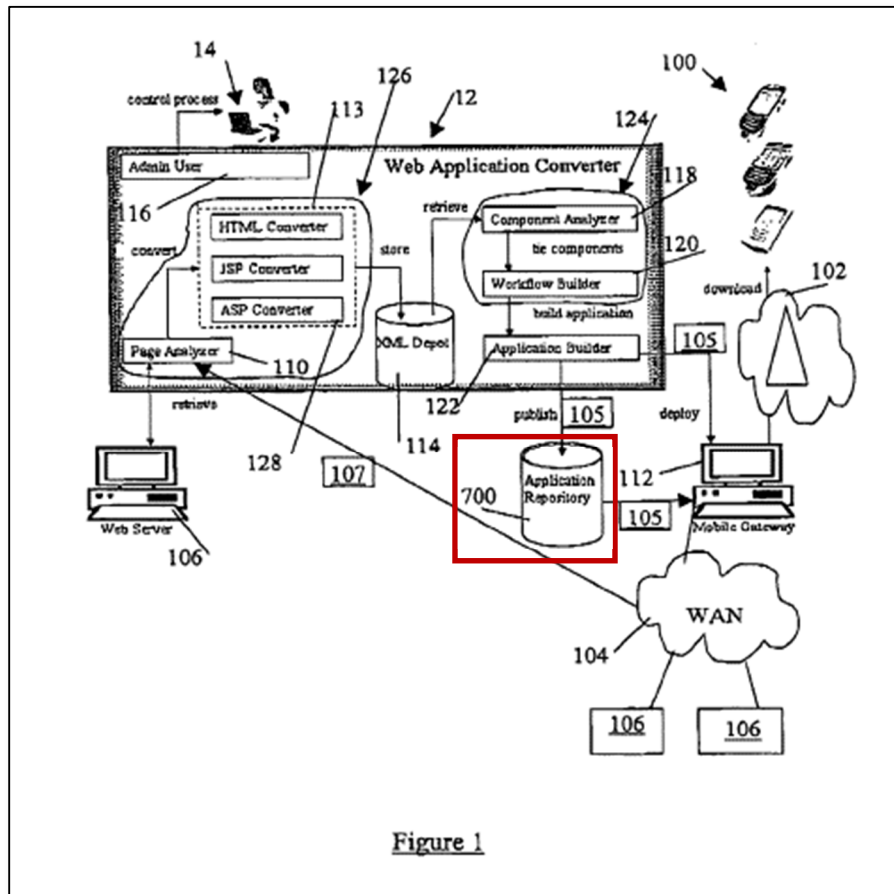
Huang discloses a *computer memory*, such as **database 124**, where “Applications ... are stored.” (Ex-1005, ¶[0042]; Ex-1002, ¶¶104-05.)



(Ex-1005, Fig. 1A.)

Shenfield discloses a registry that can be stored in a similar computer memory—
“**application**” or “**metadata repository 700**.” (Ex-1007, ¶¶[0025] (“web service
106 can ... implement an interface such as expressed using Web Services
Description Language (WSDL) registered in a Universal Discovery Description and
Integration (UDDI) services registry”), [0044] (“component applications 105 can be
hosted in a Web Service 106 registry in a metadata repository 700”).) A POSITA
would have understood that the *registry* disclosed in *Shenfield*, such as the UDDI

registry, is a structured set of data (a database) relating to accessing web services (computer functionality). (Ex-1002, ¶106.)



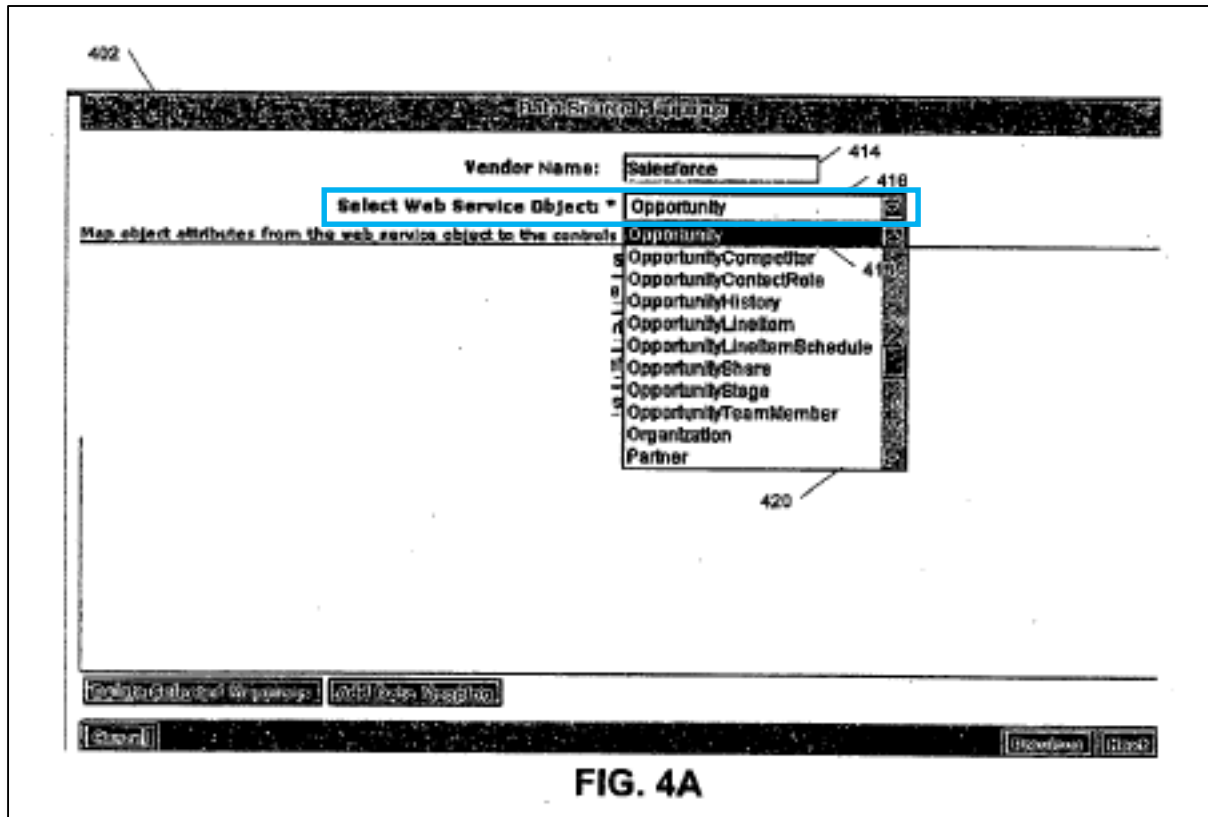
(Ex-1007, Fig. 1.)

As discussed further in Section IX.A.1.b(v), a POSITA would have found modifying *Huang's* system to store *Shenfield's* registry on *Huang's* computer memory (database 124) obvious. (Ex-1002, ¶¶107-08.) A POSITA would have found storing *Shenfield's* registry in *Huang's* database 124 an obvious modification because *Shenfield* discloses storing its registry in an analogous computer memory (application repository 700). (Ex-1007, ¶[0044].) (Ex-1002, ¶107.) Specifically,

database 124 stores substantially similar data (*Huang*'s applications) to application repository 700 (component applications), perform substantially similar functions (receive code from the system and provide it to the user), and is connected to the system in a substantially similar fashion (a stand-alone component in communication with both the system and the end user). (Ex-1007, ¶108; *compare* Ex-1005, ¶[0042] & Fig. 1a *with* Ex-1007, ¶[0044] & Fig. 1.)

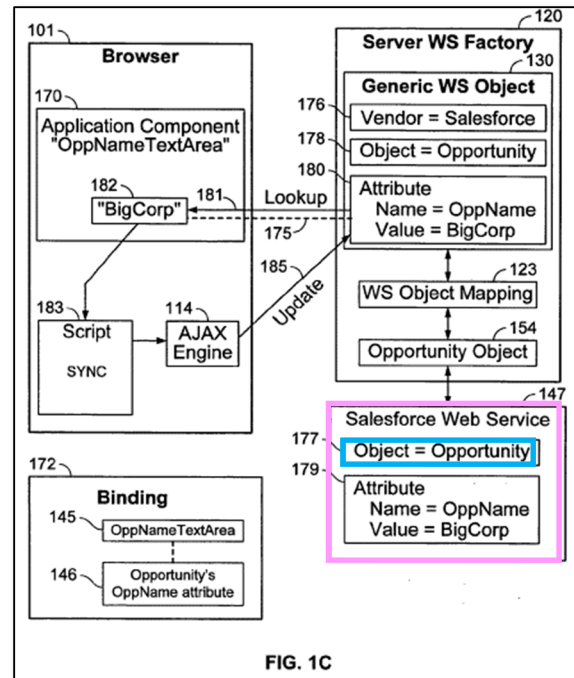
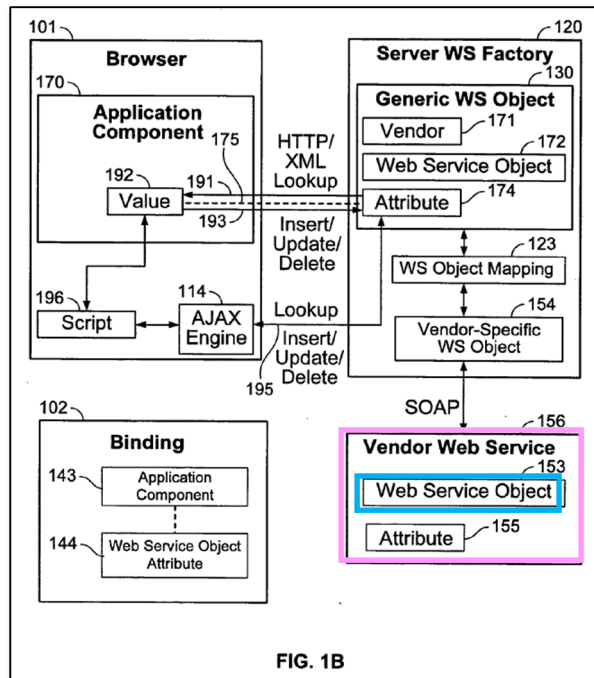
- (ii) “[a registry of]: a) symbolic names required for evoking one or more web components each related to a set of inputs and outputs of a web service obtainable over a network,”

Huang discloses *web services* and *web components*. (Ex-1002, ¶¶110-13.) *Huang* explains that web services are “computer application (software) component[s] accessible over open protocols” that “are intended for purposes of interoperability” (Ex-1005, ¶[0003]) and “use open XML standards and transport protocols to exchange data with calling clients (programs)” (*Id.*, ¶[0005]); accordingly, *Huang* discloses that its “web services” are software systems that support interaction between devices over a network. (Ex-1002, ¶110.) *Huang* discloses identifying individual web service functionalities as a “list of web service object instances,” such as the “Opportunity object type that represents a sales opportunity” **web service object** from the Salesforce website. (*E.g.*, Ex-1005, ¶¶[0044], [0109]; *see also id.* ¶[0043].)



(Ex-1005, Fig. 4A.)

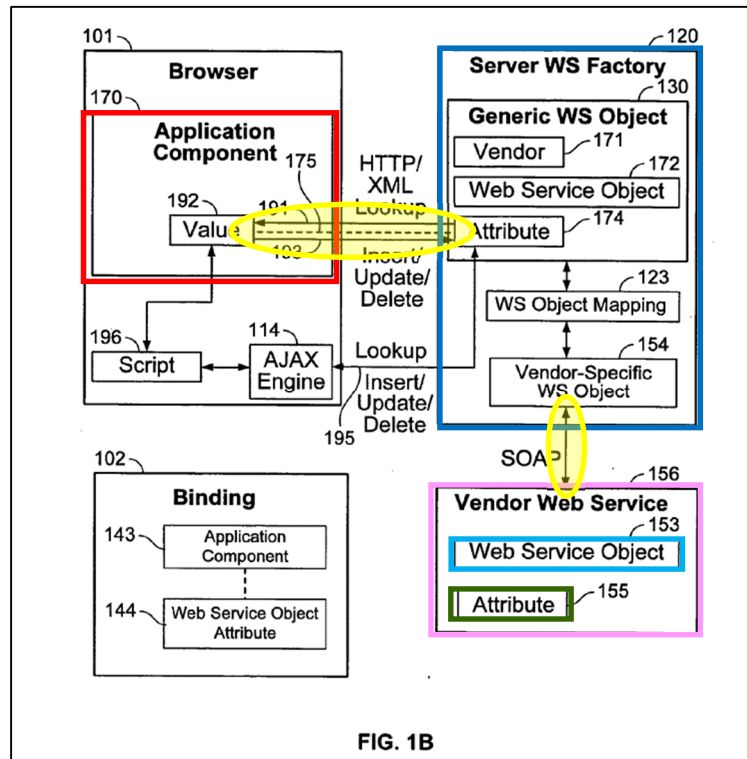
A POSITA would have understood that *Huang*'s **web objects** are software objects that provide individual functionalities of a web service to carry out specific tasks within a web service and, therefore, are *web components* of **web service 156**, such as **Salesforce Web Service 147**. (Ex-1002, ¶¶111-13.)



(Ex-1005, Figs. 1B, 1C.)

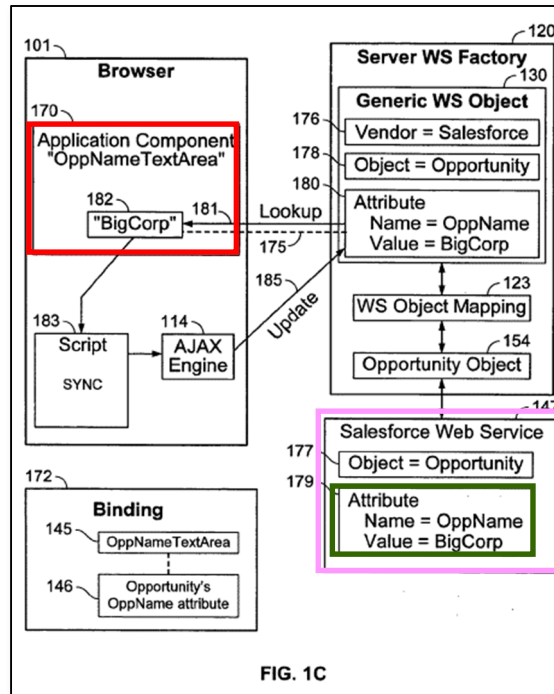
Huang further discloses that its *web components* are related to *inputs and outputs of the web service*. *Huang* discloses that each web service object has attributes, such as the “Opportunity object” having the “OpportunityName and OpportunityId” attributes (Ex-1005, ¶[0047]), and that “web service attribute 155 can be set to the value 192 of the application component 170,” (provide an *input* to the web service) and, “conversely, that the value of the application component 170 can be set to the value of the attribute 155” (obtain an *output* from the web service) via “lookup, insert, update, delete, or user-defined interaction[s].” (Ex-1005, ¶¶[0056], [0067]; Ex-1002, ¶114.) The arrows in Figure 1B below show the two-way dataflow between **attribute 155** of **web service object 153** (a part of **web**

service 156) and **application component 170**—input to or output from a web service (Ex-1002, ¶114)—via **server 120**:



(Ex-1005, Fig. 1B.)

Huang also shows an example of the **application component** “OppNameTextArea” being set to the value of the corresponding **attribute** (“OppName”) output from the **Salesforce web service** in a “lookup” interaction (Ex-1002, ¶¶115-16):



(Ex-1005, Fig. 1C; *see also id.*, ¶¶[0059], [0068].)

[0139]

APPENDIX D

```

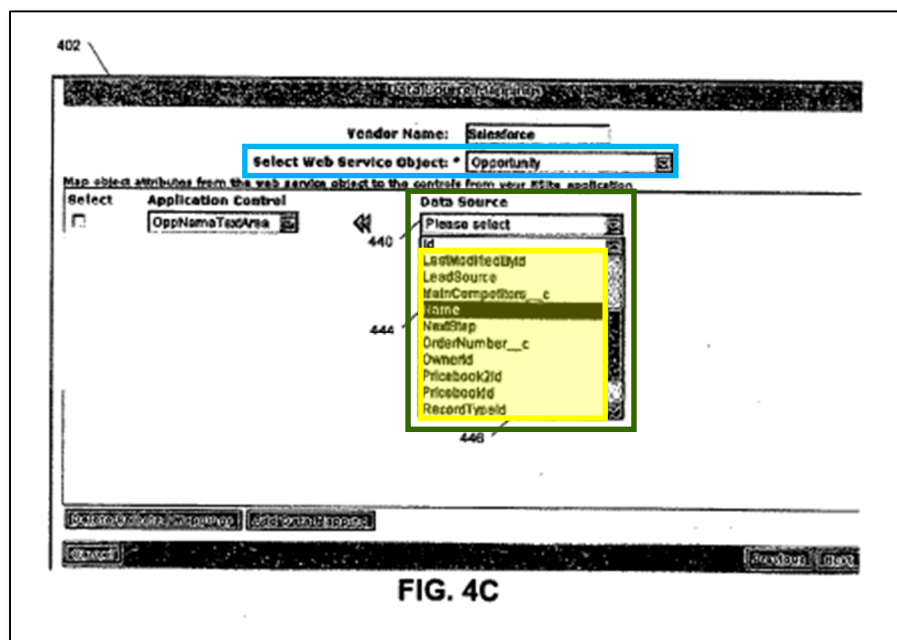
<WebServiceBinding vendorname="Salesforce"
  objectname="Opportunity"
  username="" wsurl="https://www.salesforce.com/services/Soap/u/6.0 ">
  <WSAttributeBindingList>
    <WSAttributeMapping formentity="OppNameTextArea"
      wsattribute="OppName"/>
    <WSAttributeMapping formentity="DescriptionTextArea"
      wsattribute="Description"/>
    <WSAttributeMapping formentity="StageTex.Area"
      wsattribute="StageName"/>
    <WSAttributeMapping formentity="ClosedCheckbox"
      wsattribute="IsClosed"/>
  </WSAttributeBindingList>
  <WSFilterBindingList>
    <WSFilterField filterfieldwsattribute="OppName"
      formentity=""
      filterfieldtype="Dropdown"/>
    <WSFilterField filterfieldwsattribute="CloseDate"
      formentity=""
      filterfieldtype="TextBox"/>
  </WSFilterBindingList>
  <WSTableBindingList>
    <WSLookupTableColumn wsobjattribute="OppName"
      label="Name" sortable="false"/>
    <WSLookupTableColumn wsobjattribute="Description"
      label="Description" sortable="false"/>
    <WSLookupTableColumn wsobjattribute="CloseDate"
      label="CloseDate" sortable="false"/>
    <WSLookupTableColumn wsobjattribute="CreatedById"
      label="CreatedById" sortable="false"/>
    <WSLookupTableColumn wsobjattribute="CreatedDate"
      label="CreatedDate" sortable="false"/>
    <WSLookupTableColumn wsobjattribute="IsClosed"
      label="IsClosed" sortable="false"/>
  </WSTableBindingList>
</WebServiceBinding>

```

(Ex-1005, ¶[0139].)

From these disclosures, a POSITA would have understood that *Huang*'s web objects (*web components*) are related to the *inputs and outputs of a web service*. (Ex-1002, ¶117.)

Huang discloses that its web objects (*web components*) and their attributes (*inputs and outputs of a web service*) are identified by *symbolic names*. *Huang* discloses that each *web component* (e.g., the “Opportunity” web object in *Huang*) is associated with a plurality of attributes (e.g., the “OpportunityName” and “OpportunityId” attributes in *Huang*), and these web objects and attributes are identified by *symbolic names*. (Ex-1005, ¶[0047].) For example, *Huang* describes that the “Opportunity” **web object** has a list of “Data Sources” in Figure 4C, from which a user “select[s] ... a web service **attribute**” (Ex-1005, ¶¶[0106]-[0108]):



(Ex-1005, Fig. 4C.)

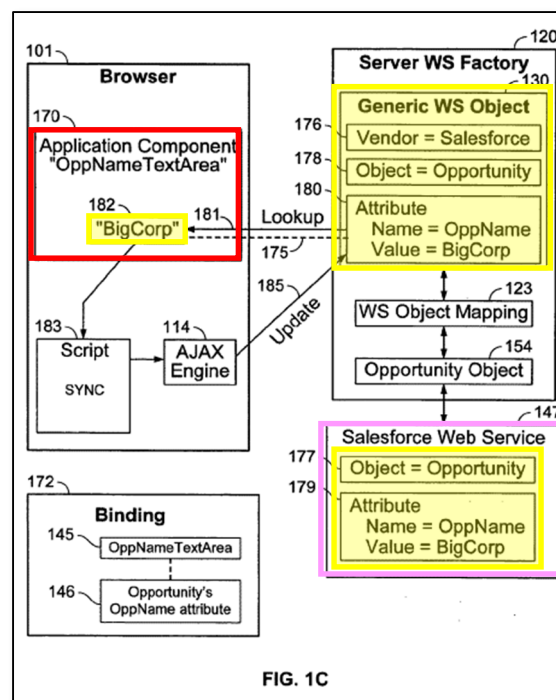
A POSITA would have understood that the names of **web objects**, such as “Opportunity,” and the names of “Data Sources”/**attributes**, such as “Name” and “NextStep,” are *symbolic names* because they include symbols, such as letters of the alphabet in a character string, that relate to the multiple **web objects** provided by a **web service**. (Ex-1002, ¶119.)

Huang explains that the *symbolic names* of the web object attributes *are required for evoking one or more web components each related to a set of inputs and outputs of a web service*. *Huang*’s application components “can be configured to invoke a web service.” (Ex-1005, ¶[0073]; *see also id.*, ¶¶[0049] (“perform web service invocations”), [0067] (“invoke web services”); [0074] (“invoking the web service object”).) *Huang* explains that these *symbolic names* for web objects and their attributes (*e.g.*, “Opportunity,” “OpportunityName,” and “OpportunityId”) serve as necessary intermediaries between the application components and the web service inputs and outputs.

[C]all[ing] vendor-specific objects directly from applications ... ties the application to a particular web services vendor. Therefore a generic web services object is introduced. ***Each vendor-specific object is mapped to the generic web services object*** by a definition stored in

an XML mapping file. *Applications interact with the generic web services object* and need not contain hard-coded dependencies on vendor specific objects.

(Ex-1005, ¶[0074]; *see also id.*, ¶[0075] (“The WS Object class represents each attribute of a vendor-specific Web Service object as a vendor-independent generic object.”).) The necessity of the symbolic names in the interactions between the **application** and the **web service** inputs and outputs can be seen (indicated in yellow) in the figure below:

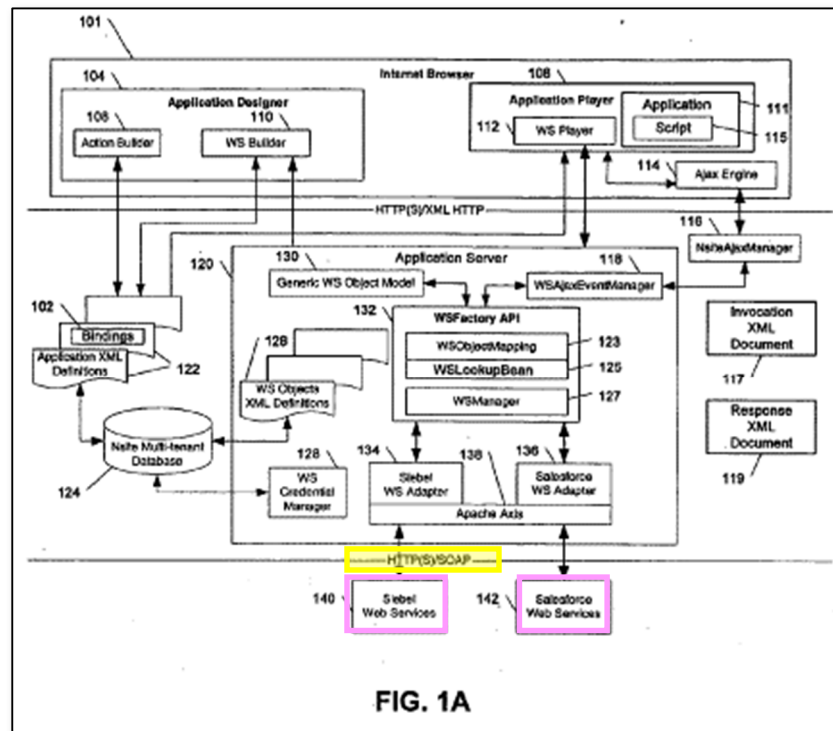


(Ex-1005, Figs. 1C; *see also id.*, ¶[0068].)

In view of *Huang*’s disclosures, a POSITA would understand that the *symbolic names* of *Huang*’s system, such as “an object named Opportunity with attributes

named OpportunityName and OpportunityId” (*id.*, ¶[0047]), are *required* for Huang’s system to call or evoke web services. (Ex-1002, ¶¶120-22.)

Huang also discloses that web service 156 is *obtainable over a network*. (Ex-1005, ¶[0043] (“The server 120 communicates with the web services via a network protocol...”); *see also id.*, ¶[0005], Fig. 1A (showing **web services** accessed via “HTTP(S)/SOAP”); Ex-1002, ¶123.)

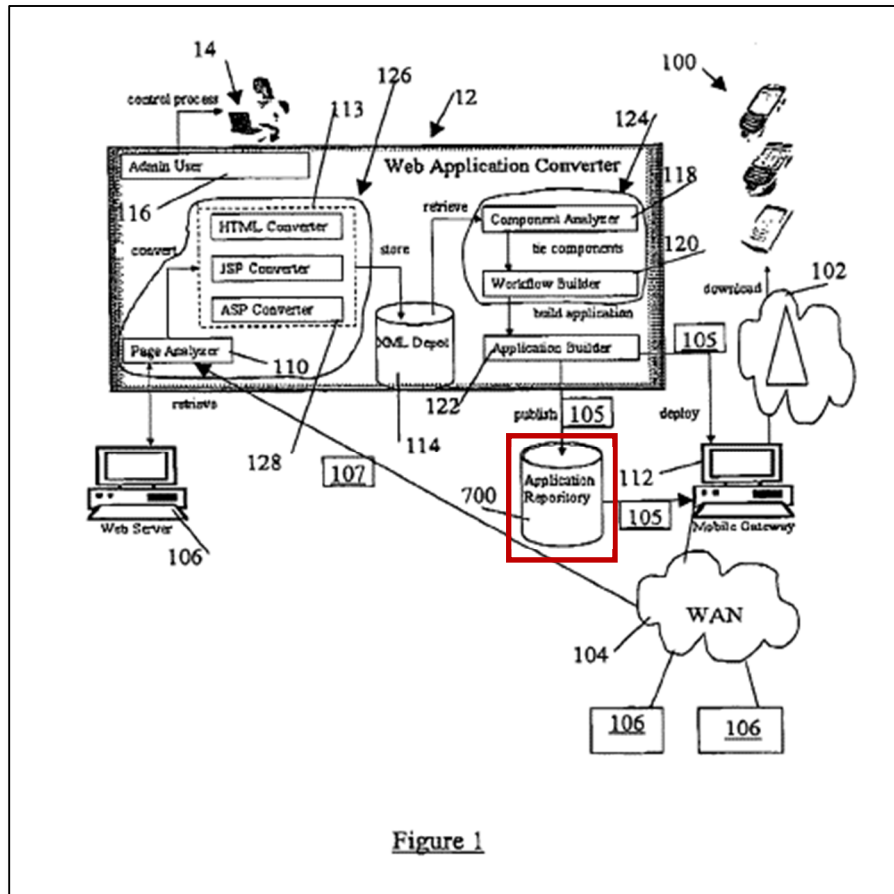


(Ex-1005, Fig. 1A.)

Indeed, a POSITA would have understood that a *web* service is obtainable over the World Wide Web (*a network*). (Ex-1002, ¶124.)

To the extent Huang does not explicitly disclose utilizing *a registry* as recited in element 1.a, a POSITA would have found element 1.a obvious over Huang in

view of *Shenfield*. *Shenfield* discloses utilizing a web service registry, for example, to bind its user interface components with web services. (Ex-1002, ¶126.) *Shenfield* explains that “component applications 105 ... communicate with external systems such as the web service 106” to utilize the “relevant business logic (methods) of the web services 106,” including via “an interface such as expressed using Web Services Description Language (WDSL) registered in a Universal Discovery Description and Integration (UDDI) services **registry** ... through an appropriate protocol such as, but not limited, to the Simple Object Access Protocol (SOAP).” (Ex-1007, ¶¶[0024]-[0025], [0042].) *Shenfield* also discloses that its component applications may be accessed via “a **Web Service 106 registry** in a **metadata repository 700** (see FIG. 1) as a bundle of platform-neutral data 400, message 404, workflow 406 component descriptors with a set of platform-specific presentation component 402 descriptors for various pre-defined client runtimes (i.e. specific runtime environments 206—see FIG. 2).” (*Id.*, ¶[0044].)



(Ex-1007, Fig. 1.)

Accordingly, a POSITA would have understood that *Shenfield* discloses that a web service registry provides the requisite information to interface with a web service (e.g., inputs, outputs, and communication structure) and discloses using a registry to obtain vendor-specific objects that would map to, for example, *Huang*'s generic objects. (Ex-1002, ¶129.)

As discussed further below in Section IX.A.1.b(v), a POSITA would have found it obvious to include *a registry* containing *Huang*'s *symbolic names*, which *Huang* explains are *required for evoking one or more web components each related*

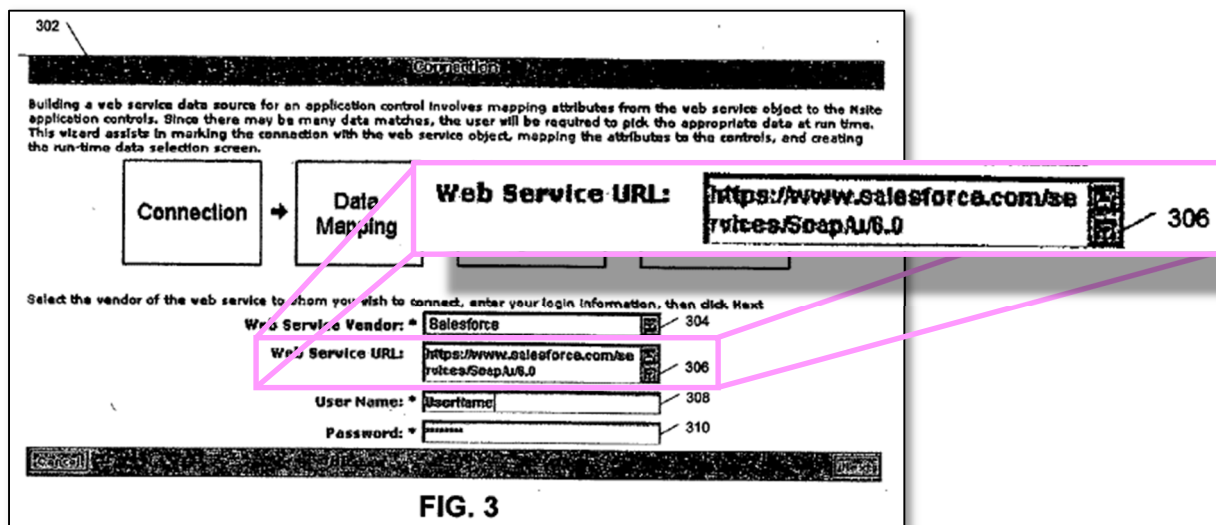
to a set of inputs and outputs of a web service obtainable over a network. For example, a POSITA would have looked to *Shenfield* because it addressed the same or similar problem, would have been motivated to apply its teachings to *Huang* to improve system performance, and would have had a reasonable expectation of success due to the similarity and compatibility between the systems of *Huang* and *Shenfield*. (Ex-1002, ¶130.) The resulting *Huang-Shenfield* system would have satisfied each requirement of element 1.a, including *a computer memory storing a registry of Huang’s symbolic names for its web service objects (web components) and their attributes (inputs and outputs), as well as the address of the web service.*

- (iii) “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”

Huang also discloses that its *symbolic names are character strings that do not contain either a persistent address or pointer to an output value accessible to the web service.* *Huang’s* symbolic names, including “Opportunity,” “OpportunityName,” and “OpportunityId” (Ex-1005, ¶[0047]), do not identify a network or memory location where data may be found but serve only as identifiers; therefore, these *symbolic names are character strings that do not contain a persistent address or a pointer.* (Ex-1002, ¶131.)

- (iv) “[where the registry includes:] b) the address of the web service”

The *Huang-Shenfield* system renders obvious a registry including *the address of the web service*. *Huang* discloses, for example in Figure 3, that an address of **web service 156** is used to locate the web service—in particular, a “URL [that] refers to a web service provided by a particular vendor” (Ex-1005, ¶[0077]; *see also id.* ¶[0105] (“a URL 306 of the web service”)):



(Ex-1005, Fig. 3.)

Similarly, *Shenfield* discloses the UDDI service registry. (Ex-1007, ¶[0025].) A POSITA would have understood that such UDDI registry includes the address of a web service. (Ex-1002, ¶134; Ex-1018, 89; Ex-1019, ¶[0006].)

As discussed in IX.A.1.b(v), a POSITA would have found a registry including *the address of the web service* in the registry of the *Huang-Shenfield* system obvious. As discussed below, the address of a web service is information necessary to access

a web service and is included in well-known web service registries; thus, a POSITA would have been motivated to—and found it necessary to—include the address of the web service in the registry and expect that doing so would not interfere with the operation of the system because the address is standard information included in registries. (*Id.*; Ex-1002, ¶135.)

(v) A POSITA Would Have Been Motivated to Modify *Huang* with *Shenfield*’s Registry with a Reasonable Expectation of Success

A POSITA would have looked to *Shenfield* because it is in the same field and pursues the same objective (cross-functionality) as *Huang*. *Shenfield*, like *Huang*, relates to the “communication of services over a network.” (Ex-1007, ¶[0001]; *see also* Ex-1005, ¶[0002] (“This invention relates to ... web services.”).) Both address the problem of web-service cross-functionality across multiple platforms. (*See* Ex-1007, ¶[0003] (stating need for programs “that can be run on client devices having a wide variety of runtime environments”), ¶[0005] (same); Ex-1005, Abstract (“allow use of web services across organizations”), ¶[0003] (“integrating various software applications with one [an]other”).) A POSITA would have recognized that *Shenfield* and *Huang* are in the same technical field and address substantially similar problems as the ’755 patent and, therefore, would have looked to and considered the teachings of *Shenfield* when implementing *Huang*’s system. (Ex-1002, ¶136.)

A POSITA would have found *Shenfield*'s disclosures of utilizing a web-services registry relevant to and compatible with implementing *Huang*'s system. (Ex-1002, ¶137.) For example, *Huang* discloses generating mapping files (or adaptors) that relate its "Generic WS Object Model 130" to vendor-specific objects by "access[ing] a WSDL (Web Service Definition Language) description for each vendor" via, for example, SOAP. (Ex-1005, ¶¶[0008], [0044]-[0045], [0077]; Ex-1002, ¶137.) A POSITA would have understood that *Shenfield* discloses accessing the same information "in a Universal Discovery Description and Integration (UDDI) services registry" and utilizing the same protocol "Simple Object Access Protocol (SOAP)" to allow "client devices 100" to communicate with the web service (Ex-1007, ¶[0025]) and, therefore, *Huang*'s system could be readily modified to utilize a web services registry, such as the UDDI registry disclosed in *Shenfield*. (Ex-1002, ¶137.) A POSITA would further understand that *Shenfield* discloses storing a registry in a computer memory (i.e., application repository 700) (Ex-1007, ¶[0044]) that stores similar data, performs similar functions, and is connected to the system and end-users in a similar manner, such that it is analogous to *Huang*'s database 124. (Ex-1002, ¶138.)

A POSITA would have understood that the *registry* of the modified *Huang-Shenfield* system would include *Huang*'s web components (i.e., *Huang*'s "Opportunity" object in the "Generic WS Object Model 130", discussed in Section

IX.A.1.b(ii)), the *symbolic names* of the inputs and outputs of web services (i.e., *Huang*'s "OpportunityName" and "OpportunityId" attributes for the "Opportunity" object, also discussed in Section IX.A.1.b(ii)), and their addresses. As discussed in Section IX.A.1.b(ii), *Huang* explains that its symbolic names for web service objects (*web components*) and their attributes (*inputs and outputs*) are necessary intermediaries to interact with web services. Further, a POSITA would have recognized that an address of the web service, such as the URL, would be a necessary piece of information to access the specified web service. (Ex-1002, ¶¶139-40; *see also* Ex-1018, 89 (listing "the network address" as a necessary piece of information").) Indeed, *Huang* confirms such information is necessary, including the symbolic names and address of the web service in its exemplary binding. (Ex-1005, ¶[0139].)

APPENDIX D

```
<WebServiceBinding vendorname="Salesforce"
  objectname="Opportunity"
  username="" wsurl="https://www.salesforce.com/services/Soap/u/6.0 ">
  <WSAttributeBindingList>
    <WSAttributeMapping formentity="OppNameTextArea"
  wsattribute="OppName"/>
    <WSAttributeMapping formentity="DescriptionTextArea"
  wsattribute="Description"/>
    <WSAttributeMapping formentity="StageTexArea"
  wsattribute="StageName"/>
    <WSAttributeMapping formentity="ClosedCheckbox"
  wsattribute="IsClosed"/>
  </WSAttributeBindingList>
```

(Ex-1005, ¶[0139]; *see also id.*, ¶[0059].)

A POSITA would have understood that the registry must contain all necessary information, such as the binding information used in *Huang*'s system, to access a web service and, therefore, would have found including in the registry *Huang*'s symbolic names and web-service addresses obvious. (Ex-1002, ¶¶139-42.)

Indeed, a POSITA would have been motivated to modify the system of *Huang* to utilize *a registry of* web components (e.g., *Huang*'s "Opportunity" object in the "Generic WS Object Model 130"), the *symbolic names of* the inputs and outputs of web services (e.g., *Huang*'s "OpportunityName" and "OpportunityId" attributes for the "Opportunity" object), and their addresses. For example, doing so would improve system performance by eliminating the need to analyze each web service for web components and the associated inputs and outputs on a per request basis. (Ex-1005, ¶[0049] ("The WSManger 127 may be invoked by the Designer 104, e.g., to get a list of web service objects."); Ex-1002, ¶143.) A registry would further enable a user of *Huang*'s system to search for web services rather than requiring pre-existing knowledge of a specific web service, its capabilities, and its address. (*Id.*) Including the address of a web service in the registry would allow the user interface of *Huang* to maintain its functionality of using a web address as an input to identify the relevant, available web services (and the corresponding symbolic names) from the registry. (Ex-1005, ¶¶[0077], [0105], Fig. 3; Ex-1002, ¶143.)

Additionally, utilizing a registry of the symbolic names of *Huang* would have been a straightforward, conventional way to implement a system that accesses web services. (Ex-1002, ¶144.) A POSITA would have known that symbolic names were used for accessing various types of data and services. (Ex-1028, Abstract, 1:10-11, 1:39-40; Ex-1029, ¶¶[0005], [0032], [0035]; Ex-1002, ¶144.) Therefore, a POSITA would have found implementing a registry in the above manner an obvious option to simplify and streamline data processing and record-keeping relating to *Huang*'s web services. (Ex-1002, ¶144.) A POSITA would also have been motivated to store the registry in *Huang*'s database 124 to leverage already-existing data stores in *Huang*'s system. (*Id.*, ¶145.)

A POSITA would also have had a reasonable expectation of success. First, the use of registries containing symbolic names and addresses in the context of web services was common and well known. (Ex-1019, ¶¶[0005]-[0007], Figs. 1-2; Ex-1007, ¶¶[0025], [0044]; Ex-1002, ¶146; *see also* Ex-1004, 1068-69 (explaining WSDL provides symbolic names for web services).) Thus, a POSITA would have known how to implement such a registry in *Huang*'s system without disrupting system function. (Ex-1002, ¶146.) Second, *Huang*'s system utilizes data formats (SOAP and WSDL) consistent with the information available on well-known web services registries, including those disclosed in *Shenfield*, such that modifying *Huang*'s system to incorporate such a feature would have been easy for a POSITA

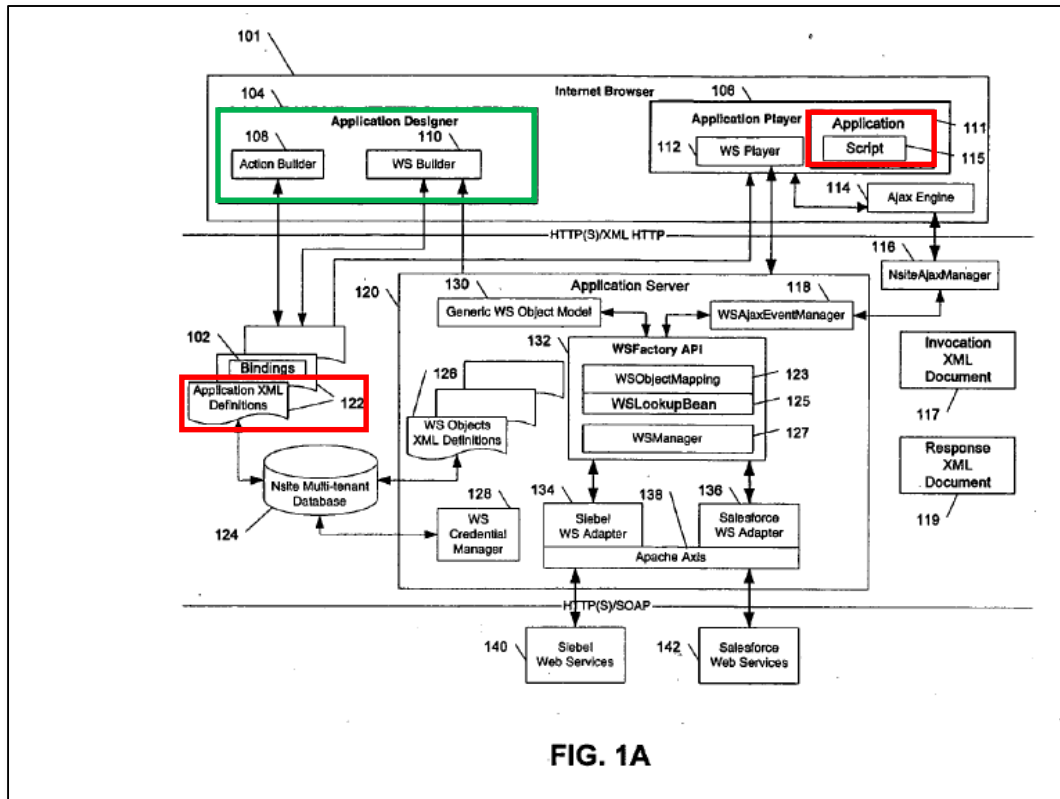
to implement. (*Id.*) Third, *Huang*'s system includes a comparable data store (database 124) to *Shenfield*'s application repository 700 that could store the registry. (*Id.*) Indeed, the above modification would have been a mere combination of known components and technologies to produce predictable results. (*Id.*) See *KSR Int'l Co. v. Teleflex Inc.*, 550 U.S. 398, 416 (2007).

- c. [1.b] “an authoring tool configured to: define a user interface (UI) object for presentation on the display, where said UI object corresponds to the web component included in said registry selected from the group consisting of an input of the web service and an output of the web service,”

Huang in view of *Shenfield* discloses or suggests element 1.b. At the district court, Express Mobile's agreed that “authoring tool configured to” means “a system, with a graphical interface, for generating code to display content on a device screen,” which the court adopted. (*Supra* Section VIII.)

- (i) “an authoring tool configured to: define a user interface (UI) object for presentation on the display,”

Huang discloses a system with **designer 104** (*authoring tool*), which “is a computer program that provides a user interface which allows a user to create the **application 111**,” which are stored as “**XML definitions 122** in a database 124” and then “loaded into Player 106 as the **application 111**” when executed. (Ex-1005, ¶¶[0040], [0042].)

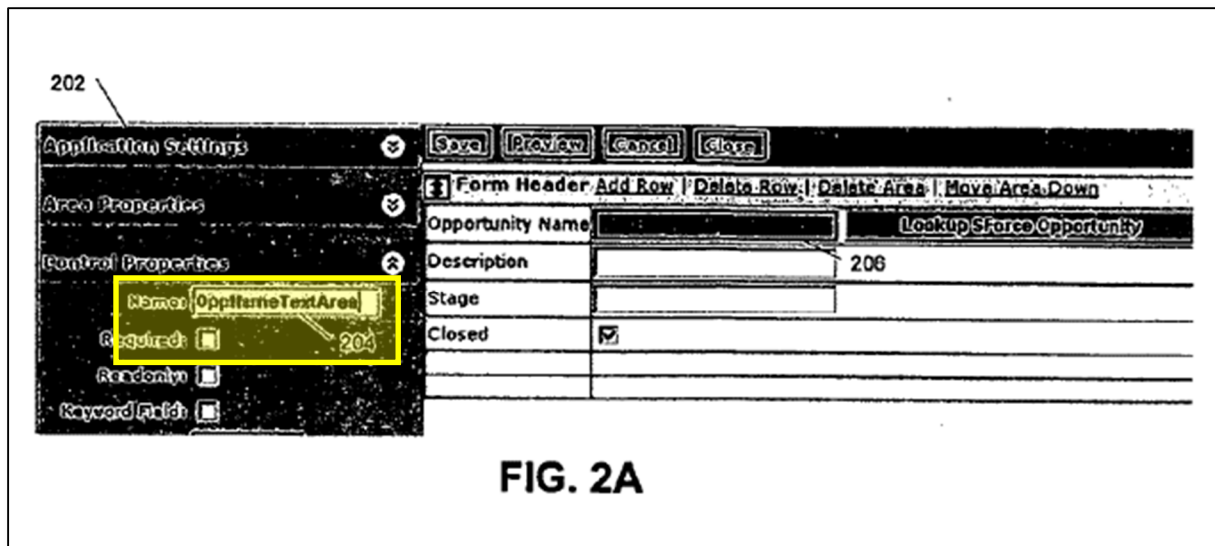


(Ex-1005, Fig. 1A.)

A POSITA would have understood from these disclosures that *Huang*'s **Designer 104** is a software tool that allows a user to generate code to display content on the display of a device (such as “**Application 111**” and “**Application XML Definitions 122**”). (Ex-1002, ¶148; *see also supra* Section IX.A.1.a.)

Huang explains that its designer is a graphical user interface that allows a user to *define user interface (UI) objects for presentation on the display* through graphical elements. (Ex-1002, ¶¶149-51.) *Huang* explains that “Designer 104 includes ... a computer program that provides a *user interface* for *defining* web services interactions in browser-based applications.” (Ex-1005, ¶[0054].) Specifically,

Huang discloses that its designer allows for defining “application components” that are displayed as objects for user interaction, such as the “OppNameTextArea” described in Figure 2A, through visual interfaces. (E.g., Ex-1005, ¶¶[0103] (“FIG. 2A is an illustrative drawing of defining an application component in an application Designer. ... The application component 206 is a text input component and has an associated name, 204, OppNameTextArea.”).)



(Ex-1005, Fig. 2A.)

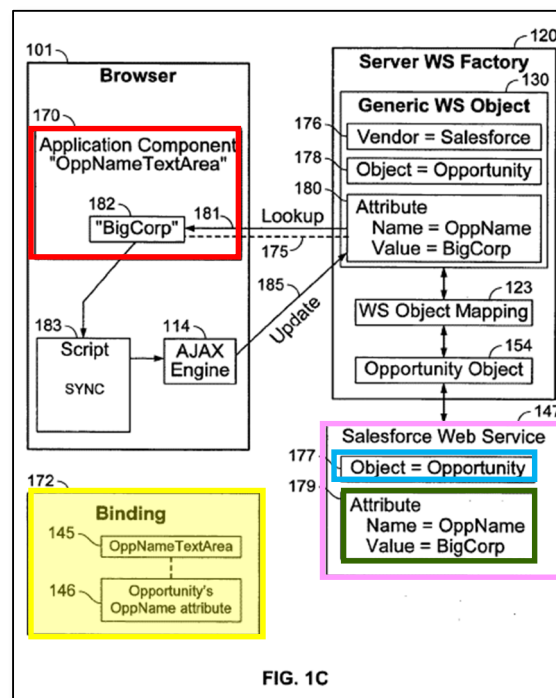
Huang further discloses, for example, buttons with which a user may “press” and, therefore, necessarily is displayed to a user. (Ex-1005, ¶¶[0118] (“The application 1102 includes a Lookup SForce Contact button 1104, which was defined using the application Designer.... A user of the application 1102 can press the Lookup SForce Contact button 1104 to display a Selection Lookup Table for retrieving values for the application components.”), [0070] (describing exemplary “lookup

interactions”), [0073] (discussing “button” components “configured to invoke a web service”).) A POSITA would have understood that *Huang*’s designer is a *graphical user interface* because it allows a user to author code through interactive graphical elements and that *Huang*’s application components, such as the “OppNameTextArea” text input field and “buttons,” are user interface objects because they allow for user interaction (e.g., input text or activate by clicking) and are displayed and presented to the user. (Ex-1002, ¶¶149-51.)

- (ii) “where said UI object corresponds to the web component included in said registry selected from the group consisting of an input of the web service and an output of the web service,”

Huang discloses that an application component (*UI object*) *corresponds to the web component included in the registry of the Huang-Shenfield system selected from the group consisting of an input of the web service and an output of the web service.* (Ex-1002, ¶¶152-57.) For example, *Huang* discloses that the “OppNameTextArea” application component may invoke the bound “Opportunity” web service object to provide an input to (e.g., insert or modify) or get an output from (e.g., lookup) the web service. (E.g., Ex-1005, ¶¶[0060] (“[T]he OppNameTextArea application component is bound to the OppName web service attribute.”), [0103] (“The application component 206 is a text input component and has an associated name 204, OppNameTextArea.”); *see also id.*, ¶¶[0056] (“The binding 193 ... specifies that a particular web service attribute 155 can be set to the value 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 ... as part of a web service interaction such as a lookup, insert, update, delete, or user-defined interaction.”.) This *correspondence* between the “OppNameTextArea” text input **application component 170** (UI object) and the “OppName” **attribute 179** (input and output of a web service) of the “Opportunity” **web object 177** (web component) from the Salesforce **web service 147** can be seen in, for example, Figure 1C below, which shows a “lookup” operation and binding 172 between “OppNameTextArea” and “Opportunity’s OppName attribute”:

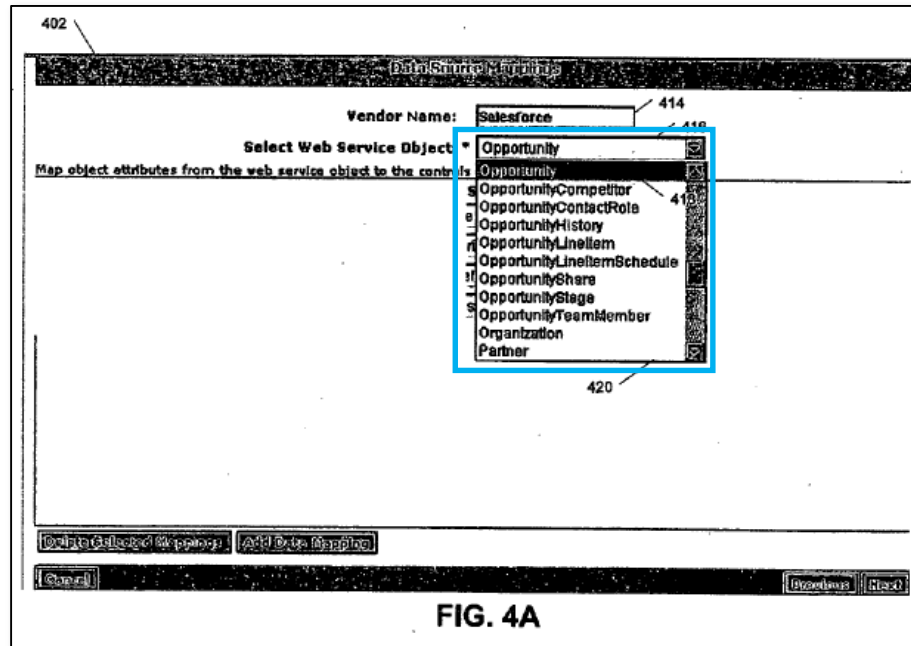


Accordingly, the application components of *Huang* are *UI objects that correspond to a web component selected from the group consisting of an input of the web service and an output of the web service*. (Ex-1002, ¶¶152-57.)

Thus, the combined *Huang-Shenfield* system discloses that the application component *corresponds* to the web service object (“*web component*”) *included in the registry* of the combined system (“*said registry*”) *selected from the group consisting of an input of web service 156 (“the web service”) and an output of web service 156 (“the web service”)*. (*Id.*)

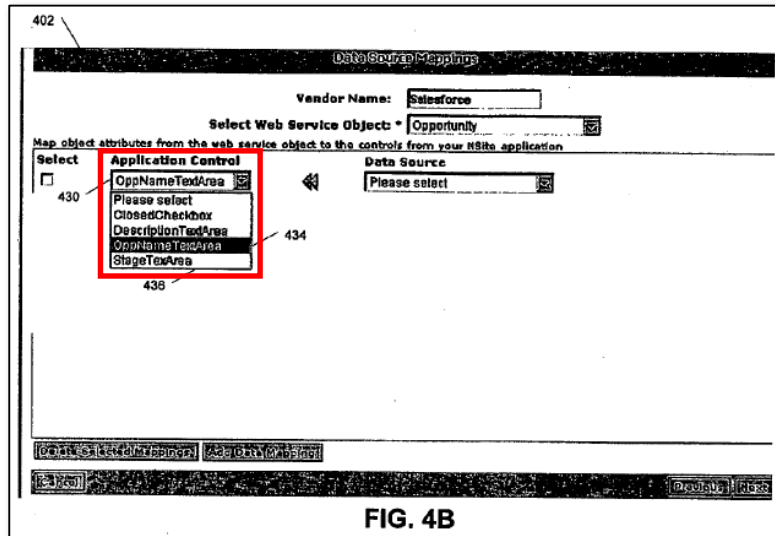
d. [1.c] “[authoring tool configured to:] access said computer memory to select the symbolic name corresponding to the web component of the defined UI object,”

Huang discloses *selecting the symbolic name corresponding to the web component of the defined UI object* in Figures 4A-4C. In Figure 4A of *Huang*, designer 104 “allows a user to choose a selected web service object 416,” which is a **web component** represented by symbolic names (e.g., “Opportunity”) from “selection menu 420” (Ex-1005, ¶[0106]):



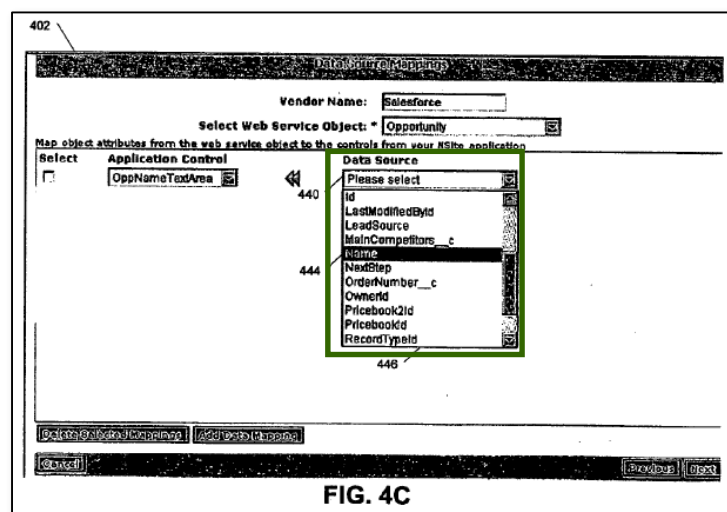
(Ex-1005, Fig. 4A.)

Upon selecting the “Opportunity” web service object, *Huang* discloses that the “user can choose an **application component 420** from the selector 436,” such as a text area of a user interface (“OppNameTextArea”), which is the *defined UI object* (Ex-1005, ¶[0107]):



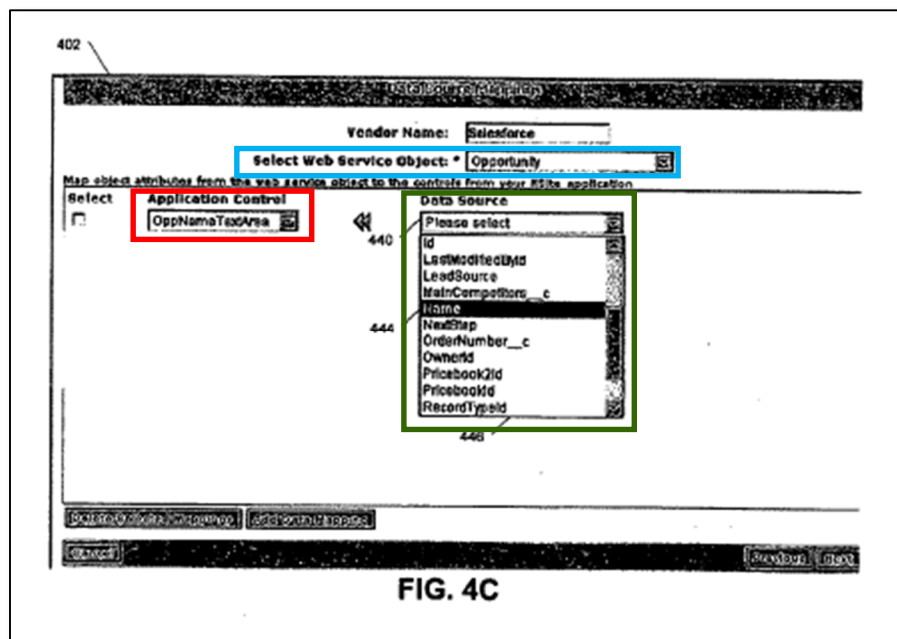
(Ex-1005, Fig. 4B.)

Huang discloses that the user then “select[s] ... a web service **attribute** using ... the web service attribute selector 440,” such as the **attribute** identified by the *symbolic name* “Name,” and that “[o]nce an application component and web service attribute have been selected, a binding that links the component to the **attribute** can be added to the application” (Ex-1005, ¶[0108]):

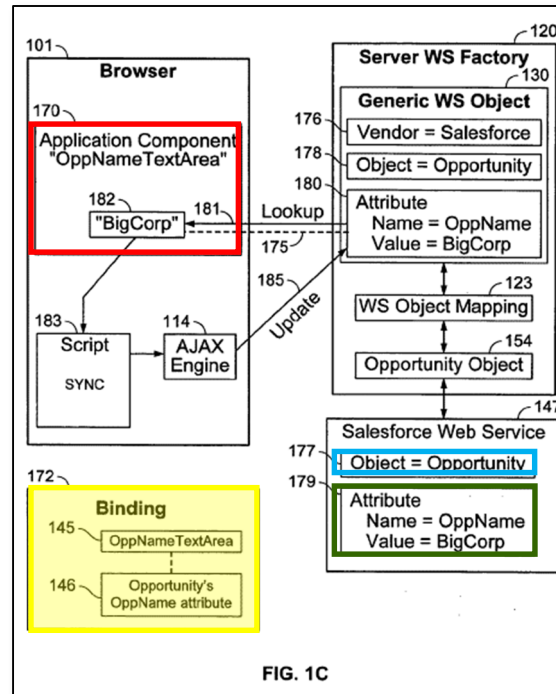


(Ex-1005, Fig. 4C.)

In this example, the user of the designer *selected* the “Name” (*symbolic name*) **attribute** (*input and output of a web service*) from the “Opportunity” **web service object** (*web component*), which is bound to and thus *corresponds to the* “OppNameTextArea” **application component** (*defined UI object*), as can be seen in Figures 4C and 1C below. (Ex-1002, ¶¶159-64.)



(Ex-1005, Fig. 4C.)



(Ex-1005, Fig. 1C.)

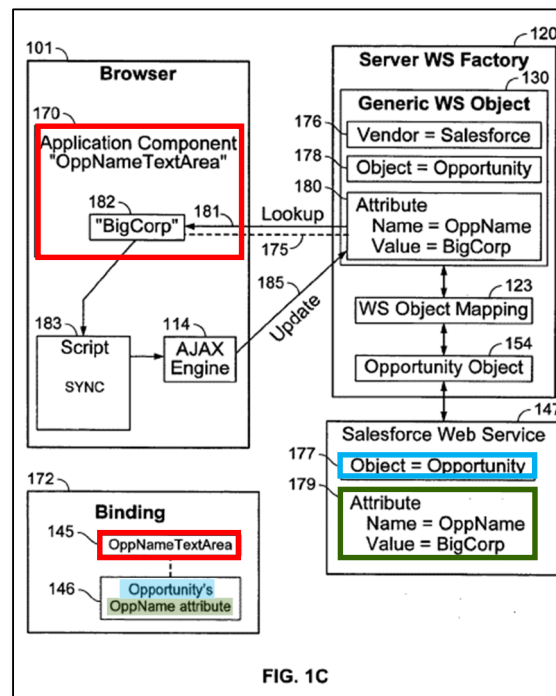
Accordingly, *Huang* discloses selecting a symbolic name from said web component corresponding to the defined UI object.

As discussed above, a POSITA would have found it obvious to modify *Huang*'s system to use the registry disclosed in *Shenfield* and store the registry in *Huang*'s database 124. (*Supra* Section IX.A.1.b.) A POSITA would have understood that *Huang*'s designer (*authoring tool*), rather than accessing each vendor web service to identify web objects (*web component*) and attributes (*inputs and outputs of the web service*) on a per request basis (*e.g.*, Ex-1005, ¶[0049]), would access the registry containing the *symbolic names* for web objects and attributes

stored on database 124 (*computer memory*) in the *Huang-Shenfield* system. (Ex-1002, ¶165.)

e. [1.d] “[authoring tool configured to:] associate the selected symbolic name with the defined UI object,”

Huang’s system creates bindings that *associate* an **application component** (*UI object*) with the *symbolic name* of an **attribute** (*input and output of a web service*) of a **web object** (*web component*) to allow for the flow of data between the web service and the UI object, as can be seen in Figure 1C below. (Ex-1002, ¶166.)

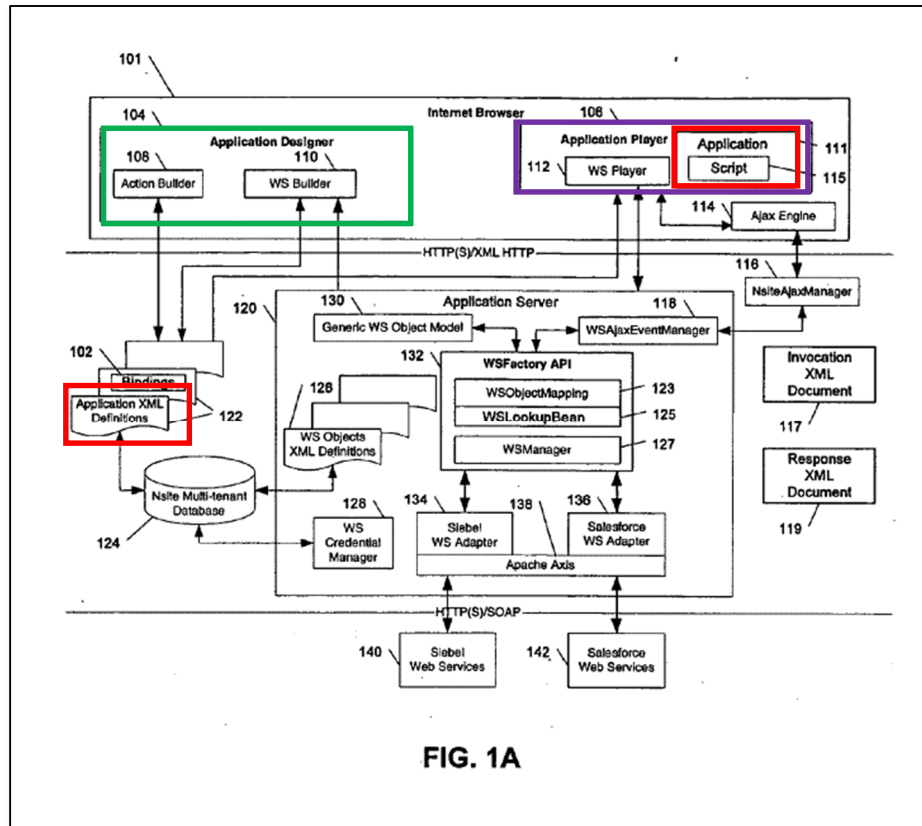


(Ex-1005, Fig. 1C.)

- f. [1.e] “[authoring tool configured to:] produce an Application including the selected symbolic name of the defined UI object, where said Application is a device-independent code, and”

Huang discloses element 1.e under the district courts’ constructions of “Application” (“device-independent code containing instructions for a device and which is separate from the Player”) and “device-independent code” (“code that is not specific to the operating system, programming language, or platform of a device”). (*Supra* Section VIII.)

Huang discloses that its designer *produces an application* and that this *application is separate from a player and is a device-independent code*. For example, Figure 1A depicts the **Designer 104** generating **application 111** (as **XML definition 122**), which is separate from and ultimately executed in **player 106**:



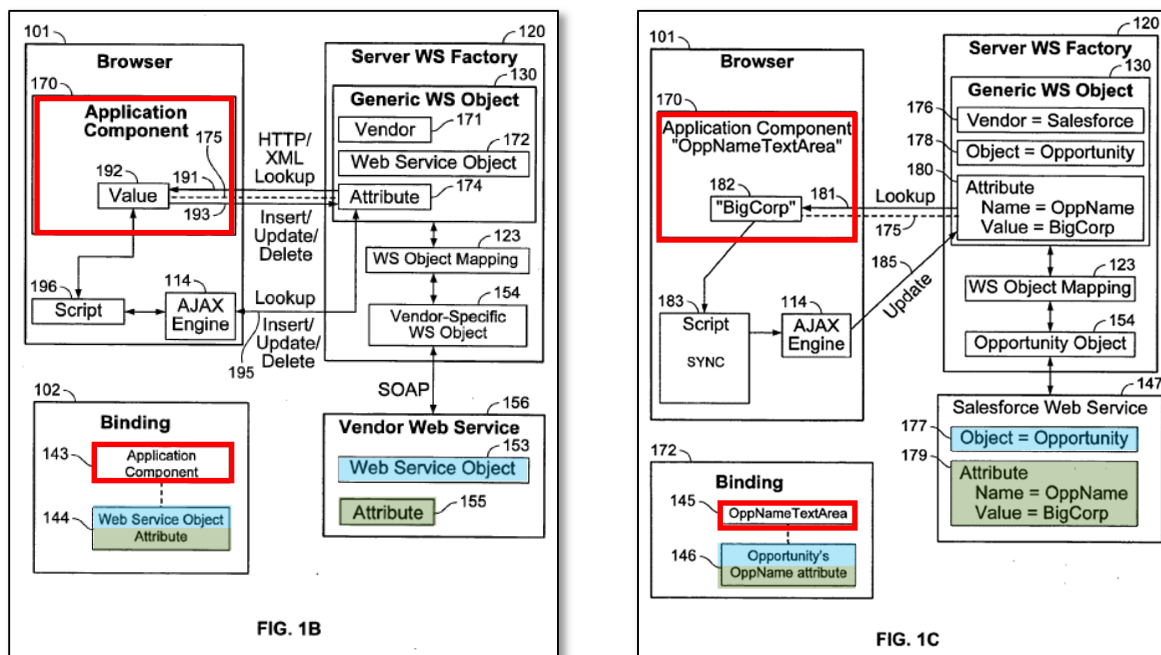
Huang explains that designer 104 allows a user to create application 111 (Ex-1005, ¶[0040]; *see also id.*, ¶[0008] (“generate composite or other software applications”)) and thus *produces the application*. (Ex-1002, ¶¶168-69.) *Huang*’s applications “are stored in Extensible Markup Language (XML) format as XML definitions 122 in a database 124.” (Ex-1005, ¶[0042].) *Nachnani*, which *Huang* incorporates by reference, confirms that such applications are written in XML. (Ex-1005, ¶¶[0006] (“See commonly owned Patent Application entitled ‘Browser Based Designer and Player,’ filed Sep. 30, 2005, inventors Pawan Nachnani et al., Ser. No. 11/241,073, incorporated herein by reference in its entirety, which describes a method of building software applications.”), [0042] (“The XML definitions are described in the

aforementioned commonly-assigned patent application[.]”); Ex-1006, ¶¶[0045], [0052], [0056], [0063], [0097], [0103]-[0104], [0106], FIG. 5C.) A POSITA would have understood that these XML applications are *device-independent code* (“code that is not specific to the operating system, programming language, or platform of a device”) (*supra* Section VIII) because, for example, a POSITA would have known that “[t]he XML Standard specifies a markup language that uses tags to specify the content and structure of the data enclosed by the tags,” and “[*d*]ata that complies with the XML Standard is platform-independent because the data’s content and structure is specified by the tags” (Ex-1026, 4:3-8; *see also id.*, 3:64-4:3 (“[D]ata types 216 are *platform-independent because they comply with the XML Standard.*”). (Ex-1002, ¶¶170-72.)

A POSITA would further have understood that *Huang’s* application 111 provides instructions for a device. As discussed above, *Huang* and *Nachnani’s* disclosures explain that *Huang’s* application includes application components (*UI objects*) that interact with web services to input or obtain values for display. (*Supra* Sections IX.A.1.a, IX.A.1.c.) A POSITA would have understood that *Huang’s* applications, therefore, provide instructions to the device regarding what content to display. (Ex-1002, ¶173.)

Huang further discloses that its applications, such as application 111, *includes the selected symbolic name of the defined UI object.* (Ex-1002, ¶¶174-77.) For

example, *Huang* explains that “a binding that links the component to the attribute can be added to the application by pressing an Add Data Mapping button.” (*E.g.*, Ex-1005, ¶[0108]; *see also id.* ¶[0042] (applications “include bindings 102 which associate application components with web service object attributes”).) For example, *Huang* discloses that the **application component** “OppNameAreaText” (*UI object*) is bound to the *symbolic names* of the **attribute** (*input and output*) (“OppName”) of a **web object** (*web component*) (“Opportunity”), as can be seen in, for example, Figures 1B and 1C:



(Ex-1005, Figs. 1B-1C.)

A POSITA would have understood from *Huang*’s disclosures that the symbolic name associated with the defined UI object is included in *Huang*’s

application. Specifically, a POSITA would have understood that bindings between an application component (here, “OppNameTextArea”) to a web service object attribute, which is represented by a symbolic name (here, “OppName”), are “add[ed]” to and “include[d]” in the application, indicating that the applications contain code specifying the symbolic names associated with the defined UI object. (Ex-1005, ¶¶[0042], [108]; Ex-1002, ¶176.) Further, a POSITA would have understood that the application must necessarily include the symbolic names of *Huang’s* web service objects and attributes so that *Huang’s* applications can “interact with the generic web services object” specified by the user because the symbolic name identifies the generic object attribute with which the applications interact. (Ex-1005, ¶[0074]; Ex-1002, ¶177.)

g. [1.f] “[authoring tool configured to:] produce a Player, where said Player is a device-dependent code;”

Huang in view of *Shenfield* renders element 1.f obvious under the district courts’ constructions of “Player” (“device-specific code which contains instructions for a device and which is separate from the Application”) and “device-dependent code” (“code that is specific to the operating system, programming language, or platform of a device”). (*Supra* Section VIII.)

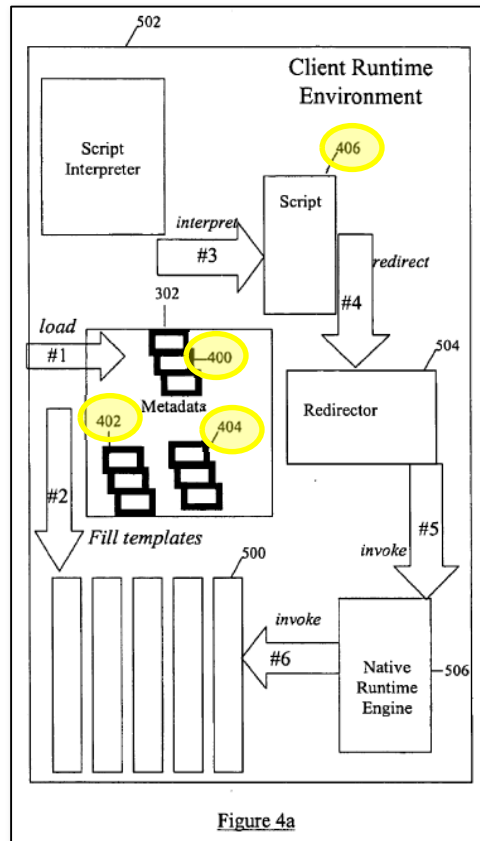
(i) *Huang* in View of *Shenfield* Discloses Element 1.f

Huang suggests that its designer 104 generates “player 106.” Specifically, *Huang’s* system is designed such that its applications are displayed by “load[ing]

[applications 111] into the Player 106.” (Ex-1005, ¶[0042].) Because *Huang* discloses that its “Player 106” is necessary to display its applications, a POSITA would have understood that *Huang*’s system must generate a player. (Ex-1002, ¶179.) Because *Huang* identifies Designer 104 as “create[ing]” applications (Ex-1005, ¶[0042]), a POSITA would have understood that *Huang* suggests that its Designer 104, which already generates code in the form of applications, could also generate “Player 106.” (Ex-1002, ¶179.)

Similarly, *Shenfield* expressly discloses a system that *produces* platform-independent content (*application*) and separate platform-dependent presentation and workflow components (*player*). (Ex-1002, ¶¶180-81.) *Shenfield* explains that its system generates component applications, including presentation components and workflow components. (Ex-1007, ¶[0059]; Ex-1002, ¶180.) “[P]resentation components 402 may vary depending on the client platform and environment of the device 100,” and *Shenfield*’s system provides “bundle[s] of platform-neutral ... component descriptors” and “platform-specific presentation component 402 descriptors for various predefined client runtimes (i.e., specific runtime environments 206—see FIG.2).” (Ex-1007, ¶[0044]; *see also id.* (“different sets of presentation components 403*a*, 403*b*, 403*c*, representing the different supported user interfaces 202 of the devices 100.”).) *Shenfield* further discloses that “workflow components 406 are written as a series of instructions” that operate “as a set of rules

for operations on the other components” and discloses “defin[ing] differing work flows based upon different supported capabilities or feature of particular devices.” (*Id.*, ¶[0045].) And *Shenfield* shows in Figure 4a, that each of data 400, presentation 402, message 404, and workflow 406 components are separate sets of code:



A POSITA would have understood that the platform-specific presentation and workflow components contain code specific to the particular device platform, provide instructions for that device, and are separate from the platform-neutral components. (Ex-1002, ¶181.)

- (ii) A POSITA Would Have Been Motivated to Modify Huang’s System with Shenfield’s Device-

Specific Components with a Reasonable
Expectation of Success

A POSITA would have looked to *Shenfield* to improve *Huang*'s system. A POSITA would have been aware of the proliferation of Internet-accessible personal devices with limited resources that made other, more resource intensive solutions insufficient. (Ex-1007, ¶¶[0002]-[0003]; Ex-1002, ¶¶183.) Thus, a POSITA would have been motivated to explore options to realize *Huang*'s suggestion of implementing its system in alternative configurations (Ex-1005, ¶[0008] (stating its system is “not so limited” to “typical” embodiments)) and would look to references that achieve the same objective as *Huang*—web-service interface interoperability—such as *Shenfield*. (Ex-1002, ¶183; Ex-1005, ¶[0003] (“Hence the need for interoperability.”); Ex-1007, ¶[0002] (“[D]eveloping software applications for a variety of devices remains a difficult and time-consuming task.”).) A POSITA would have looked to *Shenfield* in particular because it offers a solution to the precise issue of presenting web applications on myriad, less-powerful, internet-accessible devices. (Ex-1002, ¶183-184.)

Further, a POSITA would have been motivated to modify *Huang*'s Designer 104 with *Shenfield*'s solution of “platform-specific” presentation and workflow components to enable *Huang*'s applications to accommodate the resource constraints of less-powerful, internet-accessible devices. In view of *Shenfield*'s disclosures that explain that native applications “provid[e] a relatively optimized

application program for each runtime environment” over other “resource intensive” solutions, that its “component applications 105 may be executed as native code or interpreted by another software module” (which a POSITA would have understood to include web browsers), and that its invention maintains the principle “write once run everywhere” (Ex-1007, ¶¶[0005], [0029], [0056]), a POSITA would have been motivated to implement *Shenfield’s* device-specific components (*Player that is a device-dependent code*) in *Huang’s* system to achieve the same benefits and better tailor operation of its applications on a wider range of devices. (Ex-1002, ¶185.) Specifically, a POSITA would have been motivated to replace *Huang’s* players with *Shenfield’s* device-specific code (*player*) that can be tailored to provide instructions that optimize the display of content specified in *Huang’s* device-independent applications for each runtime environment, thus improving performance and user experience for a broader set of end users. (*Id.*; Ex-1007, ¶¶[0005], [0029], [0056]; *see also id.*, ¶[0054].)

Further, a POSITA would have had a reasonable expectation of success in implementing *Shenfield’s* device-dependent components to the system of *Huang*. (Ex-1002, ¶¶186-90.) Specifically, *Shenfield’s* device-specific components would have been compatible with *Huang’s* device-independent components because *Huang’s* application and *Shenfield’s* device-independent components utilize similar code (*Huang’s* XML definitions (Ex-1005, ¶¶[0008], [0042], [0059]) versus

Shenfield's "structured definition language" (Ex-1007, ¶[0040]), including XML (*id.*, ¶[0046])). (Ex-1002, ¶187.) Further, *Shenfield*'s platform-specific presentation components and device-specific workflow components would operate similarly to the functionality of *Huang*'s player, executing an application and providing instructions that are interpreted by the runtime environment. (Ex-1002, ¶188.)

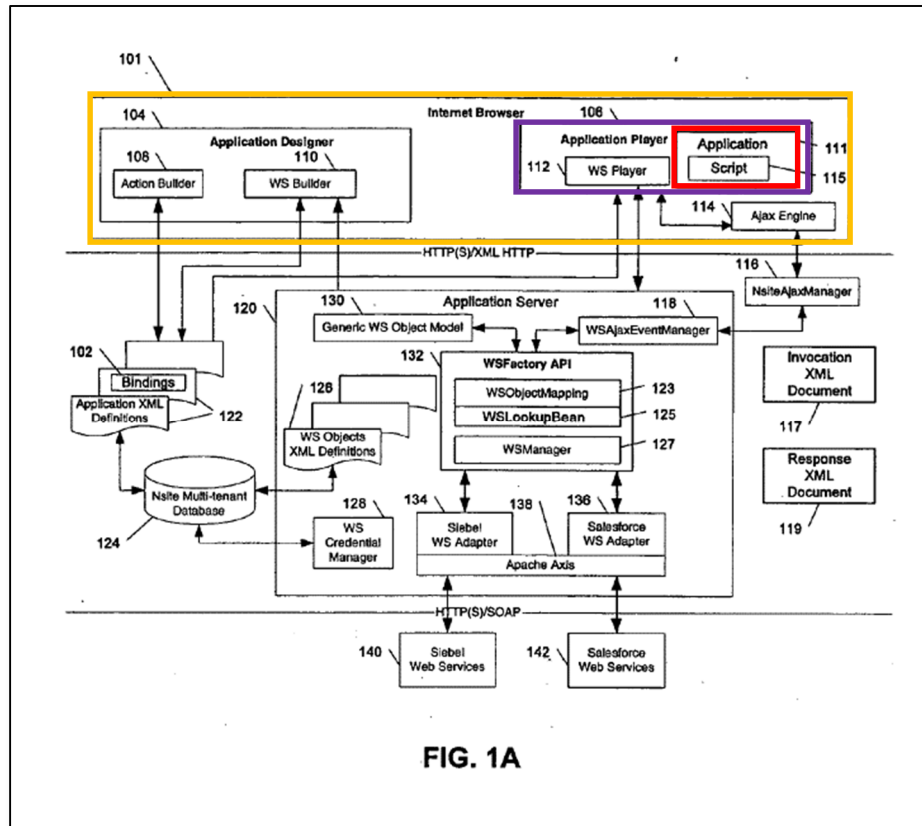
Further, in view of *Huang*'s suggestion and *Shenfield*'s disclosure of a system that generates both the application and player, a POSITA would have been motivated to produce the player using *Huang*'s designer in the *Huang-Shenfield* system. Indeed, because *Huang*'s designer already generates code, a POSITA would have found doing so beneficial because this design would efficiently leverage the same software to generate platform-specific components. (Ex-1002, ¶189.) Indeed, altering *Huang*'s system to generate a player that allows for the presentation of its application, as disclosed in *Shenfield*, would have been a simple substitution of known elements to achieve predictable results. (*Id.*)

Accordingly, a POSITA would have found *producing a Player, where said Player is a device-dependent code* obvious over *Huang* in view of *Shenfield*. (Ex-1002, ¶¶182-90.)

h. [1.g] “such that, when the Application and Player are provided to the device and executed on the device, and when a user of the device provides one or more input values associated with an input symbolic name to an input of defined UI object, 1) the device provides the user provided one or more input values and corresponding input symbolic name to the web service, 2) the web service utilizes the input symbolic name and the user provided one or more input values for generating one or more output values having an associated output symbolic name, 3) said Player receives the output symbolic name and corresponding one or more output values and provides instructions for a display of the device to present an output value in the defined UI object.”

(i) “such that, when the Application and Player are provided to the device and executed on the device, and when a user of the device provides one or more input values associated with an input symbolic name to an input of defined UI object,”

The *Huang-Shenfield* system discloses that *the Application and Player are provided to and executed on a device*. (Ex-1002, ¶¶192-94.) Figure 1A of *Huang* shows that both the **application** and **player** are provided to and are executed on the same device because they are shown operating in the same **Internet browser**:



Huang explains that “**Player 106** [is] *provided to* the **Internet Browser 101** by a server 120 as web pages” and that “**Applications** created by the Designer 104 are ... *loaded into the Player 106* as the application 111.” (Ex-1005, ¶¶[0041]-[0042]; see also Ex-1006, ¶¶[0014], [0097].) Huang also explains that “[e]ach element or component shown in FIG. 1A is a software computer program entity to be executed on a computer”; that “Player 106 execute[s] in a conventional Internet Browser 101”; and that “Player 106 is a computer program that provides a user interface which executes a browser based application.” (Ex-1005, ¶[0040].) Shenfield similarly discloses providing “bundle[s]” of both platform-neutral data (*applications*) and platform-specific components (*players*) for execution on the client device. (Ex-

1007, ¶¶[0039], [0044]-[0045]; Ex-1002, ¶193.) A POSITA would have understood that the application and player of the *Huang-Shenfield* system are provided to the device for execution. (Ex-1002, ¶194.)

Huang and *Shenfield* also disclose that *a user of the device provides one or more input values to an input of a defined UI object*. (Ex-1002, ¶¶195-97.) As discussed above, *Huang* discloses that “Player 106 executes browser-based applications 111, which have application user interface components, such as text fields” like “OppNameTextArea,” “for receiving data values from a user.” (Ex-1005, ¶¶[0054], [0103].) *Nachnani* confirms that once the device executes a player and an application, a user may provide an input value to an input of a defined UI object:

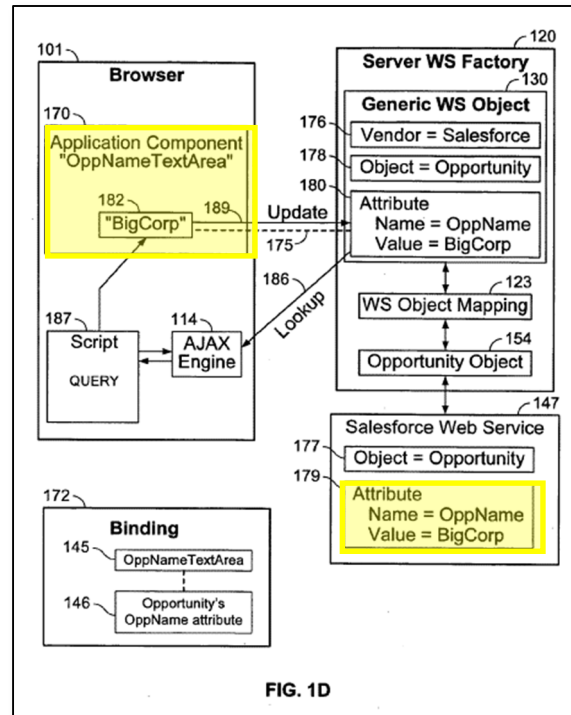
A web browser 132 executes a Player UI 134, which in turn[] executes an application 135. The Player UI 134 presents the application 135 to a user 130 via the web browser 132. The application 135 includes at least one component 136, which represents a user interface element such as a text box or a web link. ... The component 136 also includes a data value 142, e.g. a text string or numeric value, ***which can be specified by a user via the Player UI 134.***

(Ex-1006, ¶[0040]; *see also id.* ¶[0039] (“Application Data 120 may be provided by the user 118 when the Player UI 114 executes the application.”); Ex-1005, ¶¶[0006], [0040].) *Shenfield* similarly discloses that its presentation components allow for user input, such as enabling a user to “type[] in an edit box or push[] a button.” (Ex-1007, ¶[0043].)

- (ii) “1) the device provides the user provided one or more input values and corresponding input symbolic name to the web service,”

A POSITA would have understood that *Huang* discloses that the user of the device provides one or more input values *associated with an input symbolic name to an input of the defined UI object* because, as discussed above, the user provides input to the web service, e.g., for setting an attribute of a web service object, which has a symbolic name (“input symbolic name”) such as “OpportunityName,” as discussed above (*see* Ex-1005, FIG. 4A). (*Supra* Section **Error! Reference source not found.**) A POSITA would further have understood that a data value sent to a web service would be associated with symbolic name of the web service attribute bound to the application component because the web service would need this identifier to locate the attribute to which the input value relates. (Ex-1002, ¶198.)

Huang shows that *the device provides the user provided one or more input values and corresponding input symbolic name to the web service*, for example, in Fig. 1D:



As can be seen, the value of the application component (“BigCorp”) and the associated symbolic name of the attribute (“OppName”) can be sent to the web service, depending on the nature of the bindings. (*See, e.g.*, Ex-1005, ¶[0067].)

- (iii) “2) the web service utilizes the input symbolic name and the user provided one or more input values for generating one or more output values having an associated output symbolic name,”

A POSITA would have understood that *the web service utilizes the input value and the associated identifier (input symbolic name) to generate an output value having an associated output symbolic name.* (Ex-1002, ¶[200-02.]) *Huang* discloses that an output of a web service can depend on an input value. (Ex-1005, ¶[0062]) (“A user of the WS Player 112 can then select a value from the drop-down menu to cause the WS Player 112 to retrieve objects for which the corresponding attribute is

equal to the selected value from the web service.”). In this example, the input value is selected from the drop down menu and is associated with a symbolic name that identifies an attribute and is passed to the web service, which outputs data utilizing the input selection because only “objects for which the corresponding attribute is equal to the selected value” is received from the web service. (*Id.*; Ex-1002, ¶201.) *Huang* further discloses, when processing a request, that the system “extract[s] the input data for the lookup,” which includes the “web service object name.” (Ex-1005, ¶[0129].) A POSITA would similarly have understood that *Huang* discloses that the web service receives and thus *utilizes the input symbolic name and the user provided one or more input values for generating one or more output values* because a web service requires both an input value and an identifier to understand what information it is receiving. (Ex-1002, ¶202.)

- (iv) “(3) said Player receives the output symbolic name and corresponding one or more output values and provides instructions for a display of the device to present an output value in the defined UI object.”

Huang discloses that the values of its application components, which include the outputs from web services (*output value in the defined UI object*), are displayed to the user. (Ex-1005, ¶¶[0057] (“The application component includes a value 192, which is displayed to a user”), [0070] (“query the web service and **display a list of one or more values returned** by the web service according to the binding”).) And

as discussed below, *Huang* specifies that its *player receives the output from a web service and provides the instructions to display the output*.

For example, *Huang* discloses that its player is responsible for interacting with web services and displaying the output of such web services. *Huang* describes its player as interacting with web services, stating that “WS Player 112 component of the Player 106 allows applications executing in the Player 106 to interact with web services by, for example, looking up a value from a web service and storing the value in an application component associated with the application 111, or inserting, updating, or deleting a value in a web service object, where the value is specified by an application component.” (Ex-1005, ¶[0054]; *see also, e.g., id.*, ¶¶ [0057] (“Player 106 use[s] the Generic WS Object 130 to interact with the vendor web service 156, e.g., to set and get values of the vendor-specific attribute 155.”), [0072] (“the WS Player 112 uses binding information that establishes a mapping between WS object attributes and application components to display a Web Service lookup user interface for storing or retrieving the values of those components to or from web services.”).) *Huang* further states that response from a web service can be sent to “the Internet Browser 101 of FIG. 1A, or to ... Player 106 of FIG. 1A, which delivers data to the application.” (Ex-1005, ¶[0129].) The player interacting with web services can be seen in, for example, Figure 1A of *Huang*:

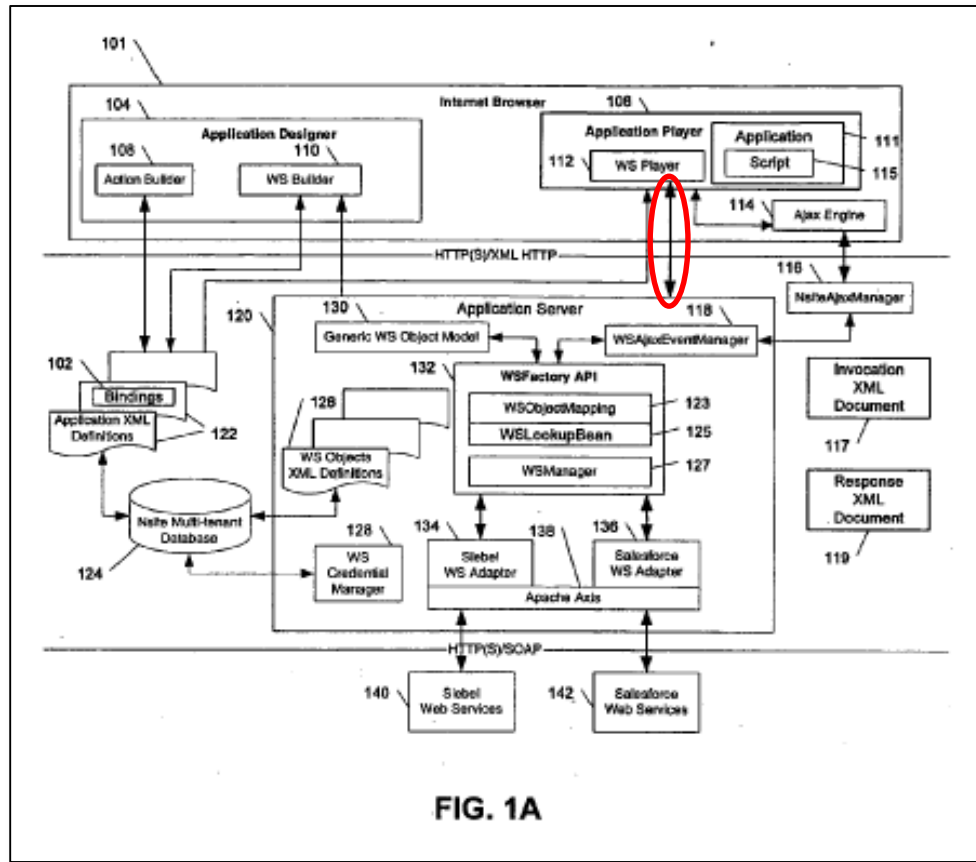


FIG. 1A

(Ex-1005, Fig. 1A.)

From these disclosures, a POSITA would have understood that *Huang's* player receives the output from the web service and provides instructions for displaying the output. (Ex-1002, ¶¶203-07.) *Nachnani*, which *Huang* incorporates by reference (Ex-1005, ¶¶[0006], [0040]), confirms this understanding, stating that the “Player UI ... executes an application 135” and “presents the application 135 to a user 130 via the web browser,” expressly stating that the “Player” displays content. (Ex-1006, ¶[0040].) A POSITA would further have understood that *Huang's* player displays the output by providing instructions because (1) a computer program is, at

a high level, a set of instructions and (2) providing instructions was the only way to cause a device to perform any functionality. (Ex-1002, ¶¶207-09.)

Huang also discloses “text fields” (Ex-1005, ¶[0054]), “text box[es]” (*id.*, ¶[0056]) and “text area component[s]” (*id.*, ¶[0067]), which a POSITA would have recognized are capable of both receiving input text and displaying output text. (Ex-1002, ¶210.) At a minimum, a POSITA would have found it predictable to use the same application component for input and output, to leverage existing resources in an efficient manner, and would have found using the same application component for inputs and outputs a predictable design choice. (*Id.*)

To the extent *Huang* does not explicitly disclose that the player of *Huang*’s system receives the output symbolic name and value(s) and provide instructions for a display of the device to present the output value in the defined application component (*UI object*), a POSITA would have found such a configuration obvious in the *Huang-Shenfield* system. (Ex-1002, ¶211.) *Shenfield* explains that its “workflow components 406,” which are “a series of instructions in the selected programming/scripting language” and can be “based upon different supported capabilities or feature of particular devices 100,” define “[p]resentation workflow and message processing.” (Ex-1007, ¶[0045].) A POSITA would have understood that the platform-specific components of *Shenfield* (*player*), therefore, would receive and process messages received from web services, including any output symbolic

name and values, and also provide the instructions for a display of the device to present the output value in the defined application component. (Ex-1002, ¶211.) As discussed above, a POSITA would have been motivated to modify *Huang*'s system to utilize the device-dependent components of *Shenfield* as a player to allow for improved optimization, efficiency, and end-user experience for a wider range of devices and a POSITA would have had a reasonable expectation of success in view of the similarity between *Huang*'s and *Shenfield*'s components. (*Supra* Section IX.A.7.f.)

2. Claim 3

- a. **“The system of claim 1, where said web component is a text chat, a video chat, an image, a slideshow, a video, or an RSS feed.”**

Shenfield discloses that “data object 800 updates asynchronously pushed from the server (web service 106) are synchronously reflected by the linked UI screen element 802 on the user interface 202” and that “[t]hese mappings 800 can be applicable for ... stock trading, news updates, alerts, weather updates.” (Ex-1007, ¶[0071].) A POSITA would have understood that such updates can be implemented as RSS feeds (updates in XML-formatted plain text). (Ex-1002, ¶213.) *Shenfield* further states that its workflow component can render “objects that represent ... pop-ups, dialog boxes, [and] text areas,” as well as providing for “image loading” and “combines user interface elements and fixed and computed text and images, and is

reactive to user interaction on the user interface 202.” (Ex-1007, ¶[0078].) A POSITA would have understood that “pop-ups, dialog boxes, [and] text areas” are elements suggesting a text-chat interface and that “image loading” and “combin[ing] user interface elements” with “images” disclose an image obtained from a web component. (Ex-1002, ¶214.)

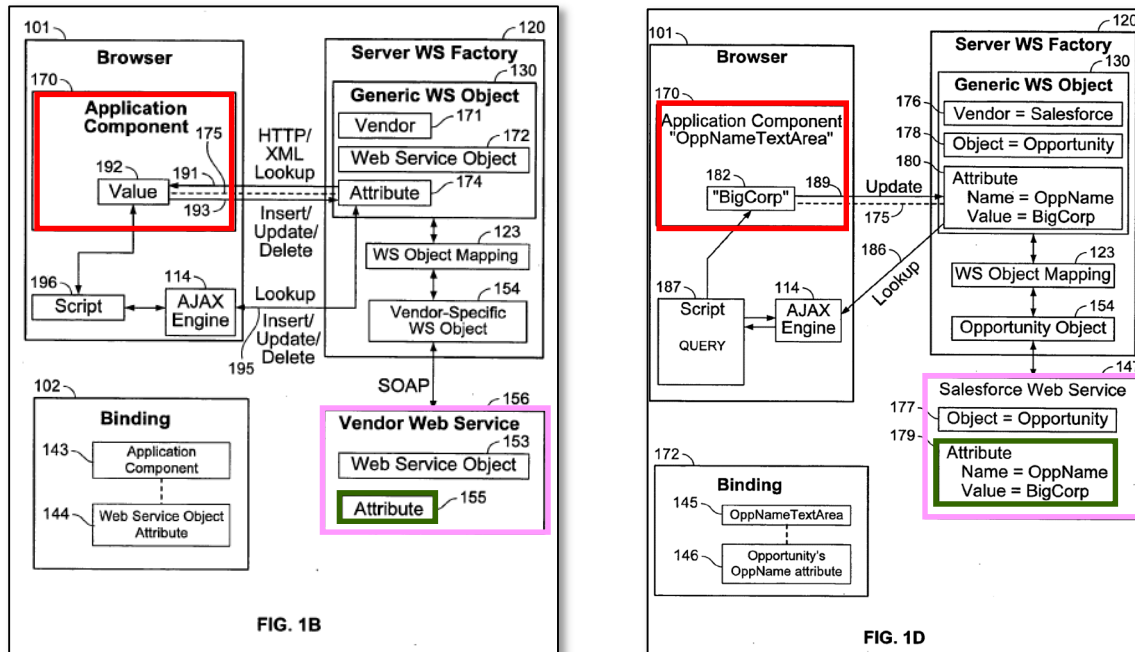
At a minimum, a POSITA would have found binding the application components of the *Huang-Shenfield* system to a text chat, image, or RSS feed web service obvious. Text chat, image, and RSS feed web services were among the finite set of well-known web-service options available to a POSITA. (Ex-1002, ¶215.) Doing so would have been a simple implementation of known technologies via standard means to achieve predictable results. (*Id.*) Indeed, *Huang* and *Nachnani* disclose components and functions that are readily suited to interact with text chat, image, or RSS feed web services, including text input, text output, and image application components that can be bound to web service inputs and outputs. (*E.g.*, Ex-1005, ¶¶[0057], [0068]; Ex-1006, ¶[0042], Fig. 2a; Ex-1002, ¶215; *see also* Ex-1005, ¶¶[0006], [0042].)

3. Claim 5

a. “The system of claim 1, where said UI object is an input field for a web service.”

Huang discloses a text input field for a web service, such as the “OppNameTextArea” **application component**. (Ex-1005, ¶¶[0054], [0103].)

Huang explains that **application components**, such as “OppNameTextArea” may be used to “insert or update” **web service attributes** (*Id.*, ¶¶[0056], [0064], [0067]), and, therefore, are input fields field for a web service. (Ex-1002, ¶¶217-19.)



(Ex-1005, Figs. 1B, 1D.)

4. Claim 6

- a. “The system of claim 1, where said UI object is an input field usable to obtain said web component, where said input field includes a text field, a scrolling text box, a check box, a drop down-menu, a list menu, or a submit button.”

Huang discloses text input fields and submit buttons that invoke web services.

(*Supra* Section IX.A.1.c.) *Huang* further discloses that a UI object can be a “**drop-down menu** that includes a list of possible values for [an] attribute,” which, in this

example, is an input to a web service used to “*retrieve* [only] objects for which the corresponding attribute is equal to the selected value,” thereby disclosing the additional requirement of claim 6. (Ex-1005, ¶[0062]; Ex-1002, ¶¶220-222; *see also* Ex-1006, ¶[0042].)

5. Claim 7

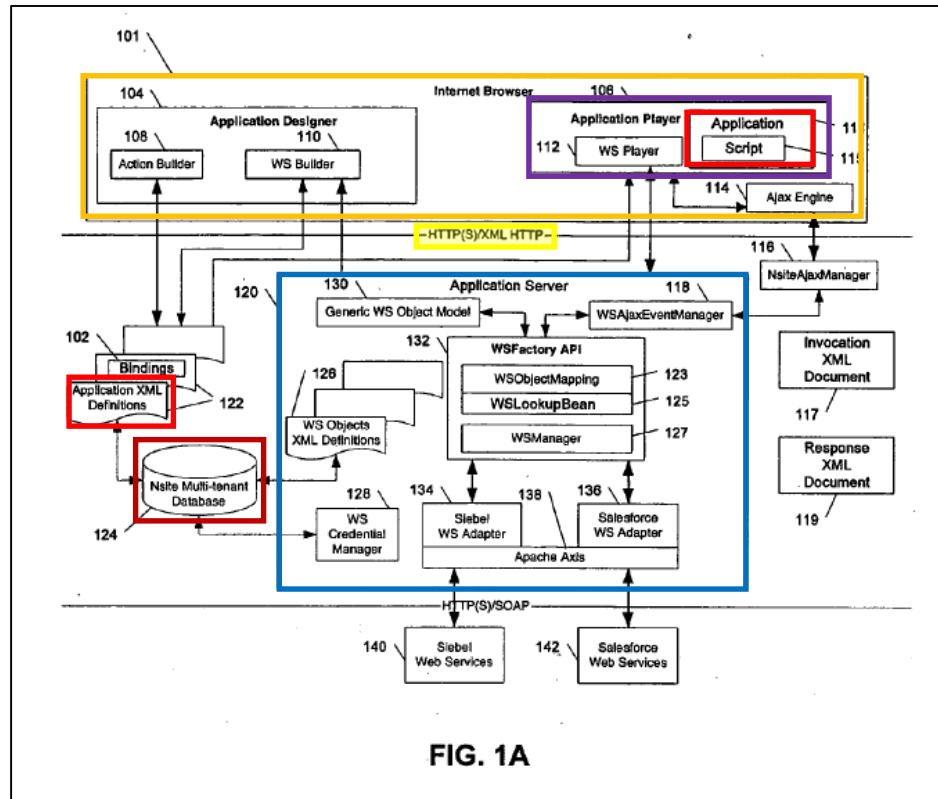
- a. “The system of claim 1, where said web component is an output of a web service, is the text provided by one or more simultaneous chat sessions, is the video of a video chat session, is a video, an image, a slideshow, an RSS display, or an advertisement.”

As discussed for claim 3, *Huang* in view of *Shenfield* renders a text chat, image, or RSS feed web component obvious. (*Supra* Section IX.A.2.) Accordingly, a POSITA would have understood that the *output of such web services* would be *the text provided by one or more simultaneous chat sessions* if a text chat, *an image* for an image web service, and *an RSS display* for a RSS web service. (Ex-1002, ¶223.)

6. Claim 11

- a. “The system of claim 1, where said code is provided over said network.”

Huang discloses that *code* (applications and players) *is provided over the network*. (Ex-1002, ¶¶224-29.) Specifically, “**Player 106** [is] provided to the **Internet Browser 101** by a **server** as web pages” (Ex-1005, ¶[0041]) and that **applications** “are stored ... in a **database 124**, and loaded into the **Player 106** as the **application 111**” (*id.*, ¶[0042]) over “HTTP(S)/XML HTTP.”

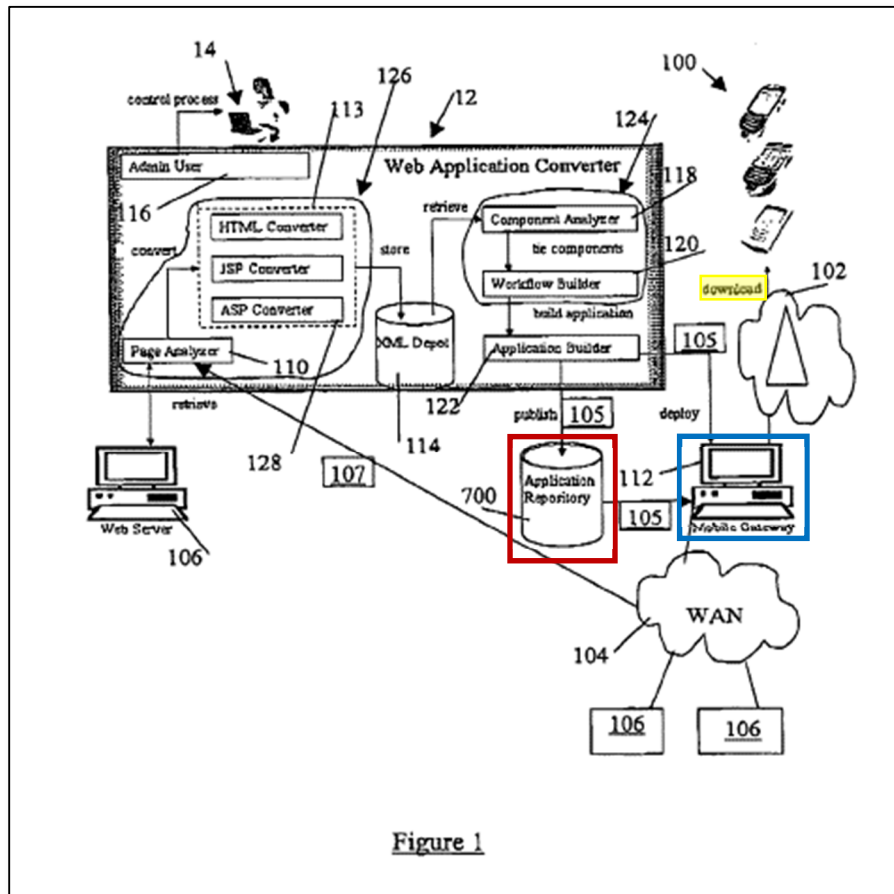


(Ex-1005, Fig. 1A.)

A POSITA would have understood that *Huang* provides its code (applications and players) over a network in view of Figure 1A identifying an “HTTPS(S)/XML HTTP” protocol between the browser 120 and database 124/server 120. (Ex-1002, ¶¶225-26.)

Shenfield similarly discloses providing its code—specifically, its component applications—over a network. *Shenfield* states that “application definitions can be hosted in the repository 700 as a bundle of platform-neutral component definitions linked with different sets of presentation components 403a, 403b, 403c, representing the different supported user interfaces 202 of the devices 100” and that a user “makes

a ... download request to the [server 112](#)” to download these components via a network. (Ex-1007, ¶[0044]; Ex-1002, ¶¶227-28.)



(Ex-1007, Fig. 1.)

A POSITA would have understood that *Shenfield*, like *Huang*, provides its code to devices over “wireless network 102” and the *Huang-Shenfield* system would do the same. (Ex-1007, ¶[0021]; Ex-1002, ¶229.)

7. Claim 12

- a. [12.pre] “A method of displaying content on a display of a device utilizing a registry of one or more web components related to inputs and outputs of a web service obtainable over a network, where each web component includes a plurality of symbolic names of inputs and outputs associated with each web service, 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, said method comprising:”**

Huang in view of *Shenfield* renders these features obvious for many of the same reasons discussed above with respect to claim 1. (*See supra* Sections IX.A.1.a-IX.A.1.b.)

- b. [12.a] “defining a user interface (UI) object for presentation on the display, where said UI object corresponds to a web component included in said registry selected from the group consisting of an input of the web service and an output of the web service;”**

Huang in view of *Shenfield* renders element 12.a obvious for the same reasons discussed for element 1.b. (*Supra* Section IX.A.1.c.)

- c. [12.b] “selecting a symbolic name from said web component corresponding to the defined UI object;”**

Huang discloses element 12.b for the same reasons discussed for element 1.c. (*Supra* Section IX.A.1.d.)

d. [12.c] “associating the selected symbolic name with the defined UI object;”

Huang discloses element 12.c. (*Supra* Section IX.A.1.e.)

e. [12.d] “producing an Application including the selected symbolic name of the defined UI object, where said Application is a device-dependent [*sic*] code; and”

Huang discloses element 12.d when properly construed. The district court construed “where said Application is a device-dependent code” to mean “where said Application is a device-*in*dependent code” because, in part, Express Mobile stipulated to this construction, and “device-independent code” to mean “code that is not specific to the operating system, programming language, or platform of a device.” (*Supra* Section VIII.) As discussed above, “device-dependent code” should be corrected to recite “device-independent code.” (*Id.*) As corrected, *Huang* discloses element 12.d. (*Supra* Section IX.A.1.f.)

f. [12.e] “producing a Player, where said Player is a device-dependent code;”

Huang in view of *Shenfield* renders element 12.e obvious. (*Supra* Section IX.A.1.g.)

- g. [12.f] “such that, when the Application and Player are provided to the device and executed on the device, and when a user of the device provides one or more input values associated with an input symbolic name to an input of defined UI object, 1) the device provides the user provided one or more input values and corresponding input symbolic name to the web service, 2) the web service utilizes the input symbolic name and the user provided one or more input values for generating one or more output values having an associated output symbolic name, 3) said Player receives the output symbolic name and corresponding one or more output values and provides instructions for a display of the device to present an output value in the defined UI object.”

Huang in view of *Shenfield* renders element 12.f obvious. (*Supra* Section IX.A.1.h.)

8. Claim 14

Claim 14 is identical to claim 3, except it depends on claim 12 rather than claim 1, and therefore is obvious for the same reasons as claim 3. (*Supra* Section IX.A.2.)

9. Claim 16

Claim 16 is identical to claim 5, except it depends on claim 12 rather than claim 1, and therefore is obvious for the same reasons as claim 5. (*Supra* Section IX.A.3.)

10. Claim 17

Claim 17 is identical to claim 6, except it depends on claim 12 rather than claim 1, and therefore is obvious for the same reasons as claim 6. (*Supra* Section IX.A.4.)

11. Claim 18

Claim 18 is identical to claim 7, except it depends on claim 12 rather than claim 1, and therefore is obvious for the same reasons as claim 7. (*Supra* Section IX.A.5.)

12. Claim 22

Claim 22 is substantially similar to Claim 11, except that it depends on claim 12 rather than claim 1, and recites “providing said Application and Player over said network,” whereas claim 11 recites “where said code is provided over said network.” (Ex-1001, Claims 11, 22.) As discussed above, the *Huang-Shenfield* system provides code over the network in the form of an Application and Player. (*Supra* Section IX.A.6; Ex-1002, ¶249.) Accordingly, *Huang* in view of *Shenfield* renders claim 22 obvious for the same reasons as claim 11. (*Supra* Section IX.A.6; Ex-1002, ¶249.)

X. THE DISCRETIONARY FACTORS FAVOR INSTITUTING TRIAL

A. This Petition Presents New, Non-Cumulative Prior Art That Has Not Previously Been Considered

Neither the same prior art nor substantially the same arguments were previously presented to the Patent Office during the prosecution of the '755 patent. (*Supra* Section VI.B.) Neither *Huang* nor *Shenfield* were cited during the prosecution of the '755 patent. The disclosures of the prior art of record differ materially from *Huang* and *Shenfield*. For example, both *Huang* and *Shenfield* disclose or suggest producing and providing separate software modules to generate a display on a device, with *Shenfield* disclosing a system that provides both device-specific and device-independent code. (*Supra* Section IX.A.1.) Accordingly, the applicant's argument to obtain allowance—that any combination of references “taken individually or in combination, does not teach or suggest providing both device-independent code and device-dependent code to a device that, when executed on the device, exchanges web service input and output names and values, and displays output on such device” (Ex-1004, 1223)—does not apply to *Huang* and *Shenfield*. Accordingly, the Board should not discretionarily deny institution under § 325(d).

B. The *Fintiv* Factors Favor Institution

The Board considers the *Fintiv* factors to determine whether to exercise its discretion under 35 U.S.C. §314(a). The weight of the factors articulated in *Apple*,

Inc. v. Fintiv, Inc., IPR2020-00019, Paper 11, at 6 (Mar. 20, 2020) (designated precedential) favors institution.

The first *Fintiv* factor (potential for a district court stay) favors institution. Upon institution, Petitioner will seek a stay of the district court litigation, i.e., the ExMo-Adobe Litigation, venued in the Northern District of California and assigned to Judge Seeborg. Judge Seeborg will likely grant a stay because he often grants stays where, like here, the case is in its early stages and the IPR petitions addresses all asserted claims. Dkt. 105, *OrthoAccel Techs., Inc. v. Propel Orthodontics, LLC*, No. 17-cv-03801-RS (N.D. Cal. Aug. 10, 2018) (Ex-1031.); Dkt. 205, *Core Wireless Licensing S.A.R.L. v. Apple Inc.*, No. 15-cv-05007-RS (N.D. Cal. June 15, 2016) (Ex-1032); *see also, e.g., CF Traverse LLC v. Amprius*, No. 20-cv-00484-RS, 2020 WL 6820942, at *1-2 (N.D. Cal. Nov. 6, 2020) (granting stay pre-institution); *Karl Storz Endoscopy-Am., Inc. v. Stryker Corp.*, No. 14-cv-00876-RS, 2015 WL 13727876, at *2-5 (N.D. Cal. Mar. 30, 2015) (same). Indeed, Judge Seeborg already granted a stay pending reexamination of another litigation, where Express Mobile asserted two other patents against Adobe's subsidiary. Dkt. 254 at 2, *X Commerce, Inc. v. Express Mobile, Inc.*, No. 17-cv-02605-RS (N.D. Cal. Mar. 5, 2021) (Ex-1033). Further, Judge Seeborg is likely to stay the ten other pending cases asserting the '755 patent and also assigned to him as well, which would further serve the interests of judicial economy.

The second *Fintiv* factor (trial date relative to final written decision due date) favors institution. No trial date is set for the ExMo-Adobe Litigation; the parties' proposed case schedules only provides for local patent rule contentions and claim construction proceedings, culminating with the claim construction hearing on December 8, 2021. Dkt. 50-1, ExMo-Adobe Litigation (Ex-1036); Dkt. 51, ExMo-Adobe Litigation (Ex-1037). In fact, the proposed case schedule provides no dates for the close of fact discovery, let alone expert discovery and dispositive motions. The same is true for the ten other cases Express Mobile brought in the same district that are assigned to the same judge and involve the '755 patent—none have trial dates and all have the December 8, 2021 claim construction hearing as the last deadline on the current schedule.

The third *Fintiv* factor (investment in the parallel proceedings) favors institution. The ExMo-Adobe Litigation is still in its infancy. Petitioner has not served its Answer, given its pending motion to dismiss under *Twombly*. No fact discovery has occurred. Indeed, fact discovery may not begin in earnest until after the claim construction hearing on December 8, 2021. Adobe files this Petition more than five months before the one-year deadline to file an IPR, and about six weeks after Patent Owner served its infringement contentions in the ExMo-Adobe Litigation. Accordingly, Adobe has exercised diligence. *Taiwan Semiconductor Mfg. Co., Ltd. v. Fraunhofer Gesellschaft Zur Förderung Der Angewandten*

Forschung EV, IPR2020-01669, Paper No. 13 at 28 (P.T.A.B. Apr. 15, 2021)

(granting petition when filed “less than two months after receiving Patent Owner’s infringement contentions, [and] more than four months before the statutory deadline”); *DISH Network L.L.C. v. Broadband iTV, Inc.*, IPR2020-01359, Paper No. 15 at 19-20 (P.T.A.B. Feb. 12, 2021) (finding diligence when the petitioner “filed its Petition ... approximately three months after first receiving notice of Patent Owner’s infringement positions for all asserted claims”).

The fourth *Fintiv* factor (overlap of issues) favors institution. Petitioner hereby stipulates that if IPR is instituted, it will not assert the same invalidity grounds in this petition, as well as any invalidity grounds with the same primary references as this petition, in the ExMo-Adobe Litigation.

The fifth *Fintiv* factor (the parties) is neutral. The ExMo-Adobe Litigation and the IPR here involve the same parties. *But see Cisco Sys., Inc. v. Ramot at Tel Aviv Univ. Ltd.*, IPR 2020-00122, Paper 15, at 10 (May 15, 2020) (APJ Crumbley, dissenting) (noting that weighing this factor against a petitioner when the parties in IPR are the same in related litigation, could “tip the scales against a petitioner merely for being a defendant in the district court.” This “would seem to be contrary to the goal of providing district court litigants an alternative venue to resolve questions of patentability.”).

The sixth *Fintiv* factor (other factors, including the merits) also favors institution. As the *Fintiv* panel noted, “if the merits of a ground raised in the petition seem particularly strong on the preliminary record ... the institution of a trial may serve the interest of overall system efficiency and integrity...” *Fintiv*, IPR2020-00019, Paper 11, at 14-15 (addressing factor 6). As detailed above, the challenged claims are clearly obvious. Therefore, the overall system efficiency and integrity would best be served by institution. Moreover, Patent Owner’s litigious behavior—*asserting the ’755 patent in dozens of district court cases*—strengthens the public interest in ensuring review of the ’755 patent and institution promotes “overall system efficiency and integrity,” which is the core aim undergirding the *Fintiv* discretionary denial analysis. *Fintiv* at 15; *see also id.* at 16.

Adobe therefore respectfully submits that the *Fintiv* factors favor institution and that discretionary denial of this Petition would neither be appropriate nor equitable.

C. The *General Plastic* Factors Favor Institution

The Board also considers the *General Plastic* factors to determine whether to exercise its discretion under §314(a) when multiple petitions are filed challenging claims of a patent. *General Plastic Industrial Co., Ltd. v. Canon Kabushiki Kaisha*, IPR2016-01357, Paper 19, at 15-16 (designated precedential). Although Google LLC filed a petition on March 31, 2021 (IPR2021-00709) (“Google IPR”), and SAP

America, Inc. filed a petition on July 2, 2021 (IPR2021-01144) (“SAP IPR”), this is the result of Express Mobile serially asserting the ’755 patent. The Board should “decline to wield [Express Mobile’s] own litigation activities as a shield in this *inter partes* review.” *Samsung Elecs. Co., Ltd. v. Iron Oak Techs., LLC*, IPR2018-01554, Paper 9 at 31 (PTAB Feb. 13, 2019); *see also Samsung Elecs. Am., Inc. v. Uniloc Luxembourg S.A.*, IPR2017-01797, Paper 8 at 33 (PTAB Feb. 6, 2018) (“Patent Owner’s complaining about the multiple *inter partes* review petitions ... is not persuasive when the volume appears to be a direct result of its own litigation activity.”). Express Mobile has asserted the ’755 patent in over 20 cases since 2019 across multiple jurisdictions and cannot now complain that multiple parties have sought to challenge the claims of the ’755 patent via IPR.

General Plastic factor 1 (whether same petitioner; same challenged claims) favors institution. First, Adobe is not the petitioner in the Google IPR or the SAP IPR and little, if any, relationship exists between these parties. *Volkswagen Grp. of Am., Inc. v. Carucel Invs., L.P.*, IPR2019-01103, Paper 7 at 16-17 (PTAB Nov. 12, 2019). Further, Adobe’s Petition is largely directed to different claims (1, 3, 5-6, 11-12, 14, 16-17, and 22) than the Google IPR (claims 1, 2, 7, and 11) and was filed only one business day after the SAP IPR such that concerns regarding “abusive serial petitions” are not implicated, *id.* at 15, thereby favoring institution. Second, Adobe, Google, and SAP have no “significant relationship”—they are not co-defendants in

the same litigation and are accused of infringing the '755 patent based on different and independently-developed products. *See Apple, Inc. v. UUSI, LLC d/b/a Naraton*, IPR2019-00358, Paper 12 at 14-15 (PTAB Aug. 5, 2019) (finding no “significant relationship” between Apple and Samsung). With no significant relationship between Adobe, Google, and SAP, Petitioner Adobe should be given its own opportunity to challenge these claims.

Factors 2 and 4 (timing of knowledge of prior art and time between knowing art and filing petition, respectively) are neutral. Although Adobe learned of *Huang* and *Shenfield* shortly before Google filed its request for IPR on March 31, 2021, this carries little weight because “two different set of claims are challenged in the two proceedings.” *See Signify Holding B.V. v. Lighting Sci. Grp. Corp.*, IPR2020-00753, Paper 16 at 13-14 (PTAB Sept. 16, 2020). Further, Adobe filed only one business day after the SAP IPR, favoring institution. *See Nevro Corp. v. Boston Sci. Neuromodulation Corp.*, IPR2017-01920, Paper 12 at 25 (PTAB Feb. 5, 2018) (“only 21 days elapsed between filing” favored institution under Factor 4); *see also Volkswagen*, IPR2019-01103, Paper 7 at 14-17 (declining to deny institution under *General Plastic* when filed “within three days of [the prior] petition” because “[t]he ‘abuse’ ... caused by petitioners ‘modify[ing] and refin[ing] their evidence and strategies after learning the initial arguments of the patent owner,’ is not present here because the two petitions were filed at or around the same time.”). More

importantly, Adobe challenges only the claims referenced in Express Mobile's infringement contentions and could not identify which claims are relevant to the parties' dispute and to challenge in this Petition until Express Mobile identified the asserted claims in the ExMo-Adobe Litigation, which it did for the first time only six weeks ago on May 27, 2021, when it served its infringement contentions. Indeed, Adobe's invalidity contentions in the ExMo-Adobe Litigation are still not due until July 19, 2021. Further, Adobe "was not a petition in the prior proceedings challenging the ['755] patent," and, therefore, "[Factor 2] has little probative value here." *United Fire Protection Corp. v. Eng'ered Corrosion Sols., LLC*, IPR2018-00991, Paper 10 at 13 (PTAB Nov. 15, 2018).

Factor 3 (receipt of preliminary response or institution decision) favors institution. Patent Owner has not filed a preliminary response in the Google IPR or SAP IPR. Nor has the Board decided whether to institute in both proceedings. Thus, no concerns of gamesmanship or road-mapping exist here and favors institution. *See Volkswagen*, IPR2019-01103, Paper 7 at 17 ("Petitioner could not have had the benefit of Patent Owner's preliminary response or the Board's decision before filing because the petitions were filed within days of one another."); *Netflix, Inc. v. Realtime Adaptive Streaming LLC*, IPR2018-01630 Paper 13 at 11 (PTAB Apr. 19, 2019); *General Plastic*, IPR2016-01357, Paper 19 at 16-17.

The fifth factor (time between petitions on same claims) favors institution.

First, as discussed above, Adobe files its petition only one business day after the SAP IPR. *Volkswagen*, IPR2019-01103, Paper 7 at 17 (“[T]he fifth factor ... is not applicable here as the time lapse (three days) was minimal.”). Second, Adobe’s Petition is directed at largely different claims than Google’s petition. Therefore, this factor supports institution. *E.g.*, *Ericsson Inc. v. Intellectual Ventures I LLC*, IPR2018-01318, Paper 6 at 50 (PTAB Jan. 15, 2019) (this factor “does not weigh in favor of ... deny[ing] institution” when the petitions are “not ... directed to the same claims of the same patent”). Second, Express Mobile served infringement contentions against Adobe on May 27, 2021, and Adobe diligently prepared and filed this Petition within six weeks. *Volkswagen Grp. of Am., Inc. v. Carucel Investments, L.P.*, IPR2019-01573, Paper 7 at 7-8 (PTAB Jan. 22, 2020) (filing Petition within weeks of being served infringement contentions favors institution).

The sixth (finite resources of Board) and seventh (requirement to issue determination within one year) factors favor institution. First, this Petition relies on only one ground to assert obviousness—*Huang* in view of *Shenfield*—and on the same base reference as the earlier-filed Google IPR. Accordingly, “the grounds asserted in [Adobe’s] Petition[] ha[s] similarities [with Google’s IPR] that will allow the Board to evaluate the two petitions efficiently.” *See Netflix*, IPR2018-01630, Paper 13 at 14. Second, Adobe filed this petition promptly before the preliminary

response in the Google IPR, thereby preserving the Board's resources to manage this proceeding efficiently with the Google IPR. Third, this Petition relies on the *Huang* and *Shenfield* references, both of which represent undisputed prior art to the '755 patent, providing further efficiency for the Board's assessment. Fourth, Adobe filed this Petition only one business day after the SAP IPR, thus allowing for the Board to resolve these petitions on the same timeline.

XI. MANDATORY NOTICES UNDER 37 C.F.R. § 42.8

A. Real Parties-in-Interest

Adobe identifies Adobe Inc. and X.Commerce Inc. as real parties-in-interest.

B. Related Matters

Adobe is concurrently filing IPR petitions challenging certain claims of U.S. Patent Nos. 9,471,287 and 9,928,044, which claim priority to the '755 patent. To Adobe's best knowledge, the '755 patent is the subject of the following co-pending IPR petitions:

1. *Google LLC v. Express Mobile, Inc.*, IPR2021-00709 (PTAB)
2. *SAP America, Inc. v. Express Mobile, Inc.*, IPR2021-01144 (PTAB)

To Adobe's best knowledge, the '755 patent is the subject of the following co-pending civil actions:

1. *Express Mobile, Inc. v. Adobe Inc. d/b/a Adobe Systems Inc.*, No. 3-20-cv-08297 (N.D. Cal.)

2. *Express Mobile, Inc. v. Google LLC*, No. 6-20-cv-00804 (W.D. Tex.)
3. *Express Mobile, Inc. v. Dropbox, Inc.*, No. 3-21-cv-01145 (N.D. Cal.)
4. *Express Mobile, Inc. v. Salesforce.com, Inc.*, No. 3-20-cv-08461 (N.D. Cal.)
5. *Express Mobile, Inc. v. Booking.com B.V.*, No. 3-20-cv-08491 (N.D. Cal.)
6. *Express Mobile, Inc. v. SAP SE*, No. 3-20-cv-08492 (N.D. Cal.)
7. *Express Mobile, Inc. v. Oath Holdings Inc. f/k/a Yahoo!*, No. 3-20-cv-08321 (N.D. Cal.)
8. *Express Mobile, Inc. v. Pinterest, Inc.*, No. 3-20-cv-08335 (N.D. Cal.)
9. *Express Mobile, Inc. v. Amazon.com, Inc.*, No. 3-20-cv-08339 (N.D. Cal.)
10. *Express Mobile, Inc. v. HubSpot, Inc.*, No. 1-20-cv-01162 (D. Del.)
11. *Express Mobile, Inc. v. Squarespace, Inc.*, No. 1-20-cv-01163 (D. Del.)
12. *Express Mobile, Inc. v. Microsoft Corp.*, No. 3-20-cv-06152 (N.D. Cal.)
13. *Express Mobile, Inc. v. Expedia, Inc.*, No. 6-20-cv-00801 (W.D. Tex.)
14. *Express Mobile, Inc. v. Facebook Inc.*, No. 6-20-cv-00803 (W.D. Tex.)

15. *Express Mobile, Inc. v. Atlassian Corporation PLC*, No. 6-20-cv-00805 (W.D. Tex.)
16. *Express Mobile, Inc. v. Web.com Group, Inc.*, No. 3-20-cv-00839 (M.D. Fl.)
17. *Express Mobile, Inc. v. GoDaddy.com, LLC*, No. 1-19-cv-01937 (D. Del.)
18. *Express Mobile, Inc. v. Wix.com, Ltd.*, No. 3-19-cv-06559 (N.D. Cal.)
19. *Shopify Inc. v. Express Mobile, Inc.*, No. 1-19-cv-00439 (D. Del.)
20. *Express Mobile, Inc. v. Slack Technologies, Inc.*, No. 3-21-cv-02001 (N.D. Cal.)

To Adobe's best knowledge, the '755 patent was formerly the subject of the following civil actions:

1. *Express Mobile, Inc. v. Dropbox, Inc.*, No. 6-20-cv-00806 (W.D. Tex.) (terminated Feb. 4, 2021)
2. *Express Mobile, Inc. v. Web.Com Group, Inc.*, No. 1-19-cv-01936 (D. Del.) (terminated July 28, 2020)
3. *Express Mobile, Inc. v. Contus Interactive, Inc.*, No. 3-19-cv-03350 (N.D. Cal.) (terminated Feb. 7, 2020)
4. *Express Mobile, Inc. v. Rishabh Business Solutions, Inc.*, No. 3-19-cv-03356 (N.D. Cal.) (terminated Mar. 13, 2020)

5. *Express Mobile, Inc. v. Possible Worldwide, LLC*, No. 2-19-cv-05110
(C.D. Cal.) (terminated Oct. 18, 2019)

C. Lead and Backup Counsel

Lead Counsel	Back-Up Counsel
Jennifer A. Sklenar (Reg. No. 40,205) Jennifer.Sklenar@arnoldporter.com Arnold & Porter Kaye Scholer LLP 601 Massachusetts Ave, NW Washington, DC 20001-3743 Tel: 202-942-5000 Fax: 202-942-5999 * Admitted in NY and CA only; practice limited to matters before federal courts and agencies	Jeffrey A. Miller (Reg. No. 35,287) jmillerrptab@apks.com Arnold & Porter Kaye Scholer LLP Five Palo Alto Square, Suite 500 3000 El Camino Real Palo Alto, CA 94306 Tel: 650-319-4500 Fax: 650-319-4700

D. Service Information

Please address all correspondence to lead and back-up counsel at the addresses shown above. Adobe consents to electronic service by e-mail.

Petition for *Inter Partes* Review

U.S. Patent No. 9,063,755

Date: July 6, 2021

Respectfully submitted,

/s/ Jennifer A. Sklenar

Jennifer A. Sklenar, Lead Counsel

Reg. No. 40,205

CERTIFICATE OF COMPLIANCE

The undersigned hereby certifies that the foregoing Petition for *Inter Partes* Review contains 13,945 words, excluding those portions identified in 37 C.F.R. § 42.24(a), as measured by the word-processing system used to prepare this paper.

/s/ Jennifer A. Sklenar

Jennifer A. Sklenar (Reg. No. 40,205)
Counsel for Petitioner Adobe Inc.

CERTIFICATE OF SERVICE

I certify that on July 6, 2021, I caused a true and correct copy of the foregoing Petition for *Inter Partes* Review of U.S. Patent No. 9,063,755 and supporting exhibits to be served via overnight delivery on the Patent Owner at the following correspondence address of record as listed on PAIR:

Steven R. Vosen
909 Marina Village Parkway #138
Alameda, CA 94501-1048

A courtesy copy was also sent via electronic mail to Patent Owner's litigation counsel listed below:

Jamie L. Lucia, jlucia@steptoe.com
James R. Nuttall, jnuttall@steptoe.com
Michael Dockterman, mdockterman@steptoe.com
Tron Fu, tfu@steptoe.com
Robert F. Kappers, rkappers@steptoe.com
Katherine H. Johnson, kjohnson@steptoe.com
Christopher A. Suarez, csuarez@steptoe.com
Timothy Devlin, tdevlin@devlinlawfirm.com

/s/ Jennifer A. Sklenar
Jennifer A. Sklenar (Reg. No. 40,205)
Counsel for Petitioner