

Case IPR2025-00564
U.S. Patent No. 7,702,719

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

CARVANA, LLC,
Petitioner

v.

INTERNATIONAL BUSINESS MACHINES CORP.
Patent Owner

Case IPR2025-00564

U.S. Patent No. 7,702,719

Title: Methods And Apparatus For Reducing The Number Of Server Interactions
In Network-Based Applications Using A Dual-MVC Approach

INTERNATIONAL BUSINESS MACHINES CORPORATION'S
PATENT OWNER PRELIMINARY RESPONSE

Mail Stop Patent Board
Patent Trial and Appeal Board
U.S. Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450

TABLE OF CONTENTS

	<u>Page</u>
I. INTRODUCTION	1
II. OVERVIEW OF THE '719 PATENT	3
A. Conventional MVC Architectures Had Shortcomings	3
B. IBM's Inventors Developed A Solution That Addressed Limitations In The Existing MVC Architecture	6
C. Prosecution History Of The '719 Patent.....	9
III. LEVEL OF ORDINARY SKILL IN THE ART	11
IV. OVERVIEW OF THE ROY REFERENCE	11
A. Roy Discloses A Computer Network System With Peer-To- Peer Networking Capabilities.....	11
V. CLAIM CONSTRUCTION	18
VI. THE '719 PATENT CLAIMS ARE PATENTABLE OVER PETITIONER'S GROUNDS 1–4	19
A. GROUND 1: Petitioner Does Not Establish A Reasonable Likelihood Of Prevailing Against Any Claim Under Ground 1 Because Roy Fails To Disclose, Expressly or Inherently, All Limitations of Challenged Claims.....	19
1. Roy Does Not Disclose “Configuring The Server To ... Execute View-Generating ... Logic Associated With The Application,” As Required By Limitations 1(a), 13(a), 25(a), 26(a)	20
2. Roy Does Not Disclose “Wherein The Client And The Server Both Locally Maintain At Least A Portion Of The Model And Execute The View-Generating And Controller Logic Associated Therewith” As Required By Limitation (d).....	27

B.	GROUND 2: Petitioner Does Not Establish A Reasonable Likelihood Of Prevailing Against Any Claim Under Ground 2 Because Roy Fails To Disclose Or Render Obvious All Limitations of Challenged Claims.....	42
1.	Petitioner Fails To Demonstrate That Roy Discloses That Player B’s Computer System 62 Is A “Server,” As Claimed In Limitation (a)	43
2.	Roy Does Not Render Obvious “Wherein The Client And The Server Both Locally Maintain At Least A Portion Of The Model And Execute The View-Generating And Controller Logic Associated Therewith” As Required By Limitation (d).....	43
C.	FOUND 3–4: Petitioner Does Not Establish A Reasonable Likelihood Of Prevailing Against Any Claim Under Ground 3 Or Ground 4.....	47
VII.	CONCLUSION	47

PATENT OWNER'S EXHIBIT LIST

EX #	DESCRIPTION
2001	<i>IBM v. Zynga</i> , 1:22-cv-00590, DI 0001 IBM Complaint (May 2, 2022) (excerpted)
2002	<i>IBM v. Zynga</i> , 1:22-cv-00590, DI 0027 IBM 1st Amended Complaint (July 21, 2022) (excerpted)
2003	<i>IBM v. Zynga</i> , 1:22-cv-00590, IBM Final Infringement Contentions (November 13, 2023) (excerpted)
2004	<i>IBM v. Zynga</i> , 1:22-cv-00590, DI 0081 Def Answer to 1st Amended Complaint (December 7, 2022) (excerpted)
2005	<i>IBM v. Zynga</i> , 1:22-cv-00590, Def Final Invalidity Contentions (December 12, 2023) (excerpted with redactions) (bearing Bates number range beginning with IBM-CARVANA00022845)
2006	<i>IBM v. Zynga</i> , 1:22-cv-00590, Def Exhibit C-13 Roy Invalidity Chart (December 12, 2023) (with redactions) (bearing Bates number range beginning with IBM-CARVANA00411693)
2007	<i>IBM v. Zynga</i> , 1:22-cv-00590, Expert Report of Dr. John K. Bennett (February 2, 2024) (excerpted with redactions) (including pages bearing Bates number range beginning with IBM-CARVANA00238717) ¹
2008	<i>IBM v. Zynga</i> , 1:22-cv-00590, Trial Transcript (September 12, 2024) (excerpted with redactions)
2009	<i>IBM v. Zynga</i> , 1:22-cv-00590, DI 0529 Jury Verdict (September 13, 2024) (with redactions)
2010	<i>IBM v. Zynga</i> , 1:22-cv-00590, DI 0539 Partial Judgment following Jury Verdict (September 24, 2024)
2011	<i>Carvana v. IBM</i> , 7:23-cv-08616, Carvana DJ Complaint (September 29, 2023)
2012	<i>Carvana v. IBM</i> , 7:23-cv-08616, DI 0012 Carvana Corrected DJ Complaint (October 2, 2023)

¹ Ex. 2007 includes excerpts of the Bates-stamped version of the Expert Report of Dr. John K. Bennett as well as the corresponding pages from a version of the same report without Bates stamps.

EX #	DESCRIPTION
2013	<i>Carvana v. IBM</i> , 7:23-cv-08616, DI 0039 Case Management Order (December 12, 2023)
2014	<i>Carvana v. IBM</i> , 7:23-cv-08616, DI 0063 IBM Amended Answer and Counterclaims (April 3, 2024) (excerpted)
2015	<i>Carvana v. IBM</i> , 7:23-cv-08616, DI 0067 Scheduling Order Amending Case Deadlines (April 15, 2024)
2016	Email from IBM to Carvana re: IBM Production of Documents Vol IBM-CARVANA010 (March 1, 2024)
2017	Email from IBM to Carvana re: IBM Production of Documents Vol IBM-CARVANA009-011 (REPROD) (March 22, 2024) (excerpted)
2018	Email from IBM to Carvana re: IBM Production of Documents Vol IBM-CARVANA016 (April 19, 2024)
2019	<i>Carvana v. IBM</i> , 7:23-cv-08616, IBM Supplemental Infringement Contentions (May 24, 2024) (excerpted with redactions)
2020	<i>Carvana v. IBM</i> , 7:23-cv-08616, Carvana Supplemental Invalidity Contentions (July 10, 2024) (excerpted)
2021	<i>Carvana v. IBM</i> , 7:23-cv-08616, Carvana Exhibit D-13 Roy Invalidity Chart (December 12, 2023)
2022	Annotated Carvana Exhibit D-13 Roy Invalidity Chart
2023	Deposition Hours Table (with redactions)
2024	<i>Carvana v. IBM</i> , 7:23-cv-08616, Claim Construction Briefing (Combined) (excerpted)
2025	<i>Carvana v. IBM</i> , 7:23-cv-08616, Markman Hearing Transcript (November 20, 2024) (excerpted)
2026	<i>Carvana v. IBM</i> , 7:23-cv-08616, DI 0146 Order Granting Motion to Stay Briefing Schedule (April 22, 2025)
2027	<i>Carvana v. IBM</i> , 7:23-cv-08616, DI 0143 IBM Letter Opposing Motion to Stay Pending IPR (February 14, 2025)
2028	Docket Navigator Time to Trial Statistics - SDNY
2029	Federal Caseload Management Statistics - SDNY (December 31, 2024)
2030	Curriculum Vitae of Dr. Henry Houh (“Houh CV”)
2031	Declaration of Dr. Henry Houh (“Houh Decl.”)
2032	Krasner, G. & Pope, S., <i>A Cookbook for Using the Model-View-Controller User Interface Paradigm in Smalltalk-80</i> , Journal of Object-Oriented Programming, August/September 1988 (“Krasner-Pope”)

TABLE OF AUTHORITIES

Page(s)

CASES

<i>Apple Inc. v. MPH Tech. Oy</i> , 28 F.4th 254 (Fed. Cir. 2022).....	24, 35, 42, 46
<i>Arendi S.A.R.L. v. Apple Inc.</i> , 832 F.3d 1355 (Fed. Cir. 2016)	46
<i>Int’l Bus. Mach. Corp. v. Zynga Inc., et al.</i> , Case No. 1:22-cv-00590-GBW (D. Del.)	<i>passim</i>
<i>Intelligent Bio-Sys., Inc. v. Illumina Cambridge Ltd.</i> , 821 F.3d 1359 (Fed. Cir. 2016)	45
<i>iRobot Corp. v. SharkNinja Operating LLC</i> , Case No. 2023-1398, 2024 WL 3687981 (Fed. Cir. Aug. 7, 2024).....	46
<i>Kinetic Concepts, Inc. v. Smith & Nephew, Inc.</i> , 688 F.3d 1342 (Fed. Cir. 2012)	46
<i>In re Magnum Oil Tools Int’l, Ltd.</i> , 829 F.3d 1364 (Fed. Cir. 2016)	46
<i>Therasense, Inc. v. Becton, Dickinson & Co.</i> , 593 F.3d 1325 (Fed. Cir. 2010)	31, 39
<i>TQ Delta, LLC v. Cisco Sys., Inc.</i> , 942 F.3d 1352 (Fed. Cir. 2019)	24
<i>Transclean Corp. v. Bridgewood Servs., Inc.</i> , 290 F.3d 1364 (Fed. Cir. 2002)	31
<i>Xerox Corp. v. Bytemark, Inc.</i> , IPR2022-00624, Paper 9 (PTAB Aug. 24, 2022)	24, 35, 42, 46

I. INTRODUCTION

Patent Owner IBM respectfully requests that the Board deny institution of Carvana’s IPR Petition challenging U.S. Patent No. 7,702,719 (“the ’719 Patent”). As explained in Patent Owner’s Brief in support of Discretionary Denial (Paper 8), Carvana’s Petition recycles the same prior art (the Roy reference), the same expert testimony (by Dr. John Bennett), and the same analysis that Zynga fully litigated—and ultimately lost—before a jury nine months ago in *Int’l Bus. Mach. Corp. v. Zynga Inc., et al.*, Case No. 1:22-cv-00590-GBW (D. Del.) (“Zynga Litigation”). Paper 8 at 1–5, 11, 26–33, 36–38. Repackaging meritless arguments and re-presenting them before a different forum does not justify a different outcome.

The ’719 Patent is directed to a **dual**-model-view-controller (“dual-MVC”) architecture. This dual-MVC architecture overcomes the shortcomings of the classic model-view-controller architecture employed by traditional web-based user interfaces, such as the drawback of excessive loading times due to network latency issues. Ex. 1001 at 1:17–2:35. In particular, the ’719 Patent’s dual-MVC architecture allows for storing a subset of the Internet application’s MVC components on the user’s client device such that the client can refresh its screen without interacting with the server, thereby reducing the number of client-server interactions. Ex. 1001 at 2:39–49. By employing a dual-MVC architecture, “the client has the ability to respond to user interaction without paying the performance

penalty of continually interacting with the server.” Ex. 1001 at Abstract. This invention has significantly improved the design of Internet-based user interfaces, reducing excessive latency due to numerous client-server interactions.

Attempting to relitigate arguments that a federal judge and jury have already rejected, *see* Paper 8, 1–5, 11, 26–33, 36–38, Petitioner asserts that the Roy reference anticipates or renders obvious the independent claims of the ’719 Patent. But as the federal judge and jury already found, the Roy reference does not invalidate the ’719 Patent for numerous reasons, including at least the two reasons discussed in this POPR below.

First, Petitioner fails to demonstrate that Roy discloses or renders obvious a server executing view-generating logic, as required by Claim 1(a). Ex. 1001 at 9:41–43. Attempting to sidestep that non-disclosure, Petitioner instead attempts to classify Roy’s “client” device as a “server.” Pet., 16. But Petitioner’s attempt to relabel Roy’s “client” device as a “server” contradicts Roy’s explicit teachings that Petitioner relies on, which merely describe two *client* devices with peer-to-peer networking capabilities—not a dual-MVC implemented in a *client-to-server* system. Indeed, Petitioner improperly attempts to read the requirement of “server” out of the claims by suggesting that a “client” can be a “server.” *See infra* Sections IV.A, VI.A.1. Petitioner’s legally and factually flawed position—which deprives “server” and “client” of any distinction—is fatal for all Grounds in the Petition.

Second, Petitioner fails to demonstrate that Roy discloses or renders obvious that “the client and the server both locally ... execute the view-generating and controller logic” as required by Claim 1(d). Ex. 1001 at 9:51–54. Under the parties’ agreed-upon constructions, “‘view-generating logic’ and ‘controller logic’ executed on the client must be the *same* ‘view-generating logic’ and ‘controller logic’ executed on the server, or some subset thereof.” Pet., 12. Petitioner argues that Roy “inherently” discloses this limitation, Pet., 40–41, 43, but it fails to even argue, let alone demonstrate through supporting evidence, that Roy meets the black-letter legal standard for inherent disclosure by “*necessarily*” teaching that limitation. This deficiency permeates every Ground. Petitioner further fails, in connection with Ground 2, to articulate any reasons to modify Roy to provide that the client and server execute the same view-generating and controller logic. Pet., 44–45.

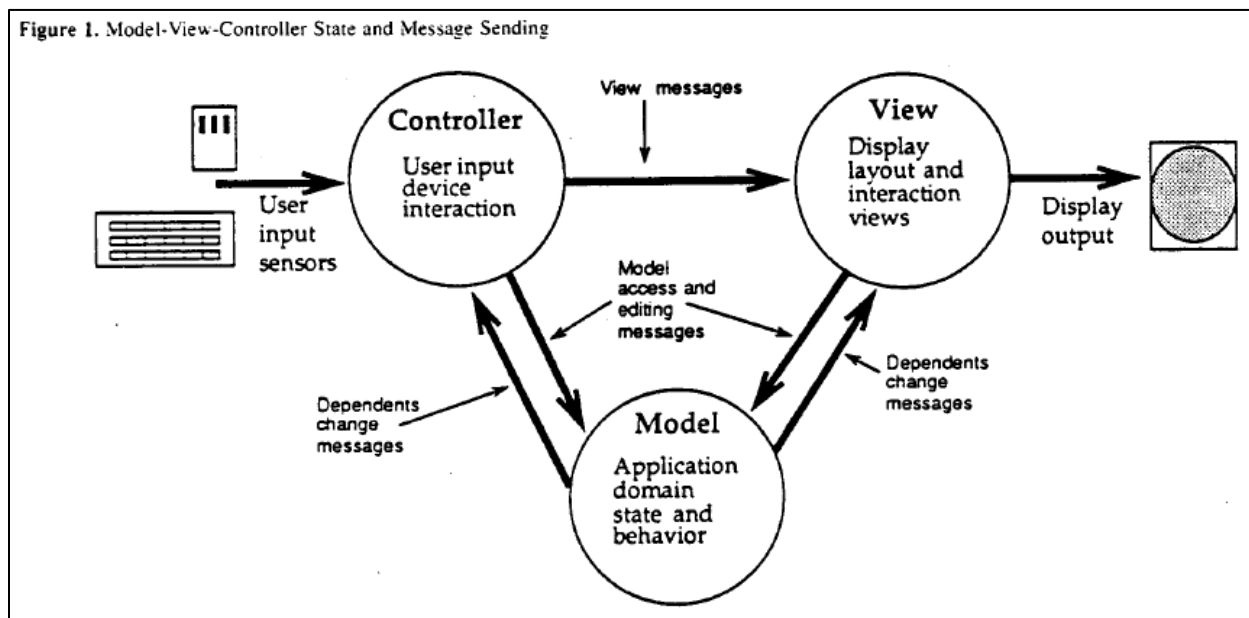
Accordingly, IBM respectfully requests that the Board deny institution.

II. OVERVIEW OF THE ’719 PATENT

A. Conventional MVC Architectures Had Shortcomings

In the 1970s, network-based client/server applications typically executed on large mainframe systems (servers), to which client devices connected via a network. Ex. 1001 at 1:17–33. Internet users’ client devices acted only as a display to view the master application data hosted on the web server. Ex. 1001 at 1:17–22. Ex. 2031, ¶ 37.

Early network-based client/server applications employed a conventional Model-View-Controller (“MVC”) architecture, a paradigm for application programming introduced in a 1988 paper entitled “A Cookbook for Using the Model-View-Controller User Interface Paradigm in Smalltalk-80” by G. E. Krasner and S. T. Pope (“Krasner-Pope”). Ex. 2032; Ex. 1001 at 1:35–41 (incorporating Krasner-Pope by reference). Krasner-Pope explains that the model, view, and controller objects maintain the application state, display the application to the user, and respond to user requests to update the state and display. Ex. 2032 at 1–2; Ex. 1001 at 1:41–47. Ex. 2031, ¶ 38.



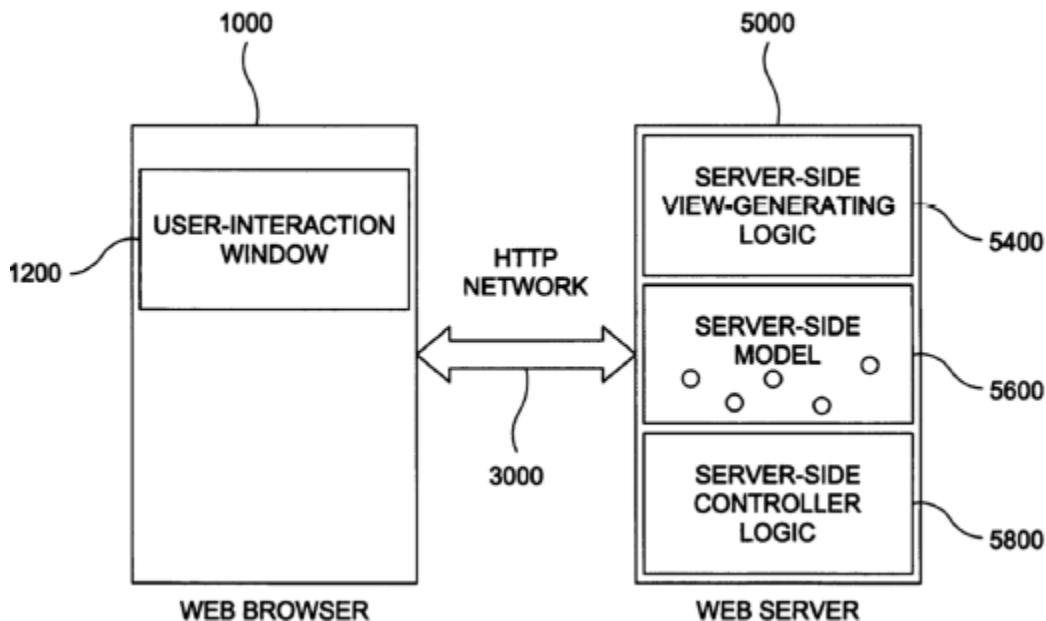
Ex. 2032 at 2.

Before the invention of the '719 Patent, network-based applications were typically developed as “fat-client” applications optimized for *client-side* execution.

Ex. 1001 at 1:34–2:5. Early network-based applications preferred the “fat-client” approach because of its improved flexibility and performance, but that approach was rife with challenges related to distributing and maintaining the applications. Ex. 1001 at 1:52–59. For example, “[u]sers tended to customize their client system, and this made it difficult to develop a client-side software base that would work properly on all systems.” Ex. 1001 at 1:54–57. Ex. 2031, ¶ 39

As the world wide web became more popular, “thin-client” applications—optimized for *server-side* execution—became increasingly popular. Ex. 1001 at 2:6–14. Unlike their fat-client counterpart, in a thin-client application “only the browser display logic, i.e., the application-independent display logic” executes on the client, so “[e]ach time the user interacts with the View, the remote server is notified to update the View and/or Model.” Ex. 1001 at 2:6–14. Ex. 2031, ¶ 40.

Figure 2 provides an example “illustrating a conventional thin-client architecture.” Ex. 1001 at 5:7–9, Fig. 2. In a thin client application, the “web browser” (i.e., a client) is configured to include only a “user-interaction window” whereas the “web server” (i.e., a server) is configured to store the “server-side model” and to execute the “server-side view-generating logic” and “server-side controller logic.” Ex. 1001, Fig. 2. Ex. 2031, ¶ 41.



Ex. 1001, Fig. 2.

Thin-client applications, however, likewise, had shortcomings. For example, thin-client applications were susceptible to poor performance and excess latency because thin-client applications require the client to interact with the server every time the user updates their display. Ex. 1001 at 2:25–31. Ex. 2031, ¶ 42.

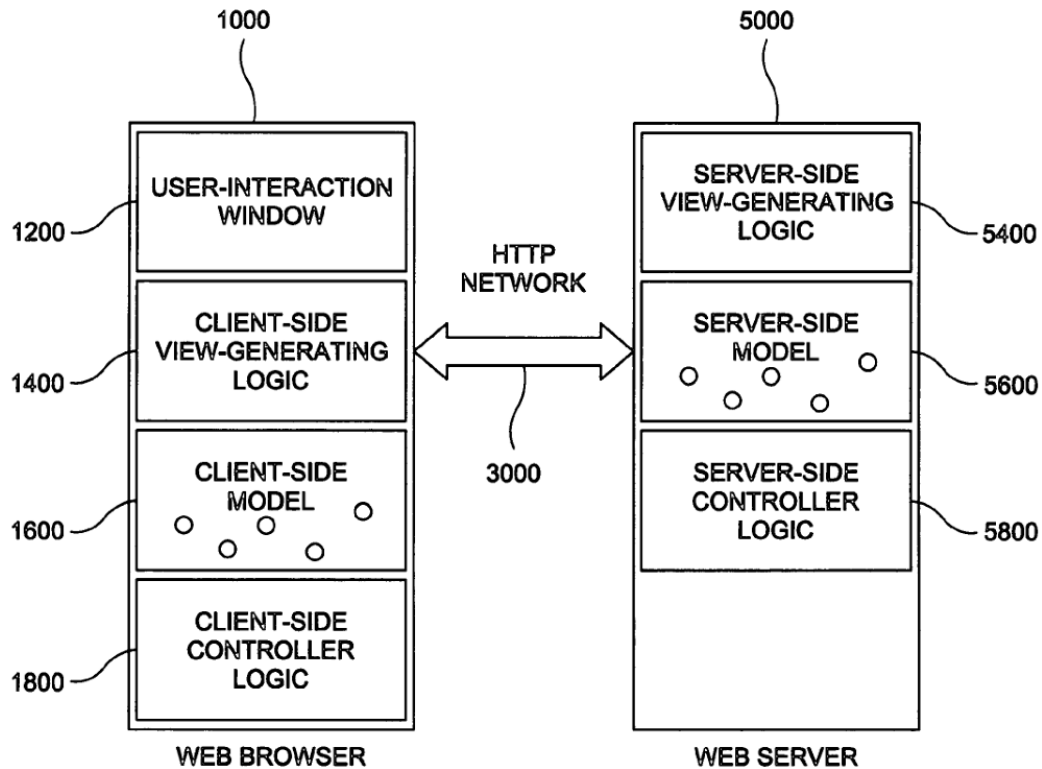
B. IBM’s Inventors Developed A Solution That Addressed Limitations In The Existing MVC Architecture

The inventors of the ’719 Patent redressed the shortcomings of both thin-client and fat-client applications by inventing a *dual*-MVC architecture in which “the client and server both maintain MVC components.” Ex. 1001 at 5:24–26. In a dual-MVC architecture, the server is configured to store a model and to execute view-generating and controller logic associated with the application, and the client is configured to store a *subset* of the model and execute a *subset* of the *same* view-

generating and controller logic associated with the application. Ex. 1001 at 5:14–26, Cl. 1. Ex. 2031, ¶ 43.

The '719 Patent's dual-MVC architecture addresses the latency problems of thin-client applications because, by storing a subset of the MVC locally at clients, the clients can reduce the number of required interactions with the server. Ex. 1001 at 2:39–49; *see also* Ex. 1001 at 3:14–25, 5:28–30. The dual-MVC architecture also addresses the challenges of fat-client applications. Unlike fat-client applications, the dual-MVC approach “does not require any application installation or persistence on the client” and, instead, “the MVC of the application is preferably downloaded from the server on demand by the user of the client device, avoiding the install process, and is preferably not saved permanently to the disk of the client.” Ex. 1001 at 2:47–49, 3:1–13. Ex. 2031, ¶ 44.

Figure 3 of the '719 Patent depicts the inventive dual-MVC architecture. Ex. 1001 at 4:16–17. Here, the “web browser” (i.e., a client) is configured to include a “user-interaction window,” store the “client-side model,” and execute the “client-side view-generating logic” and “client-side controller logic.” Ex. 1001, Fig. 3. The “web server” (i.e., a server) is configured to store the “server-side model” and to execute the “server-side view-generating logic” and “server-side controller logic.” Ex. 1001, Fig. 3. Ex. 2031, ¶ 45.



Ex. 1001, Fig. 3.

The '719 Patent's dual-MVC architecture reduces the number of communications between the client and the server because "the client machine maintains part of the Model." Ex. 1001 at 3:14–16. "When part of the Model is maintained on the client, the client has the ability to respond to user interaction without paying the performance penalty of continually interacting with the server." Ex. 1001 at 3:22–25. The dual-MVC invention enables such efficiencies because "both the client and server execute the same model and view-generating and controller logic associated with the application (albeit, with the client executing at least a subset thereof)." Ex. 1002 at 186 (underline in original). That inventive

concept—that the client and server execute the *same* model, the *same* view-generating logic, and the *same* controller logic, or some subset thereof—is reflected in the parties’ agreed upon construction that, in the context of Claims 1, 13, 25, and 26, “‘view-generating logic’ and ‘controller logic’ executed on client must be the same ‘view-generating logic’ and ‘controller logic’ executed on the server, or some subset thereof.” Pet., 12. Ex. 2031, ¶¶ 46–47.

C. Prosecution History Of The ’719 Patent

As previewed above in Section II.B, the prosecution history of the ’719 Patent confirms that the view-generating logic and controller logic executed on client must be the same view-generating logic and controller logic executed on the server, or some subset thereof. Ex. 2031, ¶ 50.

Specifically, during the prosecution of the ’719 Patent, the examiner rejected the pending claims in light of a prior art reference that disclosed the client and server running *different* view-generating and controller logic. Ex. 1002 at 184–86. In response, IBM explained that “in the claimed approach, both the client and the server execute the same model and view-generating and controller logic associated with the application (albeit, with the client executing at least a subset thereof).” Ex. 1002 at 186 (underline in original). IBM later amended the claims to reflect that, in the claimed invention, both the client and the server execute the *same* view-generating

logic and the *same* controller logic associated with the application, or some subset thereof:

1. (Currently Amended) A method for use in a client/server system of reducing interactions between a client and server in association with an application being accessed by the client at the server, the method comprising the steps of:

configuring the server to store a model associated with the application and to execute view-generating and controller logic associated with the application; and

configuring the client to store at least a subset of the model associated with the application and to execute at least a subset of the view-generating and controller logic associated with the application;

wherein one or more portions of the application are performed at the client without the client having to interact with the server, and further wherein the client and the server both locally execute the respective model and view-generating and controller logic ~~resident thereon~~ associated therewith.

Ex. 1002 at 293. Ex. 2031, ¶¶ 50–51.

As explained below, however, Roy fails to at least demonstrate the step of configuring the server and client to execute the *same* view-generating and controller logic, or a subset thereof. Ex. 2031, ¶ 52.

III. LEVEL OF ORDINARY SKILL IN THE ART

A person of ordinary skill in the art (“POSITA”) at the time the ’719 Patent was filed would have had two years of computer programming and network experience, including UI (user interface) design and being knowledgeable with MVC (Model-view-controller) concepts. The differences between IBM’s and Petitioner’s definitions of a POSITA do not materially impact the arguments presented in the POPR. Ex. 2031, ¶¶ 33–36.

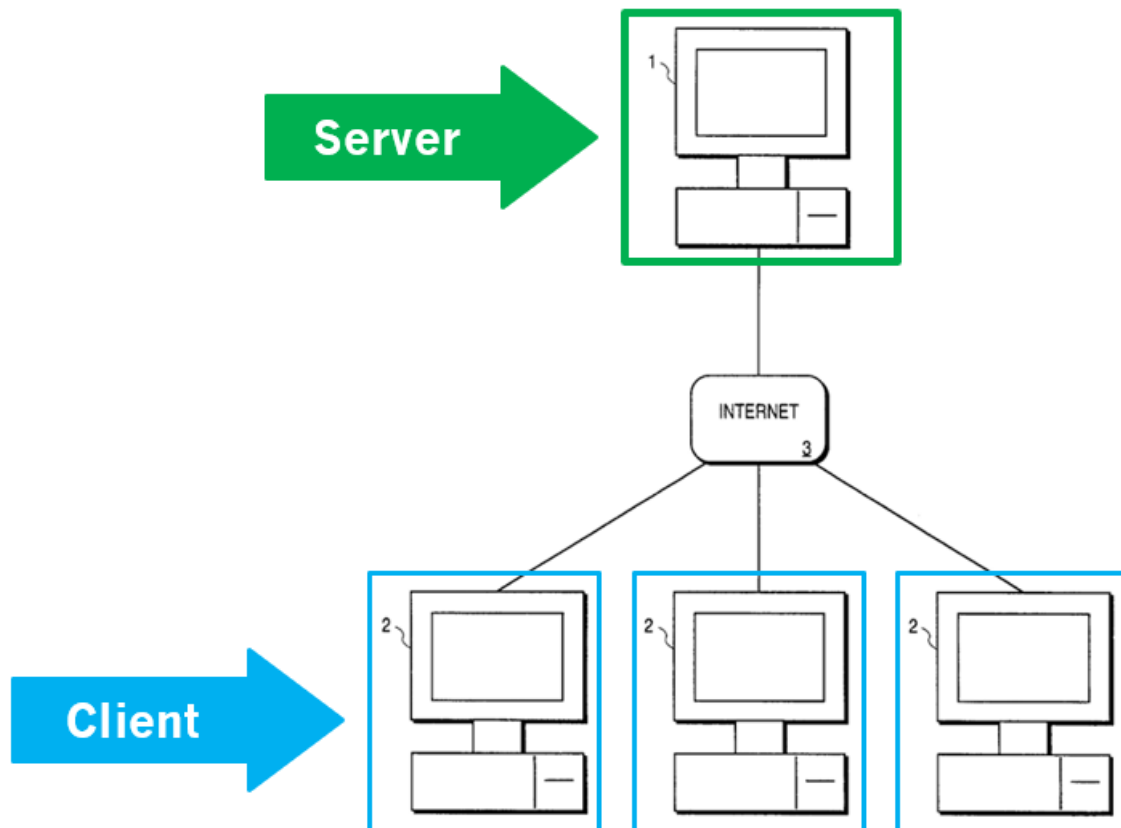
IV. OVERVIEW OF THE ROY REFERENCE²

A. Roy Discloses A Computer Network System With Peer-To-Peer Networking Capabilities

Roy discloses “[a] method and apparatus for reducing network latency during execution of a multi-player game across a computer network.” Ex. 1004 at Abstract. In Roy, a “central server computer 1 is coupled via the Internet 3 to a number of client computer systems 2,” as shown in Figure 1 below. Ex. 1004 at 3:48–51. Roy explains that “every machine [including the client computer device] that participates in the game runs software that is referred to herein as an ‘event server,’” with one

² For purposes of this POPR, IBM discusses the Roy reference only because Petitioner’s Grounds 1 and 2 challenging the validity of the ’719 Patent’s independent claims—Claims 1, 13, 25, and 26—are based solely on Roy. Pet., 3, 30, 44. Ex. 2031, ¶ 55.

event server classified as the “master event server” and the others as “slave event servers.” Ex. 1004 at 4:37–44. The “master database is located on the same machine as the master event server” and “[s]lave copies of the master database (or portions thereof) are stored on each machine other than the machine in which the master event server is located.” Ex. 1004 at 4:44–48. “By providing slave copies of the database in every other machine involved in the game, the need to send rendering commands across the network is reduced or eliminated, which reduces bandwidth consumption.” Ex. 1004 at 4:51–55. Ex. 2031, ¶¶ 56–60.



Ex. 1004, Fig. 1 (annotated).

Figure 5A of Roy depicts an embodiment in which the central server computer is coupled to two client computer systems. Ex. 1004, Fig. 5A. Roy explains that, during an online multi-player game, like simulated American football, Players A and B operate client computer systems 60 and 62, respectively. Ex. 1004 at 5:17–36. “Initially, the master database 71 and the master event server 66 are located in a central server computer system 61.” Ex. 1004 at 5:23–25. Slave event servers 65 and 67 and slave copies of the master databases 68 and 72 are located on Player A’s and B’s computer systems 60 and 62. Ex. 1004 at 5:25–29. Player A’s and B’s displays 70 and 74 are updated by rendering commands 69 and 73 from the slave copies of the master database 68 and 72, respectively, rather than from the master database 71 itself, and that purportedly reduces bandwidth consumption. Ex. 1004 at 5:29–36. Ex. 2031, ¶ 61.

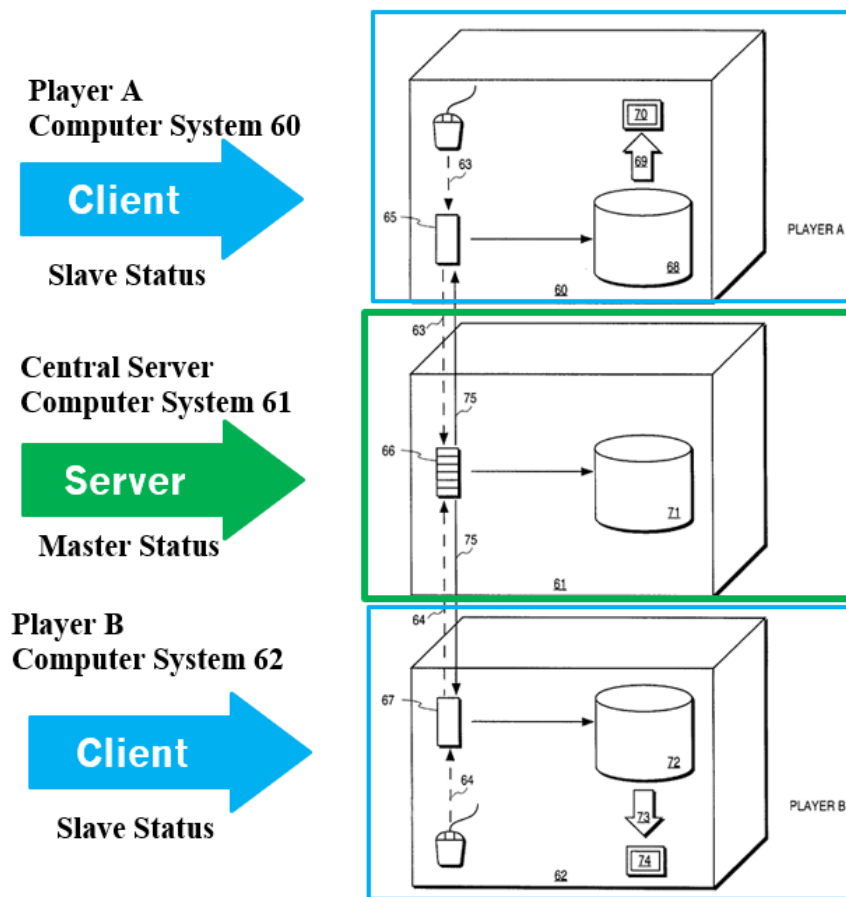


FIG. 5A

Ex. 1004, Fig. 5A (annotated).

Roy discloses that the status of the master database 71 and the master event server 66 can be “migrated” to a player’s client computer system—a process that involves changing the status of the client computer system’s existing database and event server from “slave” to “master.” Ex. 1004 at 5:65–6:14. For example, as the game progresses, “a determination is made of which computer system participating in the game is used by the player having the role which requires the least latency.” Ex. 1004 at 5:65–6:1. The status of the “master” event server and “master” database

is then migrated to *that player (i.e., client computer system)* with the role requiring the least latency. Ex. 1004 at 6:1–2. Figure 5B depicts Roy’s architecture after a migration event in which Player B’s computer system 62 requires the least latency. Ex. 1004 at 6:2–14. As a result of the migration event, “the status of event server 66 is changed from master event server to slave event server, the status of event server 67 is changed from slave event server to master event server,” and “the master database 71 is migrated to computer system 62.” Ex. 1004 at 6:6–11. Conversely, the status of slave event server 65 on Player A’s computer system 60 remains unchanged. Ex. 1004 at 6:11–12. Ex. 2031, ¶¶ 62–63.

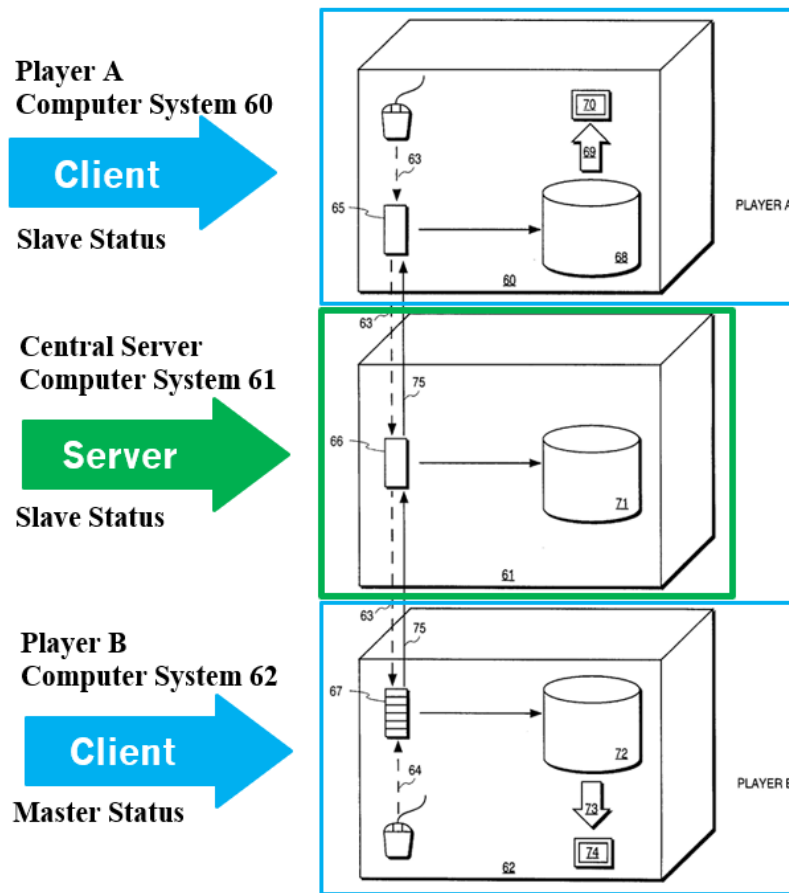


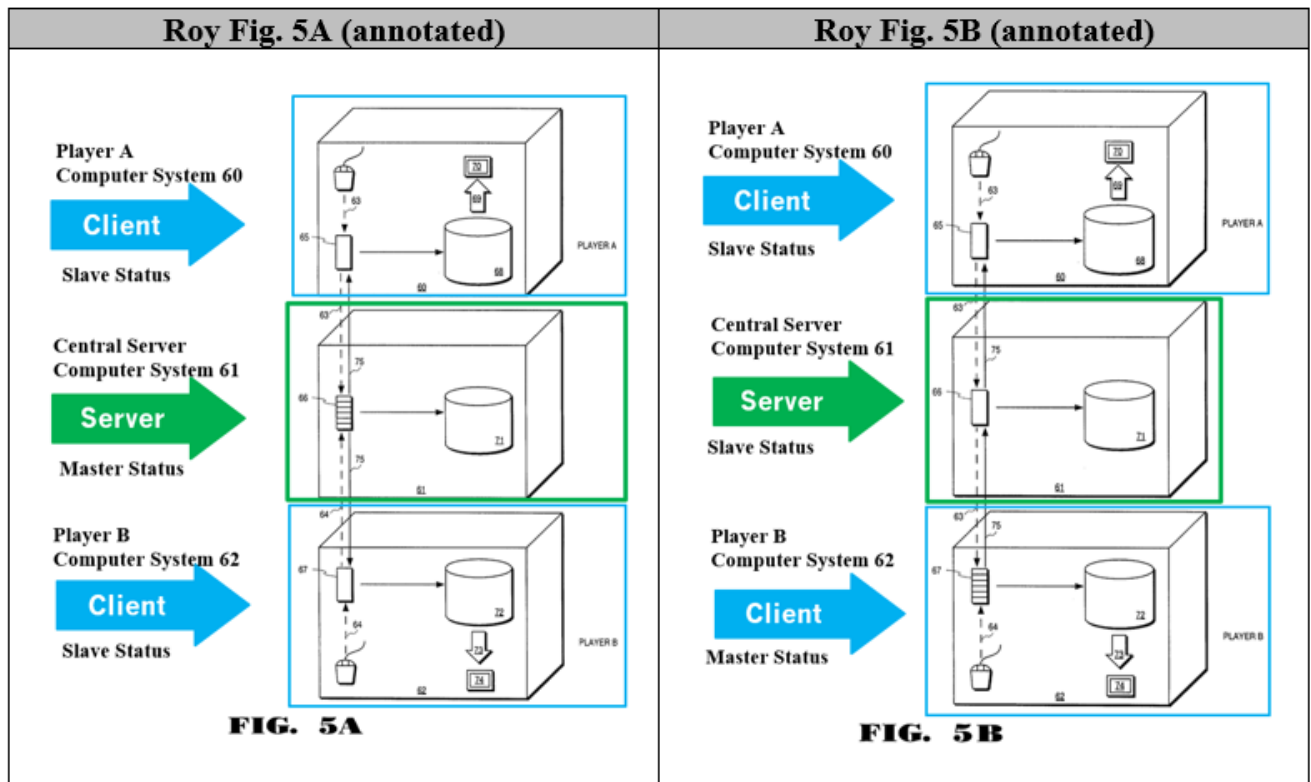
FIG. 5B

Ex. 1004, Fig. 5B (annotated).

Once the master event server and master database are migrated to the client system B, the client system B and the client system A can interact directly as a peer-to-peer system, as opposed to as a client-to-server system that interacts through a server computer system 61. For instance, Roy explains that, after the migration event depicted in Figure 5B, “it may not be desirable from a latency standpoint (or for other reasons) to require events originating from computer system 60 to pass

through the central server computer system 61.” Ex. 1004 at 8:52–56. Ex. 2031, ¶ 64.

Contrary to Petitioner’s labeling of Player B’s computer system 62 as a “server” (*see* Pet., 16), Roy consistently refers to computer system 62 as a *client* device, even after the migration event. *See, e.g.*, Ex. 1004 at Cl. 52 (“The program storage medium of claim 50, wherein the method steps further comprise the step of using the *server computer* to forward a slave event received from a *second one of the client computers* to the master event server migrated to the *first one of the client computers.*”) (emphases added); Cl. 53 (“The program storage medium of claim 50, wherein the method steps further comprise the step of using the *server computer* to forward a master event received from the master event server migrated to the *first one of the client computers to a second one of the client computers.*”) (emphases added). As shown in the diagram below comparing Roy’s Fig. 5A with Fig. 5B, Player B’s computer system 62 does not become a “server” after a migration event.



Compare Ex. 1004, Fig. 5A (annotated) with Ex. 1004, Fig. 5B (annotated).

As the above comparison shows, after a migration event, Roy’s “server” (central server computer system 61, marked in green) remains a “server.” Similarly, Roy’s “client” devices (Player A’s and B’s computer systems 60 and 62, marked in blue) remain “client” devices. Ex. 2031, ¶¶ 65–66.

V. CLAIM CONSTRUCTION

In the long-pending, parallel district court litigation, Petitioner and IBM agreed on constructions of certain terms and phrases, but dispute constructions of other terms and phrases. Pet., 12–13. For purposes of this POPR only, IBM submits that additional constructions are not necessary. Ex. 2031, ¶¶ 53–54.

VI. THE '719 PATENT CLAIMS ARE PATENTABLE OVER PETITIONER'S GROUNDS 1–4

A. GROUND 1: Petitioner Does Not Establish A Reasonable Likelihood Of Prevailing Against Any Claim Under Ground 1 Because Roy Fails To Disclose, Expressly or Inherently, All Limitations of Challenged Claims.

In Ground 1, Petitioner asserts that Roy anticipates all independent claims 1, 13, 25, and 26. Pet., 3, 30. As explained below, Ground 1 fails for at least two reasons.

First, Roy fails to disclose “configuring the *server* to ... execute view-generating ... logic associated with the application” (limitations 1(a), 13(a), 25(a), 26(a)) (hereinafter Limitation (a)). Petitioner incorrectly contends that “Player B in Figure 6 demonstrates the server side of Roy’s ‘dual-MVC’ architecture” (Pet., 18) because, according to Petitioner, “the computer system containing the master event server and master database is the ‘server’” (Pet., 16). But as explained below, Roy discloses that Player B’s computer system is a *client* device regardless of whether it contains the master event server and master database. *See infra* Section VI.A.1; Ex. 2031, ¶¶ 69–82.

Second, Roy fails to disclose that “the client and the server both locally maintain at least a portion of the model and execute the view-generating and controller logic associated therewith” (limitations 1(d), 13(d), 25(d), 26(d)) (hereinafter Limitation (d)). Petitioner appears to acknowledge that Roy does not

expressly disclose Limitation (d) under the agreed-upon construction of the phrase “the client and the server both locally ... execute the view-generating and controller logic,” asserting instead that Roy purports to disclose Limitation (d) under the doctrine of *inherency*. Pet., 40–41, 43. But Petitioner’s inherency arguments fail to properly allege that Roy necessarily teaches the missing limitation, as well as lack evidentiary support and contravene Roy’s express disclosures. *See infra* Section VI.A.2; Ex. 2031, ¶¶ 83–125.

1. **Roy Does Not Disclose “Configuring The Server To ... Execute View-Generating ... Logic Associated With The Application,” As Required By Limitations 1(a), 13(a), 25(a), 26(a)**
 - a. **Petitioner Fails To Demonstrate That Roy Discloses That Player B’s Computer System 62 Is A “Server,” As Claimed In Limitation (a)**

Every independent claim of the ’719 Patent requires Limitation (a): “configuring the *server* to ... execute view-generating ... logic associated with the application.” *See* Ex. 1001 at Cls. 1, 13, 25, 26. Petitioner incorrectly asserts that Limitation (a)’s “server” element is taught by Roy’s “computer system 62 (used by Player B),” Ex. 1004 at 5:27–28. *See* Pet., 18 (alleging that “*Player B* in Figure 6 *demonstrates the server side* of Roy’s ‘dual-MVC’ architecture.”), 34 (“[I]n Figure 6, *computer 62 is the server* ...”) (emphases added). Ex. 2031, ¶¶ 69–70. Indeed, Petitioner’s expert, Dr. Bennett, presented this same argument in the *Zynga* Litigation; the jury and district court rejected it in upholding the validity of the ’719

Patent. *See, e.g.*, Ex. 2007, ¶ 508 (Dr. Bennett stating that “computer 62 acts as a server in this client/server system”); Paper 8 at 26–33, 35–36.

Contrary to Petitioner’s assertion, Roy teaches that Player B’s computer system 62 is a *client*—not a server. Indeed, Roy consistently refers to Player B’s computer system 62—what Petitioner incorrectly asserts is the server—as a *client* device. For example, Roy explains that the “network configuration in which the present invention may be implemented” is a “central *server* computer 1 [that] is coupled via the Internet 3 to a number of *client computer systems* 2.” Ex. 1004 at 3:50–51. When applied to Roy’s exemplary embodiments—for example, Figures 5A and 6 (which Petitioner relies on) as well as Figure 5B—the “central server computer 1” corresponds to the “computer system 61,” and the “*client* computer systems 2” corresponds to the computer systems 60 and 62 operated by Players A and B. *See* Pet., 17 n.8 (admitting that the “embodiment of Roy described in Fig.5A uses *computer 61* ‘initially’ as a ‘*central server*’”) (emphases added); *see also* Ex. 1004 at Figs. 5A–6; Pet., 16–18. Indeed, Roy consistently describes the invention as being implemented on a “server computer and a plurality of client computers.” Ex. 1004 at Cls. 50, 52–55. Ex. 2031, ¶¶ 71–72.

Petitioner’s expert, Dr. Bennett, also agrees that Player B’s computer system 62 is not a server. As explained in IBM’s brief in support of discretionary denial, *see* Paper 8, 35–36, Dr. Bennett repeats his opinion from the *Zynga* Litigation, which

both the district court and the jury rejected. Indeed, in the *Zynga* Litigation, Dr. Bennett admitted at trial that Roy does *not* say that Player B's computer system 62 is a server:

Q. And figure 6, that's from Roy, the Roy patent?

A. It is.

Q. And over here in the callout, you've labeled this Box 62 as "server." Do you see that?

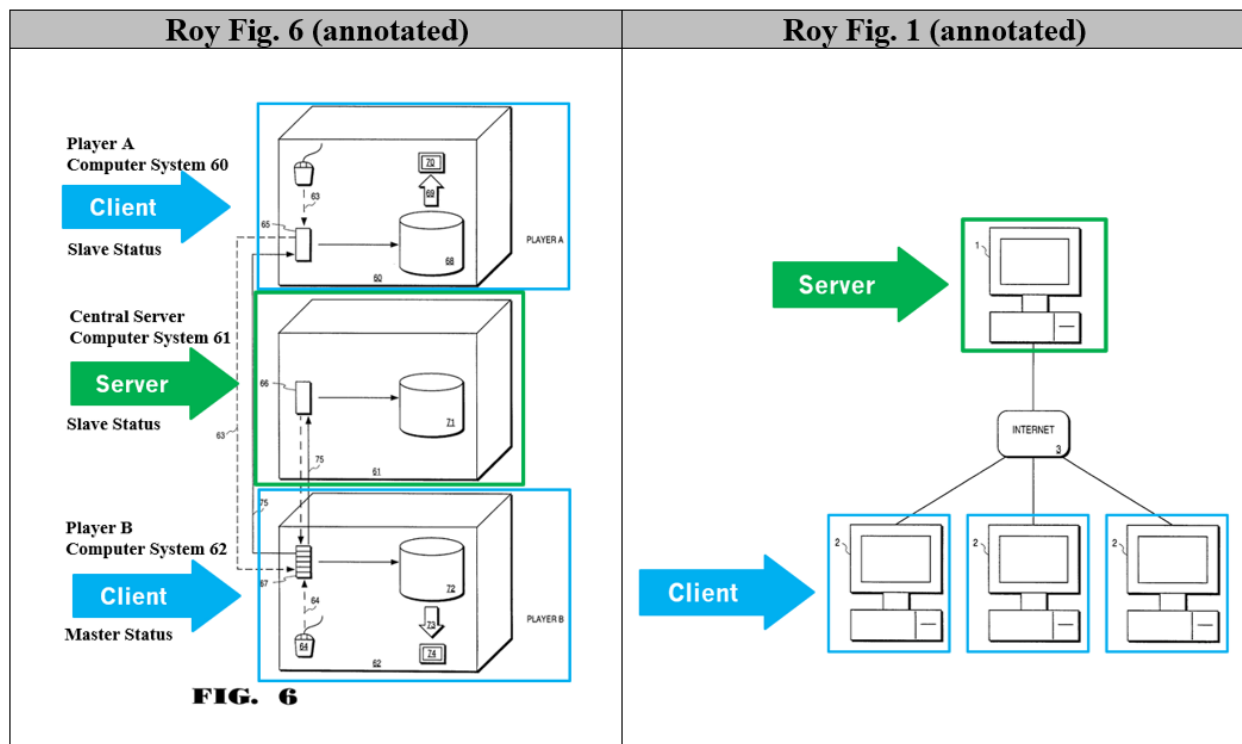
A. Correct.

...

Q. Roy doesn't say that 62 is a server, it doesn't say that, right?

A. No, not specifically.

Ex. 2008 at 1035:9–20 (emphases added).



Compare Ex. 1004, Fig. 6 (annotated) *with* Ex. 1004, Fig. 1 (annotated) (showing that Roy’s “server” in Figure 1 refers to central server computer system 61 in Figure 6 (marked in green) and Roy’s “client” devices in Figure 1 refers to Player A’s and B’s computer systems 60 and 62 (marked in blue)). Ex. 2031, ¶ 73.

Even Petitioner appears to acknowledge that Roy’s client computer 62 is not a server. Pet., 17 n.8 (acknowledging that “Fig.5A uses computer 61 ... as a ‘central server.’”). Petitioner nonetheless asserts that “[i]n Roy, the computer system containing the master event server and master database is the ‘server’, [and] the other computers are clients.” Pet., 16. Specifically, Petitioner asserts that after “the master event server and master database (i.e., the ‘server’) have migrated to the most active user [i.e., Player B’s computer system 62 in Figures 5B and 6], ... computer 62 is now the master server.” Pet., 17 n.8. Ex. 2031, ¶ 74.

Petitioner’s attempt to transform Player B’s expressly disclosed “client” system 62 into a purported “server” fails for two reasons. **First**, Roy nowhere teaches, expressly or otherwise, that the client computer system—like Player B’s computer system 62—suddenly becomes a “server” just because a component in the client computer system receives software or data for the master event server and the master database. Contrary to Petitioner’s position—and as Petitioner’s expert Dr. Bennett admitted at trial during the *Zynga* Litigation—Roy never calls Player B’s computer system 62 a server. Ex. 2008 at 1035:9–20. Rather, Roy explains that the

“network configuration in which the present invention may be implemented” is a “central *server* computer 1 [that] is coupled via the Internet 3 to a number of *client computer systems* 2.” Ex. 1004 at 3:48–51. Ex. 2031, ¶¶ 75–77.

Because Roy never teaches that the computer system 62 is Limitation (a)’s “server,” Petitioner relies heavily on the declaration of Dr. Bennett, who merely parrots the same conclusory allegation without any additional evidence or analysis. *Compare* Pet., 16 (“In Roy, the computer system containing the master event server and master database is the ‘server.’”) *with* Ex. 1003 ¶ 85 (“In Roy, the computer system containing the master event server and master database are [sic] considered the ‘server.’”). Ex. 2031, ¶ 78. Dr. Bennett’s unsupported, *ipse dixit* testimony should be given little weight, if any. *See Xerox Corp. v. Bytemark, Inc.*, IPR2022-00624, Paper 9, at 15–17 (PTAB Aug. 24, 2022) (precedential) (holding an expert’s declaration testimony is entitled to little weight when it “is conclusory and unsupported, [and] adds little to the conclusory assertion for which it is offered to support”); *Apple Inc. v. MPH Tech. Oy*, 28 F.4th 254, 262–63 (Fed. Cir. 2022) (noting that “[t]he Board was free to reject [Petitioner]’s expert testimony based on a lack of factual support”) (citing *TQ Delta, LLC v. Cisco Sys., Inc.*, 942 F.3d 1352, 1362–63 (Fed. Cir. 2019)).

Second, contrary to Petitioner’s argument, Roy does ***not*** disclose or even purport to suggest that a migration event transforms Player B’s client system 62 to a

server. As explained above, to reduce latency, Roy explains that a migration event causes a *status change* in the event server and database on both Player B’s computer system 62 and the central server computer system 61:

Consequently, master event server 66 is migrated to computer system 62. More specifically, the status of event server 66 is changed from master event server to slave event server, and the status of event server 67 is changed from slave event server to master event server. Similarly, the master database 71 is migrated to computer system 62.

Ex. 1004 at 6:5–11. Ex. 2031, ¶ 79.

But even after “the master event server 66 is migrated” Ex. 1004 at 6:5–6 to a client computer (e.g., Player B’s computer system 62), the central server computer system 61 remains a “server computer” and Player B’s computer system 62 remains a “client computer” because a migration event is nothing more than a “status ... change[]” effecting the *software* (*i.e.*, the event server and database) on each device. Ex. 1004 at 6:5–11. Indeed, Roy’s claims confirm that the central server computer system 61 remains the “server computer” and Player B’s computer system 62 remains a “client computer” even after a migration event. *See* Ex. 1004 at Cls. 50, 52–53. For example, the method steps of independent claim 50 require “migrating the master event server from the *server computer* to the first one of the *client computers*.” Ex. 1004 at Cl. 50 (emphasis added). Claims 52 and 53, which depend

on Claim 50, make clear that the “client computer” remains a “client computer” even though the master event server migrated to it, and the “server computer” remains a “server computer” even though its event server lost its master status:

52. “The program storage medium of claim 50, wherein the method steps further comprise the step of *using the server computer to forward a slave event* received from a second one of the client computers *to the master event server migrated to the first one of the client computers*.”

Ex. 1004 at Cl. 52 (continuing to refer to the server computer with a slave status as a “server” and a client computer with a master status as a “client”) (emphases added).

53. The program storage medium of claim 50, wherein the method steps further comprise the step of *using the server computer to forward a master event received from the master event server migrated to the first one of the client computers* to a second one of the client computers.

Ex. 1004 at Cl. 53 (same) (emphases added). *See also, e.g.,* Ex. 1004 at Cls. 54–55. That claim language expressly recites a “client computer” with a master event server that receives slave events or sends master events, which confirms that the “client computer[]” remains a “client” even after the master event server is migrated to it. Ex. 2031, ¶¶ 80–81.

Roy’s express labeling of a computer containing the master event server as a “client” fatally undermines Petitioner’s assertion that any computer system

containing the master event server and master database should be considered the “server.” *See* Pet., 16. In short, Petitioner relied on Player B’s computer system 62 as the “server,” but Roy makes clear that computer system 62 is the “client”—not the “server.” Accordingly, Petitioner fails to demonstrate that Roy discloses *any* aspect of the “server” element as claimed in Limitation (a). Ex. 2031, ¶ 82.

2. Roy Does Not Disclose “Wherein The Client And The Server Both Locally Maintain At Least A Portion Of The Model And Execute The View-Generating And Controller Logic Associated Therewith” As Required By Limitation (d)

Every independent claim of the ’719 Patent also includes Limitation (d), which requires “wherein the client and the server both locally maintain at least a portion of the model and execute the view-generating and controller logic associated therewith.” *See* Ex. 1001 at Cls. 1, 13, 25, 26. *See also* Pet., 12 (identifying Claims “1, 13, 25, 26” as relevant to the claim element “the client and the server both locally ... execute the view-generating and controller logic”). Petitioner concedes that Limitation (d) requires that the “‘view-generating logic’ and ‘controller logic’ executed on client *must be the same* ‘view-generating logic’ and ‘controller logic’ executed on the server, or some subset thereof.” Pet., 12 (emphasis added). For at least the reasons below, Petitioner fails to show that Roy discloses Limitation (d). Ex. 2031, ¶¶ 83–84.

a. Petitioner Fails To Demonstrate That Roy Discloses Limitation (d)’s Requirement That The Alleged “View-Generating Logic” Executed On The Client Is The Same As The Alleged “View Generating Logic” Executed On The Server, Or A Subset Thereof

Petitioner and IBM agree that Limitation (d) requires that the “‘view-generating logic’ ... executed on client *must be the same* ‘view-generating logic’ ... executed on the server, or some subset thereof.” Pet., 12 (emphasis added). Petitioner, however, fails to demonstrate that Roy discloses Limitation (d) under the agreed-upon construction. Ex. 2031, ¶¶ 83–84, 87.

Petitioner incorrectly asserts that Roy’s “rendering commands” teaches Limitation (d)’s “view-generating logic” element (i.e., “program code ...”). See Pet., 43 (“[T]he client and server ‘execute the *view-generating ... logic*’ (e.g., *rendering commands* 69 updating display 70 and *rendering commands* 73 updating display 74.”)) (emphases added). Specifically, Petitioner purports to map the “client ... execut[ing] the view-generating ... logic” element to Roy’s “rendering commands 69” issued in Player A’s computer system 60 (marked in the figure below in blue), and the “server execut[ing] the view-generating ... logic” element to Roy’s “rendering commands 73” issued in Player B’s computer system 62 (marked in the figure below in green):

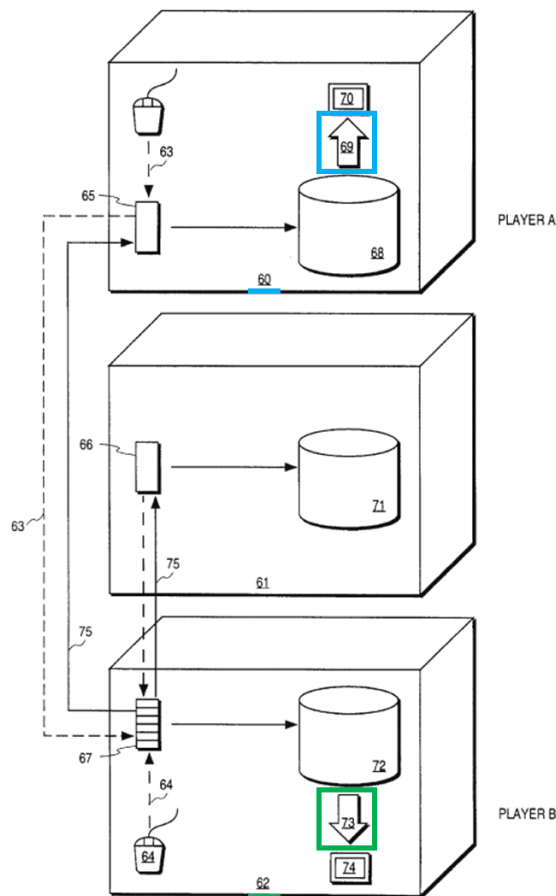


FIG. 6

Ex. 1004, Fig. 6 (annotations added); Pet., 43. *See also* Pet., 16–17 (“Roy likewise describes: ... (2) **view-generating logic** in client-side **rendering commands 69** and **display 70** and server-side **rendering commands 73** and **display 74**”) (emphases added, coloring in original), 39. Accordingly, under the agreed-upon construction, Petitioner must show that the **rendering commands 69** are either the same as the **rendering commands 73**, or a subset thereof. Ex. 2031, ¶¶ 85–86.

Petitioner purports to rely on the doctrine of inherency to suggest that the **rendering commands 69** are either the same as the **rendering commands 73**, or a

subset thereof. Pet., 40. Specifically, Petitioner asserts that “[a] POSITA would understand that the client-side and server-side view-generating logic ... are the same, as discussed above,” presumably referring to Petitioner’s argument with respect to “Limitation (b).” Pet., 43. And with respect to Limitation (b), Petitioner asserts that the purported view-generating logic—i.e., the rendering commands—would “*inherently be the same* on each computer (and therefore the same whether on the server or client) for efficiency, economic, and common sense reasons.” Pet., 40 (emphasis added). Petitioner even claims that “[i]t would be *particularly inherent* to a POSITA because in Roy’s application, each user sees the same world and it would not make sense to use different logic to generate views of the same world for each computer, especially given Roy’s teachings that client and server computers have the same architecture.” Pet., 40 (emphasis added). Ex. 2031, ¶¶ 88–89.

Dr. Bennett presented this same opinion in the *Zynga* Litigation, and the jury and the district court rejected it in upholding the validity of the ’719 Patent. *See, e.g.*, Ex. 2007, ¶ 528 (“Further, it would be understood to a POSITA, or at least rendered obvious, that the view-generating logic would be the same on each computer for efficiency and economic reasons.”). *See also* Paper 8 at 26–33, 36–38. And the jury and district court had good reason for doing so, as Petitioner’s inherency argument is deficient for at least three reasons.

First, it ignores the black-letter law that “anticipation by inherent disclosure is appropriate **only when** the reference discloses prior art that **must necessarily** include the unstated limitation.” *Therasense, Inc. v. Becton, Dickinson & Co.*, 593 F.3d 1325, 1332 (Fed. Cir. 2010) (quoting *Transclean Corp. v. Bridgewood Servs., Inc.*, 290 F.3d 1364, 1373 (Fed. Cir. 2002)). Petitioner does not even attempt to show that Roy “must **necessarily**” teach that the alleged logic in the client device—**rendering commands 69**—is the same or a subset of the alleged logic in the server—**rendering commands 73**. See Pet., 40, 43. For example, Petitioner asserts that “it would not make sense to use different logic” at the alleged client device 60 and the alleged server 62. Pet., 40. But even if that were true, that still falls short of alleging that the alleged client device 60 and the alleged server 62 “**must necessarily**” use the same logic. Ex. 2031, ¶¶ 90–92.

Second, Petitioner cannot demonstrate that Roy **inherently** discloses using **rendering commands 69** that are either the same or a subset of the **rendering commands 73**. Petitioner asserts that “it would not make sense to use different logic” at the alleged client device 60 and the alleged server 62 supposedly because “in Roy’s application, each user sees the same world”—i.e., sees the exact same “representation of a subset of the model.” Pet., 12, 40. Petitioner is wrong both on the law and the facts. Ex. 2031, ¶¶ 90, 93.

Under the correct standard for inherency, the operative issue is *not* whether it would “*make sense to use different logic*” at the alleged client device 60 and the alleged server 62 but rather whether it is “*necessary to use the same logic*” at the alleged client device 60 and the alleged server 62. Even assuming for the sake of argument that “it would not make sense to use different logic” at the alleged client device 60 and the alleged server 62—which Patent Owner disagrees for the reasons provided below—that would still fall far short of demonstrating that it would be “necessary” to use the same logic at the alleged client device 60 and the alleged server 62.

Furthermore, it would indeed make sense to use different rendering commands (the alleged logic) at the alleged client device 60 and the alleged server 62. Roy discloses an embodiment involving “a simulated (American) football game in which one player plays offense and a second player plays defense at any given point in time.” Ex. 1004 at 3:23–25. In that embodiment, the player on *offense* would execute *offense*-oriented program code for generating a visual representation of an *offense* point of view of the game world along with offense-specific interaction options (e.g., “movement of the players, including the speed and direction”). Ex. 1004 at 3:23–29. In contrast, the player on *defense* would execute a *different* set of defense-oriented program code for generating a visual representation of a *defense* point of view of the game world along with defense-specific interaction options (e.g.,

“selecting a predefined defensive pattern”). Ex. 1004 at 3:23–33. Accordingly, *different* players are expected to see *different* representations of the game world based on their roles in the game. Ex. 2031, ¶¶ 94–98.

As another example, Roy discloses an embodiment involving “a role-playing fantasy game” where players have either a “superhero” role or a “peripheral” role. Ex. 1004 at 3:35–38. In that embodiment, the “superhero” player would execute “superhero”-oriented program code for generating a visual representation of a “superhero” point of view of the game world along with “very active” superhero-specific interaction options. Ex. 1004 at 3:35–39. In contrast, the “peripheral” player would execute a different set of “peripheral”-oriented program code for generating a visual representation of a “peripheral” point of view of the game world along with “peripheral” control options (e.g., setting game conditions and selecting obstacles). Ex. 1004 at 3:35–47. Thus, again, *different* players would see *different* visual representations of the game world and would execute *different* logic for generating those different visual representations as a result. Ex. 2031, ¶¶ 99–102.

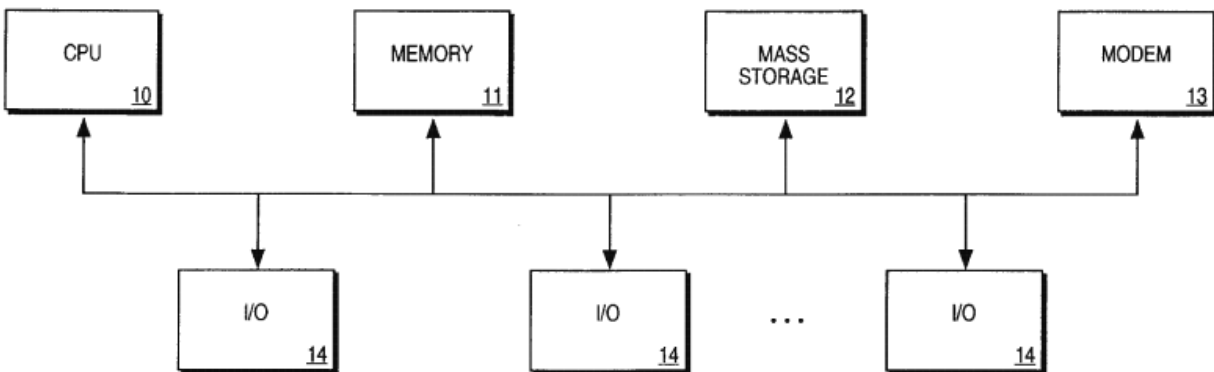
Indeed, both Petitioner and its expert, Dr. Bennett, appear to concede that the *rendering commands 69* would not necessarily be either the same or a subset of the *rendering commands 73*. For example, in arguing that “Roy’s view-generating logic would inherently be the same,” both Petitioner and Dr. Bennett assert that “any POSITA would, *whenever technically feasible*, seek to avoid the expense and

inefficiency of developing different logic and program code.” Ex. 1003, ¶ 155 (emphasis added). *See also* Pet., 40 (citing Ex.1003¶¶154–155), 45. That assertion, however, itself concedes that there would be circumstances in which a POSITA would **not** execute the same view-generating logic on both the client and the server, including in circumstances where it is not “technically feasible” to do so. Accordingly, even the Petitioner and its expert recognize that it is **not necessary** for the view-generating logic to be the same in the client device and the server. Notably, neither Petitioner nor its expert argue, much less show, that it would **necessarily** be “technically feasible” in Roy’s disclosed system to execute the same view-generating logic. Pet., 40, 43, 45. Ex. 2031, ¶¶ 103–06.

Third, Petitioner’s inherency argument is deficient because it is conclusory and unsupported by evidence. For example, Petitioner alleges that “in Roy’s application, each user sees the same world”—i.e., sees the exact same “representation of a subset of the model.” Pet., 40. Petitioner, however, identifies no evidence supporting this conclusory allegation.

As another example, Petitioner incorrectly alleges that “it would not make sense to use different logic to generate views of the same world for each computer, especially given Roy’s teachings that client and server computers have the same architecture.” Pet., 40. Again, Petitioner identifies no evidence supporting its conclusory assertion that having the same architecture inherently discloses executing

the same view-generating logic on the client and the server. Indeed, the sole portion of Roy cited by Petitioner merely refers to a simple diagram of basic computer hardware components—“CPU”; “MEMORY”; “MASS STORAGE”; “MODEM”; “I/O” components—and does not address why it allegedly would have been inherent to use the same view-generating logic on the server and the client. Ex. 2031, ¶¶ 107–08.



Ex. 1004 at Fig. 2. *See also* Pet., 40 (citing Ex.1004, Fig. 2, 3:57–59).

Petitioner’s reliance on Dr. Bennett is similarly unavailing because Dr. Bennett’s testimony simply parrots Petitioner’s conclusory statements and cites only the same unavailing disclosure. *See* Pet., 40 (citing Ex.1003¶154–155; Ex.1004, Fig. 2, 3:57–59), 45 (citing Ex.1003, ¶¶154–155, 167); Ex. 1003, ¶¶ 154–155 (citing Roy at Figure 2 and 3:57-59 (Ex. 1004)), 167. In addition, Dr. Bennett’s assertion that “code reuse” is “routinely taught,” again, cites no supporting evidence. *See* Ex. 1003, ¶ 155. Accordingly, Dr. Bennett’s testimony is entitled to little weight, if any. *See Xerox* at 15–17; *Apple*, 28 F.4th at 262–63.

For the reasons above, Petitioner has failed to demonstrate that Roy discloses that the alleged “view-generating logic” on the client is the same as the alleged “view-generating logic” on the server, or some subset thereof, as required by Limitation (d). Ex. 2031, ¶ 109.

b. Petitioner Fails To Demonstrate That Roy Discloses Limitation (d)’s Requirement That The Alleged “Controller Logic” Executed On The Client Is The Same As The Alleged “Controller Logic” Executed On The Server, Or A Subset Thereof

As discussed above, Petitioner and IBM agree that Limitation (d) requires that the “‘controller logic’ executed on client *must be the same* ... ‘controller logic’ executed on the server, or some subset thereof.” Pet., 12 (emphasis added). Petitioner fails to demonstrate that Roy discloses this limitation under the agreed-upon construction for Limitation (d). Ex. 2031, ¶¶ 83–84, 112.

Petitioner asserts that Roy’s “event server” software teaches Limitation (d)’s “controller logic” element (i.e., “program code ...”). See Pet., 43 (“[T]he client and server ‘execute the ... *controller logic*’ (e.g., *event server 65 processing events 63* and *event server 67 processing events 64*).”). Specifically, Petitioner purports to map the “client ... ‘execut[ing] the ... controller logic” element to Roy’s “event server 65” in Player A’s computer system 60 (marked in the figure below in blue), and the “server ‘execut[ing] the ... controller logic” element to Roy’s “event server 67” in Player B’s computer system 62 (marked in the figure below in green):

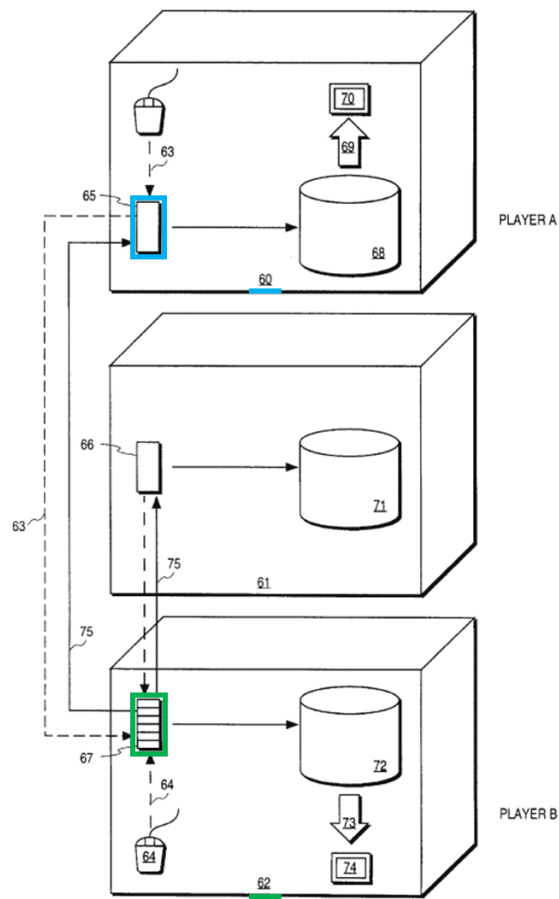


FIG. 6

Ex. 1004, Fig. 6 (annotations added); Pet., 43. *See also* Pet., 16–17 (“Roy likewise describes: ... (3) **controller logic** in client-side generated events 63 and **event server 65** and server-side generated events 64 and **event server 67**.”) (emphases added, color in original), 36, 40–41. Accordingly, under the agreed-upon construction, Petitioner must show that the **event server 65** is either the same as the **event server 67**, or a subset thereof. Ex. 2031, ¶¶ 110–11.

Petitioner again purports to rely on the doctrine of inherency to suggest that the **event server 65** is either the same as the **event server 67**, or a subset thereof. Pet.,

41. Petitioner alleges that “[a] POSITA would understand that the client-side and server-side ... controller logic are the same, as discussed above,” again presumably referring to Petitioner’s argument with respect to “Limitation (b).” Pet., 43. And with respect to Limitation (b), Petitioner alleges that the purported controller logic—i.e., the event servers—would “*inherently be the same* on each computer (and therefore the same whether on the server or client) for efficiency, economic, and common sense reasons.” Pet., 41 (emphasis added). Petitioner also alleges that “the same controller logic can be executed on any computer” because “every participating computer has an event server,” and that “[a] POSITA would understand that each computer processes ‘purely local events’ in the same way, such as local events for setting display options.” Pet., 41. Ex. 2031, ¶¶ 113–14.

Dr. Bennett presented this same argument in the *Zynga* Litigation, and the jury and the district court rejected it in upholding the validity of the ’719 Patent. *See, e.g.*, Ex. 2007, ¶ 533 (“For example, it would be obvious to a POSITA that each computer would process ‘purely local events’ in the same way, such as local events for setting display options.”). *See also* Paper 8 at 26–33, 36–38. Petitioner’s inherency argument fails for substantially similar reasons as with respect to the “view-generating logic” element discussed in Section VI.A.2.a.

First, Petitioner again ignores well-established black letter law that the inherent disclosure requires that the prior art “*must necessarily* include the unstated

limitation,” *Therasense*, 593 F.3d at 1332, and does not even attempt to show that Roy “must *necessarily*” teach that the alleged control logic in the client device—*event server 65*—is the same or a subset of the alleged control logic in the server—*event server 67*. See Pet., 41, 43. For example, Petitioner asserts that “every participating computer has an event server ... [t]hus, the same controller logic can be executed on any computer.” Pet., 41. But even assuming that were the case—and it is not—that still falls short of showing that the alleged client device 60 and the alleged server 62 “*must necessarily*” use the same logic. Ex. 2031, ¶¶ 114–16.

Second, Petitioner cannot demonstrate that Roy inherently discloses using *event server 65* that is either the same or a subset of the *event server 67*. Petitioner asserts that “POSITA would understand that the controller logic would inherently be the same on each computer” supposedly because “each computer processes ‘purely local events’ in the same way, such as local events for setting display options.” Pet., 41. Petitioner appears to assert that the alleged client device 60 and the alleged server 62 would execute the same controller logic because they would process the same “local events” in the same way. Ex. 2031, ¶¶ 117–18.

Petitioner’s assertions are incorrect. For example, as explained above in Section VI.A.2.a, Roy discloses embodiments where players will have *different* roles with drastically *different* methods of interaction with the game world. See, e.g., Ex. 1004 at 3:24–35 (describing an exemplary embodiment where players have

either an “offense” role or a “defense” role, with different corresponding controls), 3:35–47 (describing an exemplary embodiment where players have either a “superhero” role or a “peripheral” role, with different corresponding controls). Roy’s invention is premised on the understanding that players will always have *different* roles and game interactions that will require different levels of latency. See, e.g., Ex. 1004 at 3:18–22 (“In particular, the present invention is suited to implementing a game ... in which one particular player of the game has a role with a latency requirement that is significantly more restrictive than the other players’ roles.”), 5:65–6:14. Thus, *different* computer systems would process *different* events and would execute *different* logic for processing those events as a result. Ex. 2031, ¶¶ 118–20.

Moreover, both Petitioner and its expert, Dr. Bennett, appear to concede that the *event server 65* would *not necessarily* be either the same or a subset of the *event server 67*. For example, in arguing that “the controller logic would inherently be the same on each computer,” both Petitioner and Dr. Bennett assert that “POSITA would, *whenever technically feasible*, seek to avoid the expense and inefficiency of developing different logic and program code.” Ex. 1003, ¶ 158 (emphasis added). See also Pet., 41 (citing Ex.1003¶¶158), 45. However, that assertion—which mirrors Petitioner’s and its expert’s allegation with respect to executing the same view-generating logic—itself concedes that there would be circumstances in which

a POSITA would not execute the same controller logic on both the client and the server, including in circumstances where it is not “technically feasible” to do so. *See* Section VI.A.2.a. Accordingly, even the Petitioner and its expert recognize that it is *not necessary* for the controller logic to be the same in the client device and the server. Notably, neither Petitioner nor its expert argue, much less show, that it would *necessarily* be technically feasible in Roy’s disclosed system to execute the same controller logic. Pet., 41, 43, 45. Ex. 2031, ¶¶ 121–24.

Third, Petitioner’s inherency argument is legally deficient because, like Petitioner’s argument with respect to “view-generating logic,” it is not supported by any evidence. For example, Petitioner argues that “every participating computer has an event server” and incorrectly asserts that “[t]hus, the same controller logic can be executed on any computer.” Pet., 41. However, Petitioner cites no support for the latter conclusory allegation that simply having an event server on one computer means that this computer will now execute “the same controller logic” as other computers with different event servers. As a further example, Petitioner claims that “each computer processes ‘purely local events’ in the same way, such as local events for setting display options.” Pet., 41. But again, Petitioner cites no supporting evidence for this assertion. *See* Pet., 41.

Once again, Petitioner’s reliance on Dr. Bennett is misplaced. Dr. Bennett’s testimony continues to simply parrot Petitioner’s conclusory statements and cites the

same unresponsive portions of Roy. *See* Pet., 41 (citing Ex.1003¶158), 45 (citing Ex.1003¶154–155, 167); Ex. 1003, ¶¶ 154–155 (citing Roy at Figure 2 and 3:57-59 (Ex. 1004)), 158 (citing Roy; 5:40-43 (Ex. 1004)), 167. *See also* Section VI.A.2.a. Accordingly, Dr. Bennett’s testimony is again entitled to little weight. *See Xerox* at 15–17; *Apple*, 28 F.4th at 262–63.

For the reasons above, Petitioner has failed to demonstrate that Roy discloses that the “controller logic” Petitioner alleges is executed on the client must be the same as the “controller logic” Petitioner alleges is executed on the server, or some subset thereof, as required by Limitation (d). Ex. 2031, ¶ 125.

B. GROUND 2: Petitioner Does Not Establish A Reasonable Likelihood Of Prevailing Against Any Claim Under Ground 2 Because Roy Fails To Disclose Or Render Obvious All Limitations of Challenged Claims.

In Ground 2, Petitioner contends that all independent Claims 1, 13, 25, and 26 are rendered obvious by Roy. Pet., 3, 44. Petitioner’s substantive analysis for Ground 2 is limited to addressing elements of Limitation (d). *See* Pet., 45 (alleging that “Roy’s ‘view-generating logic’ and ‘controller logic’ are the same”). As explained below, Ground 2 fails for at least two reasons. ***First***, Petitioner has failed to demonstrate that the client computer system disclosed by Roy is a “server,” as claimed in Limitation (a). ***Second***, Petitioner has failed to demonstrate that Roy

renders obvious having the client execute the same “view-generating logic” and “controller logic” as the server, or some subset thereof. Ex. 2031, ¶¶ 126–27.

1. Petitioner Fails To Demonstrate That Roy Discloses That Player B’s Computer System 62 Is A “Server,” As Claimed In Limitation (a)

For the majority of the ’719 claim limitations—including Limitation (a)—Petitioner simply refers back to “the reasons stated above with respect to Ground 1” and relies solely on the alleged presence of anticipation to support its obviousness allegations. Pet., 44 n.17 (asserting without further discussion that “[i]t is well settled that a disclosure that anticipates under § 102 also renders the claim invalid under § 103”). For the reasons discussed above in Section VI.A.1, Petitioner fails to establish Roy discloses Limitation (a). Ex. 2031, ¶¶ 128–30. Indeed, as explained above, Dr. Bennett presented this same argument in the *Zynga* Litigation, and the jury and the district court rejected it in upholding the validity of the ’719 Patent. *See, e.g.*, Ex. 2007, ¶ 508. *See also* Section VI.A.1; Paper 8 at 26–33, 36–38.

2. Roy Does Not Render Obvious “Wherein The Client And The Server Both Locally Maintain At Least A Portion Of The Model And Execute The View-Generating And Controller Logic Associated Therewith” As Required By Limitation (d)

With respect to Limitation (d), the only difference between Ground 1 and Ground 2 is that, in Ground 1, Petitioner asserts that Roy *inherently* discloses Limitation (d) (“wherein the client and the server both locally maintain at least a

portion of the model and execute the view-generating and controller logic associated therewith”), whereas, in Ground 2, Petitioner asserts that Roy *renders obvious* Limitation (d). *See* Pet., 40–41, 43, 45. As explained below, Petitioner’s obviousness arguments with respect to Limitation (d) are facially deficient. Ex. 2031, ¶¶ 131–32.

a. Petitioner Fails To Demonstrate That Roy Discloses Or Renders Obvious Limitation (d)’s Requirement That The Alleged “View-Generating Logic” And “Controller Logic” Is The Same Between The Alleged Client System and the Alleged Server System

Petitioner incorrectly asserts—in a single sentence, no less—that “[a] POSITA would *inherently understand* that Roy’s ‘view-generating logic’ and ‘controller logic’ are the same whether on the master/server or slave/client system,” and cites largely to the same conclusory bases Petitioner advanced for Ground 1. Pet., 45 (purporting to assert that “any POSITA would seek to employ code reuse and *avoid expense and inefficiency* of developing different logic and program code to perform the same task when technically feasible, as a matter of *common sense* and good engineering practice”) (emphases added); Pet., 40 (“[A] POSITA would understand that Roy’s view-generating logic would inherently be the same on each computer ... for *efficiency, economic, and common sense* reasons.”) (emphases added). Petitioner’s inherency argument fails for the same reasons discussed above

in Sections VI.A.2.³ Ex. 2031, ¶¶ 133–35. Indeed, as explained above, Dr. Bennett presented these same arguments in the *Zynga* Litigation, and the jury and the district court rejected it in upholding the validity of the ’719 Patent. *See, e.g.*, Ex. 2007, ¶¶ 528, 533. *See also* Section VI.A.2; Paper 8 at 26–33, 36–38.

To the extent that Petitioner purports to assert that it would have been obvious to modify Roy such that the alleged “view-generating logic” on the client is “the same” as the alleged “view-generating logic” on the server, or some subset thereof, Petitioner’s assertion fails for at least the following reasons. Ex. 2031, ¶ 136.

First, the Petition fails to articulate that a POSITA would have modified Roy to use the same “view-generating logic” and “controller logic” in the server and the client. *See* Pet., 45. Petitioner’s failure to articulate any reasons for the modification renders its obviousness argument fatally deficient. *See Intelligent Bio-Sys., Inc. v. Illumina Cambridge Ltd.*, 821 F.3d 1359, 1367–68 (Fed. Cir. 2016) (“It [is Petitioner’s] burden to demonstrate both ‘that a skilled artisan would have been motivated to combine the teachings of the prior art references to achieve the claimed

³ In Ground 2, Petitioner addresses both the “view-generating logic” element and the “controller logic” element using the same argument. *See* Pet., 45. Accordingly, the deficiencies in Petitioner’s Ground 2 argument applies equally to both the “view-generating logic” element and the “controller logic” element.

invention, and that the skilled artisan would have had a reasonable expectation of success in doing so.”) (quoting *Kinetic Concepts, Inc. v. Smith & Nephew, Inc.*, 688 F.3d 1342, 1360 (Fed. Cir. 2012)). Ex. 2031, ¶ 137.

Second, to the extent Petitioner’s inherency allegations purport to be the bases for alleged obviousness, those allegations are deficient because they are conclusory and unsupported. For example, Petitioner asserts that “any POSITA would seek to employ code reuse and avoid the expense and inefficiency of developing different logic and program code to perform the same task when technically feasible, as a matter of common sense and good engineering practice.” Pet., 45. But the only purported “evidence” in support of Petitioner’s assertion is the declaration of its own expert, and that declaration in turn simply parrots Petitioner’s conclusory allegation without any supporting evidence. See Pet., 45 (citing Ex.1003, ¶¶154–155, 167). That is legally deficient. See *In re Magnum Oil Tools Int’l, Ltd.*, 829 F.3d 1364, 1380 (Fed. Cir. 2016) (“To satisfy its burden of proving obviousness, a petitioner cannot employ mere conclusory statements.”); *Xerox* at 15–16; *Apple*, 28 F.4th at 262–63. See also *Arendi S.A.R.L. v. Apple Inc.*, 832 F.3d 1355, 1362 (Fed. Cir. 2016) (“‘[C]ommon sense’—whether to supply a motivation to combine or a missing limitation—cannot be used as a wholesale substitute for reasoned analysis and evidentiary support, especially when dealing with a limitation missing from the prior art references specified.”); *iRobot Corp. v. SharkNinja Operating LLC*, Case

No. 2023-1398, 2024 WL 3687981, at *10 (Fed. Cir. Aug. 7, 2024). Ex. 2031, ¶ 138.

Accordingly, Petitioner has failed to demonstrate that Roy renders obvious that the alleged “view-generating logic” and “controller logic” on the server and the client are the same, as required by Limitation (d). Ex. 2031, ¶ 139.

C. GROUNDS 3–4: Petitioner Does Not Establish A Reasonable Likelihood Of Prevailing Against Any Claim Under Ground 3 Or Ground 4

Grounds 3 and 4 are solely directed towards dependent claims, which incorporate by reference all the limitations of the underlying independent claims 1 and 13. *See* Pet., 3; Ex. 1004 at Cls. 2–12, 14–24. Petitioner does not rely on the additional references identified in Grounds 3 and 4—Fennell and O’Reilly—to demonstrate any of the limitations of independent claims 1 and 13. *See* Pet., 46–77 (addressing only dependent claim limitations). Accordingly, all deficiencies identified with respect to Grounds 1 and 2 above apply equally to Grounds 3 and 4, and Petitioner has failed to demonstrate that: (1) Ground 3 renders obvious dependent claims 2, 4–12, 14, 16–24, and (2) Ground 4 renders obvious dependent claims 3 and 15. Ex. 2031, ¶¶ 140–42.

VII. CONCLUSION

For the reasons set forth above, Patent Owner respectfully requests that the Board deny institution of this *inter partes* review.

Case IPR2025-00564
U.S. Patent No. 7,702,719

Dated: June 11, 2025

Respectfully submitted,

/Taeg Sang Cho/

Taeg Sang Cho

(Reg. No. 69,618)

tcho@desmaraisllp.com

DESMARAIS LLP

230 Park Ave

New York, NY 10169

212-808-1060 (telephone)

212-351-3401 (facsimile)

*Lead Counsel for Patent Owner
International Business Machines Corp.*

Case IPR2025-00564
U.S. Patent No. 7,702,719

CERTIFICATION UNDER 37 C.F.R. § 42.24(d)

I hereby certify that this Patent Owner's Preliminary Response, excluding the portions exempted under 37 C.F.R. § 42.24(a), has 9,584 words as counted by the word-processing system used to prepare this document, in compliance with 37 C.F.R. §42.24(d).

Dated: June 11, 2025

Respectfully submitted,

/Taeg Sang Cho/

Taeg Sang Cho

(Reg. No. 69,618)

tcho@desmaraisllp.com

DESMARAIS LLP

230 Park Ave

New York, NY 10169

212-808-1060 (telephone)

212-351-3401 (facsimile)

*Lead Counsel for Patent Owner
International Business Machines Corp.*

CERTIFICATE OF SERVICE

Pursuant to 37 C.F.R. § 42.6(e), the undersigned certifies that a complete copy of the foregoing Patent Owner's Preliminary Response and corresponding exhibits was caused to be served on counsel of record for Petitioner by filing this document through PTACTS and by sending this document via electronic mail to the following addresses:

Charley F. Brown (Reg. No. 52,658)
browncf@ballardspahr.com
Ballard Spahr LLP
999 Peachtree Street, NE, Suite 1600
Atlanta, GA 30309

Brian W. LaCorte
lacorteb@ballardspahr.com
Ballard Spahr LLP
1 East Washington St., Suite 2300
Phoenix, AZ 85004-2555

Samhitha Medatia (Reg. No. 70,996)
medatias@ballardspahr.com
Ballard Spahr LLP
999 Peachtree Street, Suite 1000
Atlanta, GA 30309-3915

Drew Hensley (Reg. No. 79,305)
hensleyd@ballardspahr.com
Ballard Spahr LLP
1 East Washington Street, Suite 2300
Phoenix, AZ 85004

BallardCarvana719IPR@ballardspahr.com

Dated: June 11, 2025

Respectfully submitted,

/Taeg Sang Cho/
Taeg Sang Cho
(Reg. No. 69,618)
tcho@desmaraisllp.com
DESMARAIS LLP
Lead Counsel for Patent Owner
International Business Machines Corp.