

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

DRAFTKINGS INC.
Petitioner

v.

WINVIEW IP HOLDINGS, LLC
Patent Owner

IPR2026-00140
U.S. Patent No. 11,451,883

DECLARATION OF SANDEEP CHATTERJEE, PH.D.
U.S. PATENT NO. 11,451,883
CHALLENGING CLAIMS 1, 2, 5-9, 17-23, 26-30,
38-40, 81-84, 87-91, AND 99-101

TABLE OF CONTENTS

	Page
I. INTRODUCTION	1
II. BACKGROUND AND QUALIFICATIONS	1
III. LEGAL PRINCIPLES.....	5
A. Anticipation	5
B. Obviousness.....	6
C. Claim Construction.....	8
IV. MATERIALS CONSIDERED	9
V. SUMMARY OF OPINION	12
VI. TECHNOLOGY BACKGROUND.....	12
A. Personal Computing	12
B. Mobile Devices.....	14
C. Mobile Gaming.....	16
D. Downloading Pieces of Files.....	20
VII. OVERVIEW OF THE '883 PATENT	21
A. Alleged Problem.....	21
B. Alleged Invention	23
C. Prosecution History	24
VIII. LEVEL OF ORDINARY SKILL	27
IX. OVERVIEW OF THE PRIOR ART	28
A. Montagna.....	28
1. Montagna uses similar architecture to '883 patent	29
2. Montagna recognizes same problem and solution as '883 patent	32
B. Lewin.....	34
C. Zilliacus	37

X.	SPECIFIC GROUNDS FOR UNPATENTABILITY	39
A.	Ground 1: Claims 20-23, 26-30, 38, 1, 2, 5-9, 17, 81-84, and 87-91 are Obvious Over Montagna Alone or in View of Lewin	39
1.	Independent Claim 20	40
2.	Dependent Claim 21	97
3.	Dependent Claim 22	98
4.	Dependent Claim 23	100
5.	Dependent Claim 26	101
6.	Dependent Claim 27	104
7.	Dependent Claim 28	105
8.	Dependent Claim 29	107
9.	Dependent Claim 30	108
10.	Dependent Claim 38	110
11.	Independent Claim 1	111
12.	Dependent Claim 2.....	126
13.	Dependent Claim 5.....	126
14.	Independent Claim 81	127
15.	Dependent Claims 6-9, 17, 82-84, 87-91, 99.....	130
B.	Ground 2: Claims 28, 7, and 89 are Obvious Over Montagna Alone or in View of Lewin, and Further in View of Mathews	131
1.	Dependent Claim 28	131
2.	Dependent Claims 7 and 89	135
C.	Ground 3: Claims 39, 40, 18, 19, 100, and 101 are Obvious Over Montagna Alone or in View of Lewin, and Further in View of Zilliacus	135
1.	Dependent Claim 39	136
2.	Dependent Claim 40	142
3.	Dependent Claims 18, 19, 100, and 101	144
XI.	CONCLUSION.....	144
XII.	AVAILABILITY FOR CROSS-EXAMINATION	144
XIII.	RIGHT TO SUPPLEMENT.....	145
XIV.	JURAT	145

I, Sandeep Chatterjee, Ph.D., declare as follows:

I. INTRODUCTION

1. I have been retained by DraftKings Inc. to offer opinions regarding U.S. Patent No. 11,451,883 (the “’883 patent”) (Exhibit 1001). I have been asked to consider whether certain prior art references disclose or suggest the features recited in claims 1, 2, 5-9, 17-23, 26-30, 38-40, 81-84, 87-91, and 99-101 (the “challenged claims”) of the ’883 patent. My opinions are set forth below.

II. BACKGROUND AND QUALIFICATIONS

2. My qualifications are stated more fully in my *curriculum vitae*, which is attached in Appendix A. Below is a summary of my education, work experience, and other qualifications.

3. I am the Chief Executive Officer of Experantis LLC (“Experantis”), a technology consulting company. Previously, I was the Co-founder, Executive Vice President and Chief Technology Officer of SourceTrace Systems, Inc., a technology and services company enabling the delivery of secure remote electronic services over landline and wireless telecommunications networks.

4. I received my Bachelor’s degree in Electrical Engineering and Computer Science from the University of California, Berkeley in 1995. I received my Master’s degree in Computer Science from the Massachusetts Institute of Technology (MIT) in 1997, and my Doctorate in Computer Science from MIT in

2001. I received a certificate of completion for an executive education program on global leadership from Harvard University in 2011.

5. My doctoral dissertation at MIT, entitled “Composable System Resources for Networked Systems,” involving networked and distributed computer systems and architectures, was selected as one of the top inventions in the history of MIT’s Laboratory for Computer Science. This invention is showcased in a time capsule at the Museum of Science in Boston, Massachusetts. Other recipients of this honor include Bill Gates, the founder of Microsoft, and Tim Berners-Lee, the inventor of the World Wide Web.

6. In 2011, I was named a Young Global Leader. This honor, bestowed each year by the World Economic Forum, recognizes and acknowledges the top leaders—all below the age of 40—from around the world for their professional accomplishments, commitment to society, and potential to contribute to shaping the future of the world. In 2016, I was appointed to the World Economic Forum’s expert network as an expert in technology and innovation.

7. From 1997, I was the Entrepreneur-in-Residence at FidelityCAPITAL, the venture capital arm of Fidelity Investments. In 1999, I founded and served as President and Chief Technology Officer (CTO) of Satora Networks, which developed tools and technologies for building appliances and services for the Internet using wireless and other technologies to extend it beyond the desktop.

8. In 2001, I joined Bluestone Software's Mobile Middleware Labs as a Senior Engineer developing applications and systems infrastructure for enterprise Java/J2EE, Web services, and enterprise mobile solutions. After the completion of Hewlett-Packard's (HP) acquisition of Bluestone, I became a Senior Member of the Technical Staff at HP's Middleware Division. I was responsible for architecting and developing the company's next-generation Web services platform for enterprise as well as mobile environments, known as the Web Services Mediator.

9. I was part of the Expert Group that developed the JSR-00172 J2ME (Java 2 Platform, Micro Edition) Web Services Specification, the worldwide industry standard for mobile Web services. I am the co-author, with James Webber, of the book "Developing Enterprise Web Services: An Architect's Guide" (published by Prentice-Hall in 2004). This book has been adopted by over 100 universities and colleges around the world, and has been translated or reprinted in a number of countries around the world.

10. I have significant experience in developing complex computing systems. For example, through a contract between HP and the United States Agency for International Development (USAID), I architected and led the development of one of the first mobile banking solutions. This system enabled customers to use their mobile phones and other wireless handsets to connect with the core banking systems of banks and other financial institutions, and perform transactions without having to

travel to bank branches. This system supported many banking transactions, including deposits, withdrawals, loan applications, loan disbursements and loan repayments.

11. Later, after SourceTrace Systems' acquisition of this technology, I led the expansion of this solution into multiple countries and into multiple industries. Banks and other financial services companies utilized this technology to make their tellers more efficient, to provide self-service ATMs within branches, and to provide remote access to banking services. Additionally, through our licensing agreement with Telefonica, one of the largest cellular and telecommunications companies in the world, this solution was deployed in various other industries, including logistics and asset management and customer relationship management. Bloomberg Television selected and featured this technology and the company I co-founded to commercialize this technology on Bloomberg TV's "Bloomberg Innovators" program.

12. I have been a retained expert witness for various disputes that involved significant technology issues, and I have been qualified as a technology expert by U.S. District and State Courts, including in California, Delaware, Florida and Texas in technology areas that are relevant to this case, including but not limited to: computer software systems, distributed computing systems, and mobile and wireless systems. I have previously testified through declaration or expert report, at

deposition, and at trial in numerous intellectual property and commercial litigation matters, including for patent litigation, copyright and trade secret misappropriation litigation, and contract dispute cases. I have submitted more than one hundred and eighty expert declarations and expert reports, testified at deposition more than one hundred times, and testified at trial or at hearings at least eighteen times. I have been identified as one of the top 1000 patent professionals in the world, and I am listed in the IAM Patent 1000, which identifies the world's leading patent litigation and prosecution attorneys, as well as damages and technology expert witnesses.

13. Experantis is being compensated for my time on this matter at my standard hourly rate of \$895 per hour and reimbursed for any expenses that I incur related to my work in this matter. Neither Experantis nor I have any financial interest in the outcome of this matter, and Experantis will be paid for my time regardless of the outcome of this matter.

III. LEGAL PRINCIPLES

14. I am not an attorney. For purposes of this declaration, I have been informed about and understand certain aspects of the law that are relevant to my analysis and opinions.

A. Anticipation

15. I have been informed and understand that a claim is invalid if it was anticipated or not novel. I also have been informed and understand that a claim is

anticipated if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference.

B. Obviousness

16. I have been informed and understand that a patent claim can be considered to have been obvious to a person of ordinary skill in the art (“POSITA”) at the time of the alleged invention. This means that, even if all the requirements of a claim are not found in a single prior art reference, the claim is not patentable if the differences between the subject matter in the prior art and the subject matter in the claim would have been obvious to a POSITA at the time of the alleged invention.

17. I have been informed and understand that a determination of whether a claim would have been obvious should be based upon several factors, including, among others:

- the level of ordinary skill in the art at the time the application was filed;
- the scope and content of the prior art; and
- what differences, if any, existed between the claimed invention and the prior art.

18. I have been informed and understand that the teachings of two or more references may be combined if such a combination would have been obvious to one having ordinary skill in the art. In determining whether a combination based on

multiple references would have been obvious, I have been informed that it is appropriate to consider, among other factors:

- whether the teachings of the prior art references disclose known concepts combined in familiar ways, and when combined, would yield predictable results;
- whether a POSITA could implement a predictable variation, and would see the benefit of doing so;
- whether the claimed elements represent one of a limited number of known design choices, and would have a reasonable expectation of success by those skilled in the art;
- whether a POSITA would have recognized a reason to combine known elements in the manner described in the claim;
- whether there is some teaching or suggestion in the prior art to make the modification or combination of elements claimed in the patent; and
- whether the claimed invention applies a known technique that had been used to improve a similar device or method in a similar way.

19. I have been informed and understand that a POSITA has ordinary creativity, and is not an automaton.

20. I have been informed and understand that in considering obviousness, it is important not to determine obviousness using the benefit of hindsight derived from the patent being considered.

C. Claim Construction

21. I have been informed and understand that “claim construction” is the process of determining a patent claim’s meaning. I have been informed that claim construction is a matter of law and that in this proceeding, claim construction will be determined by the Patent Trial and Appeal Board (“the Board”). I understand that, during *inter partes* review, claim terms must be given their plain and ordinary meaning as would have been understood by a POSITA at the time of the claimed invention, having taken into consideration the language of the claims, the specification, and the prosecution history of record. I also understand that the scope of claim terms may be limited by statements in the specification or prosecution history where the applicant clearly disavows or disclaims subject matter, acts as his/her own lexicographer, or otherwise indicates the meaning of claim terms through statements made by the applicant.

22. I understand that the Board need only construe the claims when necessary to resolve the underlying controversy, and I do not believe that the Board needs to do so here. Given the close correlation between the asserted prior art and the challenged claims of the ’883 patent, I believe that the Board need not construe

any terms of the challenged claims to resolve the underlying controversy, as any reasonable interpretation of those terms consistent with their plain and ordinary meaning reads on the prior art.

IV. MATERIALS CONSIDERED

23. In preparing this declaration, I have reviewed the claims, specification, and file history of the '883 patent.

24. I have reviewed and understand the references and exhibits cited in this declaration, including the following:

Exhibit	Description
1001	U.S. Patent No. 11,451,883
1003	U.S. Patent Application Publication No. 2004/0242322 to Montagna ("Montagna")
1004	U.S. Patent No. 7,465,231 to Lewin ("Lewin")
1006	U.S. Patent No. 6,832,230 to Zilliacus ("Zilliacus")
1007	U.S. Patent No. 6,853,973 to Mathews ("Mathews")
1010	Prosecution History of U.S. Patent No. 11,451,883
1011	U.S. Patent Application Publication No. 2003/0228866 ("Pezeshki")
1012	Microsoft Computer Dictionary, 5 th ed., 2002 (relevant sections)
1022	James W. Cortada, <i>How the IBM PC Won, Then Lost, the Personal Computer Market</i> , IEEE Spectrum (July 21, 2021), https://spectrum.ieee.org/how-the-ibm-pc-won-then-lost-the-personal-computer-market (last visited Nov. 12, 2025)

1023	<i>Timeline of Computer History: 1984</i> , Computer History Museum, https://www.computerhistory.org/timeline/1984/ (last visited Nov. 12, 2025)
1024	Eskandari Abdi, <i>From Dial-up to Fiber Optic: The Evolution of the Internet</i> , 14 J. Mass Commc'n Journalism 595 (2024), https://www.hilarispublisher.com/open-access/from-dialup-to-fiber-optic-the-evolution-of-the-internet.pdf (last visited Nov. 12, 2025)
1025	<i>A history of mobile phones</i> , BBC, https://www.bbc.co.uk/bitesize/articles/z62gjfr# (last visited Nov. 12, 2025)
1026	<i>Blackberry Connection</i> , Blackberry https://www.blackberry.com/news/connection/200201.shtml (last visited Nov. 12, 2025)
1027	Ivica Crnkovic, <i>Component-based Software Engineering – New Challenges in Software Development</i> , J. Computing and Info. Tech. 151 (2003), available at https://scispace.com/pdf/component-based-software-engineering-new-challenges-in-5bcdjbyh7r.pdf (last visited Nov. 12, 2025)
1028	Richard Baguley, <i>The Gadget We Miss: The Nokia 9000 Communicator</i> , Medium (Aug. 1, 2013), https://medium.com/people-gadgets/the-gadget-we-miss-thenokia-9000-communicator-ef8e8c7047ae (last visited Nov. 12, 2025)
1029	<i>Nokia 9000 Communicator</i> , Mobile Phone Museum, https://www.mobilephonemuseum.com/phone-detail/nokia-9000-communicator (last visited Nov. 12, 2025)
1030	Matt Shipman, <i>Tracing the History of Mobile Games</i> , NC State (Jan. 28, 2021), https://news.ncsu.edu/2021/01/tracing-the-history-of-mobile-games/ (last visited Nov. 12, 2025)
1031	Ayla Angelos, <i>Snake by Nokia</i> , It's Nice That (February 23, 2021), https://www.itsnicethat.com/features/taneli-armanto-the-history-of-snake-design-legacies-230221 (last visited Nov. 12, 2025)

1032	<i>All About the Snake Game: From Arcades to Mobiles and Beyond</i> , https://www.snakegame.net/about-snake (last visited Nov. 13, 2025)
1033	Tim Larimer, Time Magazine (Mar. 5, 2001), <i>Internet A La I-Mode</i> , available at https://content.time.com/time/subscriber/article/0,33009,999355,00.html (last visited Nov. 12, 2025)
1034	Bram Cohen, <i>Incentives Build Robustness in BitTorrent</i> (May 22, 2003), available at https://web.archive.org/web/20050128112626/http://bittorrent.com/bittorrentecon.pdf (last visited Nov. 21, 2025)
1035	Bitcomet, https://community.ccleaner.com/t/bitcomet-pc-not-shuting-down/736 (last visited Nov. 20, 2025)
1036	Paul Kryzhanowski, <i>Lectures on distributed systems: Client-server communication</i> , Rutgers University (1999), https://formations.telecom-bretagne.eu/syst/TPDIST/Additional-Reading/07-networking-client-server.pdf (last visited Nov. 12, 2025)

25. I have been informed and understand that the earliest possible priority date that the '883 patent can claim is June 20, 2005—the filing date of the provisional application to which the '883 patent claims priority. For purposes of my analysis herein, I have used this date as the relevant time period.¹

¹ I am not offering an opinion that the '883 patent should be entitled to this priority date. I have not been asked, and I have not performed an analysis to determine whether any of the claims of the '883 patent should be entitled to this priority date.

V. SUMMARY OF OPINION

26. In my opinion, claims 1, 2, 5-9, 17-23, 26-30, 38-40, 81-84, 87-91, and 99-101 of the '883 patent would have been obvious in light of the prior art.

27. The table below lists the reference and combination of references based on which claims 1, 2, 5-9, 17-23, 26-30, 38-40, 81-84, 87-91, and 99-101 of the '883 patent would have been obvious.

Ground	Basis	Claims	References
1	§ 103	1, 2, 5-9, 17, 20-23, 26-30, 38, 81-84, 87-91, 99	Montagna and Lewin
2	§ 103	7, 28, 89	Montagna, Lewin, and Mathews
3	§ 103	18, 19, 39, 40, 100, 101	Montagna, Lewin, and Zilliacus

VI. TECHNOLOGY BACKGROUND

A. Personal Computing

28. Personal computing quickly developed from the 1980s to the early 2000s. Throughout the development of personal computing devices, system designers have been concerned with managing memory in such devices and the amount of bandwidth used in transmitting information to such devices. In the 1980s, computers were first developed for household use. IBM's first personal computer was introduced in 1981. Exhibit 1022 (IBM). Apple launched the Macintosh in 1984. Exhibit 1023 (Computer History). With the release of "dial up" in the early

1990s, the Internet became accessible to home computers, which utilized a home's telephone line to deliver information. Exhibit 1024 (Dial Up History). However, signals over telephone lines were suboptimal for the transfer of information, and due to poor conditions of telephone lines and high noise, Internet speeds were often less than 100 kilobits per second.² *Id.* As a result, programs that transferred data over the Internet and other networks were focused on limiting the amount of information that needed to be transmitted at any given time. Thus, programmers of the time were adept at managing the amount of information sent to and from early Internet-connected computing devices.

29. Another concept present in the '883 patent that would have been well known to someone in working in the field, before the earliest possible priority date of the '883 patent, is component-based software. Component-based software engineering is a paradigm where parts of a software program can be reused from one application to another. This improves scalability, and the effective management of resources and complexity. *See generally*, Exhibit 1027 (Crnkovic). This design

² Modern Internet speed is measured in megabits per second (1000 times one kilobits per second). In 2025, broadband speed may be in the hundreds of megabits per second, making it over 1000 times faster than the speed of early dial-up.

paradigm is not new and had already been used for many years as of the filing of the '883 patent. *Id.*

B. Mobile Devices

30. Well before earliest possible priority date of the '883 patent, a POSITA would have been aware of the trend towards mobile computing. For example, IBM Simon, which was released in August 1994, was considered the first smartphone, because it had a touchscreen, could send faxes, and also had email and calendar functionality. Exhibit 1025 (IBM Simon).

31. In 1996, Nokia launched the 9000 Communicator, which had 8 MB of memory, a 33 MHz processor, and which included email, web browsing, fax, word processing, and spreadsheets in a single pocketable device. *See* Exhibit 1028 (Baguley); *see also* Exhibit 1029 (Mobile Phone Museum). The 9000 Communicator had two screens; one on the outside that offered similar functionality to other devices of the day, and the other on the inside that was a black and white LCD with a then-high resolution of 640 x 200 pixels. The 9000 Communicator could offer a graphical web browser on a mobile device. *See* Exhibit 1028 (Baguley).



Exhibit 1028 (Baguley), Nokia 9000 Communicator

32. The 9000 Communicator could connect to the Internet using a 9600 bits per second GSM modem, which worked with the digital GSM phone networks that were becoming commonplace at the time. It also could connect to a PC, allowing users to download and install software. *Nokia 9000 Communicator*, Mobile Phone Museum.

33. In 2001, the third generation (“3G”) of cellphones were released. 3G was able to deliver broadband access, with speeds of up to several megabit/s, to smartphones and laptop computers. Shortly thereafter, the first Blackberry phone, the Blackberry 5810, was released with wireless email and built-in applications. The Blackberry also included a browser for accessing the Internet. Exhibit 1026 (Blackberry). While mobile devices were developing and improving, the ways that they were used was evolving as well.

C. Mobile Gaming

34. By the 1990s, mobile phones had become an important platform for gaming. *See, e.g.*, Exhibit 1030 (Shipman). For example, “Snake,” released for the Nokia 6110 in 1997, is widely considered one of the biggest successes of the early mobile gaming era. *See* Exhibit 1031 (Angelos); *see also* Exhibit 1032 (All About the Snake Game).



Exhibit 1032 (All About the Snake Game), Snake on Nokia 6110

35. In the early 2000s in Japan, a service called I-mode became one of the earliest proofs of concept for downloadable content to mobile devices. I-mode boasted “40,000 sites designed specifically for cell phones, many of them truncated versions of regular Web portals,” including applications dedicated to sports, astrology, animation, navigating train routes, following the stock market, and games. Exhibit 1033 (Larimer). I-Mode caught the attention of U.S. markets, as well,

garnering news attention, e.g., partnerships with mobile giants like AT&T Wireless.
See id.

36. Thus, by the earliest possible priority date of the '883 patent, a POSITA would have been familiar with the use of mobile devices to interact with the Internet, as well as a variety of user applications, including games. *See, e.g.*, Exhibit 1006 (Zilliacus), 1:56-59 (“Downloading content to a mobile terminal is regularly performed. Oftentimes content is downloaded to a mobile terminal directly or indirectly utilizing the WWW.”), 1:13-19 (“Recent years have witnessed the emergence and proliferation of the Internet, the World Wide Web (“WWW”), mobile communication technologies and mobile computing devices. As technology advances, telecommunications and computing are converging. Convergence is producing devices that communicate over networks and integrate media, telephony and computing.”), 1:38-39 (“[C]ell phones are now able to run software applications and provide Internet access.”), 1:65-66 (“[E]xisting mobile terminals are capable of receiving content downloaded thereto.”); 2:49-51 (“Games are often available in a limited dataset where only the first level is available to the user.”). The '883 patent itself recognizes this. Exhibit 1001 ('883 patent), 1:62-63 (“In the United States alone there are over 170 million registered cell phones”), 2:12-14 (“the ability to play games and enjoy other activities corresponding to current events on cell phones exists”).

37. The '883 patent also recognizes that the servers, from which software and other applications are downloaded, were already widely used. *See* Exhibit 1001 ('883 patent), 2:33-40 (“Servers of the entertainment provider also have limited resources yet must be able to ensure the potentially millions of simultaneous users have the necessary and proper software resident on their cell phones in order to participate in scheduled interactive programs. Again, servers are continuously growing more powerful, quicker and more stable; however there is still potential to overload a server with excessive traffic.”). Other references that I reviewed make clear that different components that allow the download of gaming applications would also have been well known to a POSITA. For example, Montagna, Lewin, and Zilliacus all mention the use of servers and the Internet to download applications to user devices. *See, e.g.*, Exhibit 1003 (Montagna), [0031] (“The mentioned assembling process can include retrieving user interface components from a networked resource, for example, a Web server or Networked File Structure.”), [0055] (“The network can be, for example, any combination of: the Internet, a network that includes Wireless Application Protocol (‘WAP’) clients, and a network that includes non-browser based clients.”); Exhibit 1004 (Lewin), 17:4-16 (“Thus, a first GUI player 82 that is written to support a high-end computer system may produce a presentation that shows a user interface and host avatar as a full-screen, dynamic, three-dimensional image. While a second GUI player 82 that is written for

a lower-end computer system may produce a two-dimensional presentation of the interface and avatar. And a third GUI player 82 written for a handheld wireless device or mobile phone may limit the presentation to a text message. GUI players 82 can thus modify the presentation of the system 10 to meet the specific hardware requirements of a user device. Used this way, the GUI players 82 allow the content distribution system 10 to support a broad range of user devices.”), 10:46-49 (“Additional users may run the client application 15 on a set top box of a cable or satellite system, a personal digital assistant (PDA) device, cell phone or other digital electronic device.”); Exhibit 1006 (Zilliacus), 1:64-67 (“Downloading content to a mobile terminal is regularly performed. Oftentimes content is downloaded to a mobile terminal directly or indirectly utilizing the WWW.”).

38. A POSITA would have also recognized that, depending on the size of an application, there could be a sizeable amount of information being transmitted to the user’s device. *See, e.g.*, Exhibit 1004 (Lewin), 9:55-63 (“For example, games that were originally written for the Atari and Nintendo video console systems are typically less than a megabyte in size and require just a few seconds to download to the client application 15. In contrast, a medium-size game written for the Sony PlayStation console is about 100 megabytes in size, and a game written for a personal computer typically ranges between several hundred megabytes to several gigabytes of data.”). Thus, the need to reduce the amount of information being transferred,

given the limits to Internet connectivity and memory storage of the day, was also recognized. Exhibit 1003 (Montagna), [0014] (“Furthermore, resources from previous game sessions can be reused in a current game session and the time needed to download resources to a client terminal is thus reduced.”); Exhibit 1006 (Zilliacus), 3:1-4.

D. Downloading Pieces of Files

39. Downloading pieces of files was well known to a POSITA.

40. For example, the BitTorrent protocol starts when “a static file with the extension .torrent is put on an ordinary web server.” Exhibit 1034 (Cohen) at 2. The .torrent file contains information about an entire file: length, name, hashing information, and the URL of a tracker. *See id.* “Trackers are responsible for helping downloaders find each other.” *Id.*

41. Cohen explains that: “BitTorrent cuts files into pieces of fixed size, typically a quarter megabyte. Each downloader reports to all of its peers what pieces it has. To verify data integrity, the SHA1 hashes of all the pieces are included in the .torrent file, and peers don’t report that they have a piece until they’ve checked the hash.” *See id.*

42. To make a file available, a “downloader” that already has the complete file, known as a seed, must be started. *Id.* Peers continuously download pieces of the file from peers, but cannot download from peers to which they are not

connected. *See id.* At times, peers do not have any piece of the file or may choose to prevent download of a piece of the file. *See id.*

43. Torrents were used to download games. *See, e.g.,* Exhibit 1035 (Bitcomet).

44. Thus, by the earliest possible priority date of the '883 patent, a POSITA would have been well versed in the features described in and claimed by the '883 patent, including memory and bandwidth conservation, component and file transmission, as well as gaming.

VII. OVERVIEW OF THE '883 PATENT

A. Alleged Problem

45. The '883 patent is titled “[m]ethod of and system for managing client resources and assets for activities on computing devices.” Exhibit 1001 ('883 patent), Title. The '883 patent describes an “entertainment service on computing devices” that include “set-top boxes, personal computers, gaming consoles, laptop computers, cell phones, PDAs and combination devices such as cell phone/mp3 players.” *Id.*, 5:22-27. In particular, the '883 patent describes an entertainment service that allows users to play “TV-related activities” on their devices. *See id.*, 5:28-6:29.

46. The '883 patent contends that device memory and network bandwidth posed constraints on timely delivery of user-selected and personalized content. *See,*

e.g., Exhibit 1001 ('883 patent), Abstract, 2:15-24 (“Cellular networks and other networks are only able to handle a limited amount of traffic, so when millions of participants utilize the network at the same time for a regularly scheduled event, there are concerns of the network being overloaded and bogged down to a point where users do not receive necessary data in a timely manner. Furthermore, cell phones and other computing devices have their own limitations as well. Although cell phone memory is increasing as is typical with technology, cell phones still have a limited amount of memory.”).

47. The '883 patent further contends that these constraints caused issues with delivering an entertainment service that was compatible with a user device and that would arrive in time for a televised event that a user would like to play along with. Exhibit 1001 ('883 patent), 2:33-37 (“Servers of the entertainment provider also have limited resources yet must be able to ensure the potentially millions of simultaneous users have the necessary and proper software resident on their cell phones in order to participate in scheduled interactive programs.”). According to the '883 patent, these issues were only made worse when the activity that the user seeks to participate in is tied to a televised event. Exhibit 1001 ('883 patent), Abstract (“Furthermore, since time is of the essence for activities based on live or televised events, it is necessary to ensure a computing device is able to receive the necessary updates”).

B. Alleged Invention

48. To address these bandwidth and memory problems in interactive TV activity, the '883 patent describes a way to minimize a transfer to a user device, by sending only assets that are not already on the device, rather than everything that the user needs to participate in the activity. Exhibit 1001 ('883 patent), Abstract (“By providing only the needed components instead of entire data packages, the present invention is able to much more efficiently prepare users’ computing devices in time.”), 4:65-67 (“The server device sends only assets in the list of assets that are not in the list of client assets for each client device of the client devices.”). To implement this alleged solution, the '883 patent describes a “list of assets” that the device can compare with what is stored in its local memory to determine what assets it has and what additional assets it needs to request to complete the user’s selected game. Exhibit 1001 ('883 patent), 3:34-38 (“The method further comprises receiving a list of assets from the server, comparing the list of assets with a first set of assets on the computing device and receiving only a second set of assets within the list of assets that are not already resident on the computing device.”). The '883 patent contends that using such a list prevents excess transfers of data, because only the missing assets are transferred.

C. Prosecution History

49. The '883 patent issued from U.S. Patent Application No. 16/893,180, which was filed on June 4, 2020. It claims priority to Provisional Application No. 60/692,356, which was filed on June 20, 2005.

50. The Patent Office issued a non-final Office Action on April 5, 2021, rejecting all originally-filed claims as being anticipated by U.S. Patent Application No. 2003/0228866 to Pezeshki et al. Exhibit 1010 (File History of '883 patent) at 1220-27. Pezeshki describes a method of interacting with applications on a mobile Internet-connected computing device from application servers. Exhibit 1011 (Pezeshki), [0001] (“The present invention relates generally to mobile communication devices, and more particularly relates to the dynamic provision of applications to mobile communication devices via application servers.”), [0004] (“Mobile communication devices such as PDAs and mobile phones are simple computers with limited computational power and resources, such as storage, memory and clock frequency. Mobile communication devices can be equipped with client software to communicate with a remote server through a wireless network. Such devices are usually loaded with a few client applications either by the user, the manufacturer or the network operator. It is a common practice to distribute applications through the Internet. Some additional equipment such as a personal computer and special cables and connectors are needed to transfer the client

application to the mobile communications device.”). As the Patent Office noted, Pezeshki teaches that applications can be transmitted to a mobile device. Exhibit 1010 (File History) at 1223; Exhibit 1011 (Pezeshki), [0055] (“In the present invention, SIM 208 is provisioned with a set of additional applications to enable the enhanced functionality. These applications are typically loaded prior to distribution to an end-user, but can also be uploaded to SIM 208 via an over-the-air (OTA) system.”). The Patent Office also noted that Pezeshki teaches a method where a list of applications are displayed for the user, and the user is allowed to select one of the applications. Exhibit 1010 (File History) at 1223; Exhibit 1011 (Pezeshki), [0077] (“In the case where the commands are received in response to an initial request for a list of applications 316 hosted by application server 300, the commands can direct user interface 204 to display the list of applications 316 and allow a user to select one of the applications 316. Upon selection of an application 316 from the list of applications 316, the mobile device sends an initial message to the selected application 316. Where the selected application resides on a separate server, mobile device 200 initiates a communication with the separate server.”), [0031]. Finally, the Patent Office noted that Pezeshki updates the list of available applications when a new application becomes available. Exhibit 1010 (File History), at 1223; Exhibit 1011 (Pezeshki), [0066] (“When a new application 316D is to be made available to mobile device 200, new application 316D is first installed on application server 300.

The new application is then added to the list of applications available on application server 300 that is accessed through directory application 320. As a result, any subsequent responses by directory application 320 to requests for a list of applications will enable the user of mobile device 200 to select and access new application 316D.”).

51. To overcome the rejection based on Pezeshki, the Applicant amended the pending claims and argued that the assets are *necessary for executing* a single application and *directly affect the operation of that application*, and that only missing assets (second set of assets) are sent to the device. Exhibit 1010 (File History) at 1429 (“The independent Claims 67, 86, 107, 128, 147, 168, 189 have been amended by the above amendments to specify receiving a list of assets necessary for executing the application, to make it clear that the list of assets relates to a single application and only the missing assets (second set of assets) are sent to the device.”) (emphasis in original), 1588 (“The independent Claims 67, 86, 107, 128, 147 and 168 have been amended by the above amendments to specify wherein the assets necessary for executing the application *directly affect the operation of the application*, to make it clear that the assets relate to and affect a *single* application”) (emphasis in original). Applicant did not argue that mobile Internet-connected computing devices or applications were not present in the art. The Patent Office subsequently allowed claims 67-214, which issued as claims 1-146.

52. As I explain below, the prior art on which I rely to show the invalidity of the challenged claims shows that sending a second set of assets necessary for executing a single application was well known in the art.

VIII. LEVEL OF ORDINARY SKILL

53. I have been informed and understand that, for the purposes of this analysis, an assessment of the claims of the '883 patent should be undertaken from the perspective of a POSITA at the time of the alleged invention. I have been informed and understand that the earliest possible priority date for the alleged invention is June 20, 2005—the filing date of the provisional application to which the '883 patent claims priority. Even if the Patent Owner later alleges an earlier invention date, my opinion would not change as long as the references cited below remain prior art to the '883 patent.

54. In my opinion, a POSITA as of June 2005³ would have possessed at least the equivalent of a Bachelor's degree in computer science, computer engineering, or a related subject and two or more years of experience in the field of

³ Even if Patent Owner later alleges that the claims of the '883 patent are entitled to an earlier priority date due, for example, to an earlier conception of the claimed subject matter, that would not change my view of the level of ordinary skill in the art.

distributed systems, networking, and software design. My opinions would not change if a slightly different formulation was used. I believe that if an individual had less work experience but had a higher level of education, such as a Master's degree, they would still be considered a POSITA, and the same would be true of a lower level of education, but additional experience.

55. My opinions regarding the level of ordinary skill in the art are based on, among other things, my years of experience in the field of computer science and engineering, my understanding of the fundamental qualifications that would be relevant to an engineer or scientist tasked with investigating methods and systems in the relevant area, and my familiarity with the backgrounds of students, colleagues, and coworkers, both past and present.

56. At the relevant time (*e.g.*, June 2005), I had the qualifications and experience that were at least those of the hypothetical POSITA defined above. My analysis and opinions regarding the '883 patent have been based on the perspective of a POSITA as of June 2005.

IX. OVERVIEW OF THE PRIOR ART

A. Montagna

57. U.S. Patent Application Publication No. 2004/0242322 to Montagna ("Montagna") (Exhibit 1003) was published on December 2, 2004, at least six months before the earliest possible priority date of the '883 patent.

58. Montagna discloses a method of delivering a game interface to a user device. Montagna's system utilizes information about the user, including their location, to build a customized interface to play games, including casino games like poker. Exhibit 1003 (Montagna), [0006] ("In general, in one aspect, the invention provides a method for generating in real time a personalized gaming user interface definition. The method includes receiving a request for a user interface for a game"), [0093]; *see also id.*, [0004] ("Gaming applications can provide different types of games of chance, examples of which include, and are not limited to, craps, blackjack, roulette, baccarat, poker, keno, bingo, and progressive jackpots").

1. Montagna uses similar architecture to '883 patent

59. To deliver a customized game, Montagna and the '883 patent disclose the same system architecture. The '883 patent discloses an activity client (i.e., an application) that receives from a server the assets necessary to operate a game on a computing device (i.e., a user device). Exhibit 1001 ('883 patent), 2:65-3:5. This relationship is shown in Figure 5 below. The **computing device 500** is shown in pink, the **Activity Client 504** is shown in red, and the **server 512** is shown in green. Exhibit 1001 ('883 patent), 10:20-31 ("FIG. 5 illustrates a block diagram of an embodiment of the present invention. A computing device 500 such as a set-top box, personal computer, gaming console, laptop computer, cell phone, PDA or combination cell phone/mp3 device includes a memory 502 for storing data. As

described above, the memory 502 varies in size depending on the type and brand of computing device, where more recently produced computing devices have larger memories and are thus able to store more data. However, older computing devices only store a minimal amount of data. Within the memory an “Activity Client” 504 is stored for receiving activity-specific data 508.”), 10:46-51 (“The server 512 also includes a communications interface 524 for communicating with the client device 500, an asset module 526 for determining which assets the client device 500 needs and a distributing module 528 for sending the assets needed to the client device 500.”).

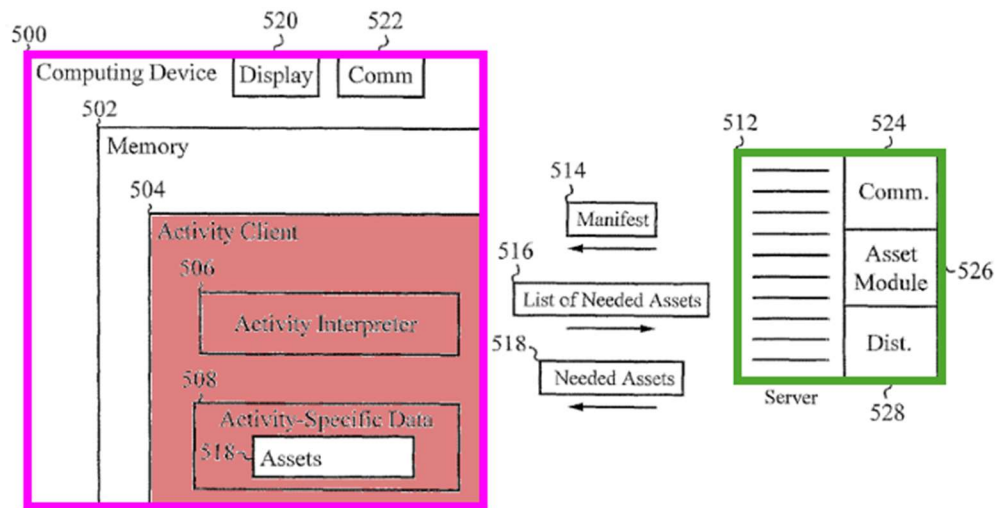


Fig. 5

Exhibit 1001 ('883 patent), Figure 5 (annotated)

60. Montagna discloses a similar architecture in its Figure 3, reproduced and annotated below. Montagna discloses a **game client 304** (i.e., an application) which resides on a **client terminal 316** (i.e., a user device), which communicates

with **server 314**. Exhibit 1003 (Montagna), [0049] (“FIG. 3 shows game platform 300 that implements the above-described system 100. Game platform 300 includes a game definition engine (‘GDE’) 302, a game client 304, a game manager 306, a Random Number/Outcome Generation application (‘RiNGO application’) 308, a database 310, and a write-only log 312. GDE 302, game manager 306, RiNGO application 308, database 310, and write-only log 312 reside on a server 314 while the game client 304 resides on a client terminal 316.”). Montagna’s system is distributed over a network and distributes a gaming application to the game client 304. *Id.*, [0055] (“As shown in FIG. 4, game platform 300 can be distributed over a network. The network can be, for example, any combination of: the Internet, a network that includes Wireless Application Protocol (‘WAP’) clients, and a network that includes non-browser based clients.”). The **game client 304** receives assets from the **server 314** in order for a game to be played. *Id.*, [0051] (“Game client 304 collects user input (such as a wager) and communicates it back to the server-side components of the game platform 300.”), [0061] (“Game client 304 loads game assets in the user’s preferred language and currency.”). I note that Montagna’s Figure 3 contains a typo: the game client is labelled as 306, rather than 304, which is how it is referred to elsewhere in the specification.

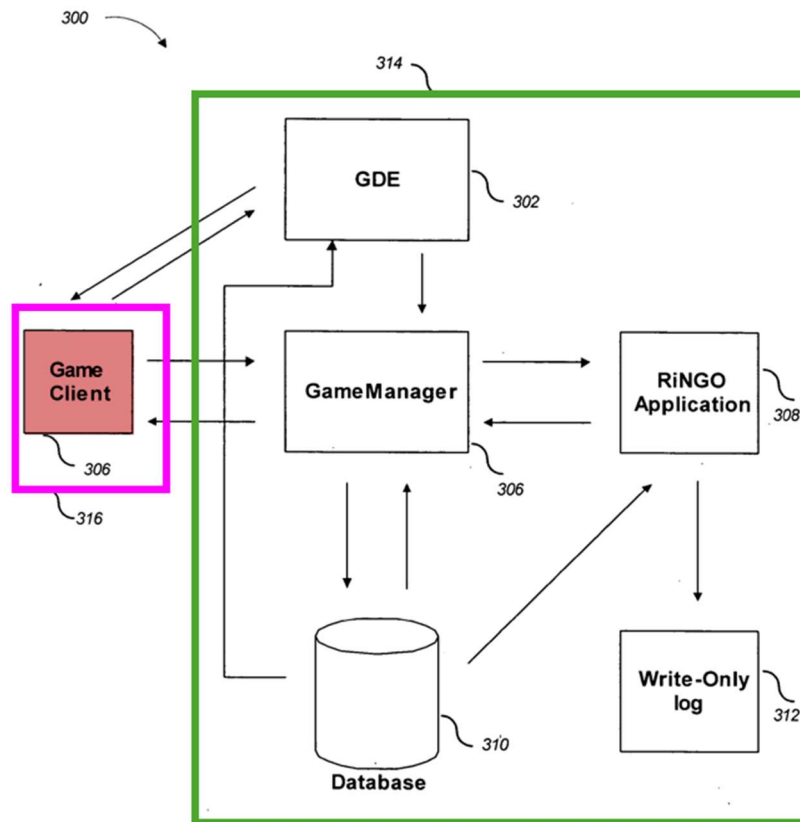


Exhibit 1003 (Montagna), Figure 3 (annotated)

2. Montagna recognizes same problem and solution as '883 patent

61. Montagna recognizes that excessive download time creates a negative experience for users. Exhibit 1003 (Montagna), [0014] (“Furthermore, resources from previous game sessions can be reused in a current game session and the time needed to download resources to a client terminal is thus reduced.”). Montagna describes the same solution as the '883 patent—by sending the user device only the assets it needs, and not the assets that it already has, network bandwidth and memory can be conserved. *Id.*, [0045] (“In the instant example, the UI definition indicates

that several user interface components are available at a certain location on a network accessible Web server.... [C]ertain resources can be available in local cached memory resulting from previous personalized user interfaces. UI interpreter 112 will not have to retrieve these resources.”), [0089] (“In another implementation, the Flash Game Loader first checks for locally cached resources before requesting a download of the resources from the gaming server. If a cached resource exists, it is retrieved from the cache.”), [0010] (“The system is configured to personalize user interfaces in real time. In response to user input, for example, a request to participate in on-line gaming, the system can generate a user interface that is personalized for the specific user and *for the current request with little latency.*”).⁴ Montagna also implements this solution in the same way as the ’883 patent—by building a list of assets that describes the needed assets for an activity, allowing the device to determine what game components it already has, and what components it will need to request from the server. *Id.*, [0036] (“Rules 106 can specify, for example, that given a particular user and a particular enterprise, the UI definition should include instructions that identify certain user interface resources such that the user interface will include components generated by the resources.”), [0042] (“In gaming, for example, the UI definition can include instructions for providing a complete game with out further

⁴ Emphasis is added throughout unless I note otherwise.

interaction with system 100.”), [0040] (“A user interface definition interpreter (‘UI interpreter’) 112 requests from definition engine 108 and interprets a UI definition to construct a user interface 114.”).

B. Lewin

62. U.S. Patent No. 7,465,231 to Lewin (“Lewin”) (Exhibit 1004) was filed on May 20, 2004—more than a year before the earliest possible priority date of the ’883 patent—and published on December 16, 2008.

63. Lewin describes methods to distribute entertainment content across a network. Exhibit 1004 (Lewin), 6:16-19 (“The present invention is described herein in the context of systems and methods that are configured to distribute entertainment content to clients over a computer network.”), 3:3-11 (“The present invention is directed to methods and systems of delivering entertainment content via a network. An embodiment of the delivery system uses a graphical user interface and host avatar to introduce users to and allow them to select from available content. With regard to the delivery of games, emulators execute game code that is written to run on various gaming platforms. A dynamic interface launches and manages emulators in a manner that is largely transparent to the user.”).

64. As shown in Figure 1 (reproduced below), Lewin discloses “several client applications 15 communicating with a host server 20, preferably composed of one or more logical servers deployed on one or more physical server, and one or

more asset servers 25 via a network 45.” Exhibit 1004 (Lewin), 6:28-31. Lewin describes that the “volume of entertainment data passed to the client applications 15 is typically sufficiently large that a broadband link is preferably used to handle the communication that occurs between the client applications 15 and the network 25.” *Id.*, 6:40-44. However, Lewin explains that “the volume and type of data transmitted to and from the client applications 15 can be tailored to the requirement of the device used to execute the client application 15.” *Id.*, 6:45-47.

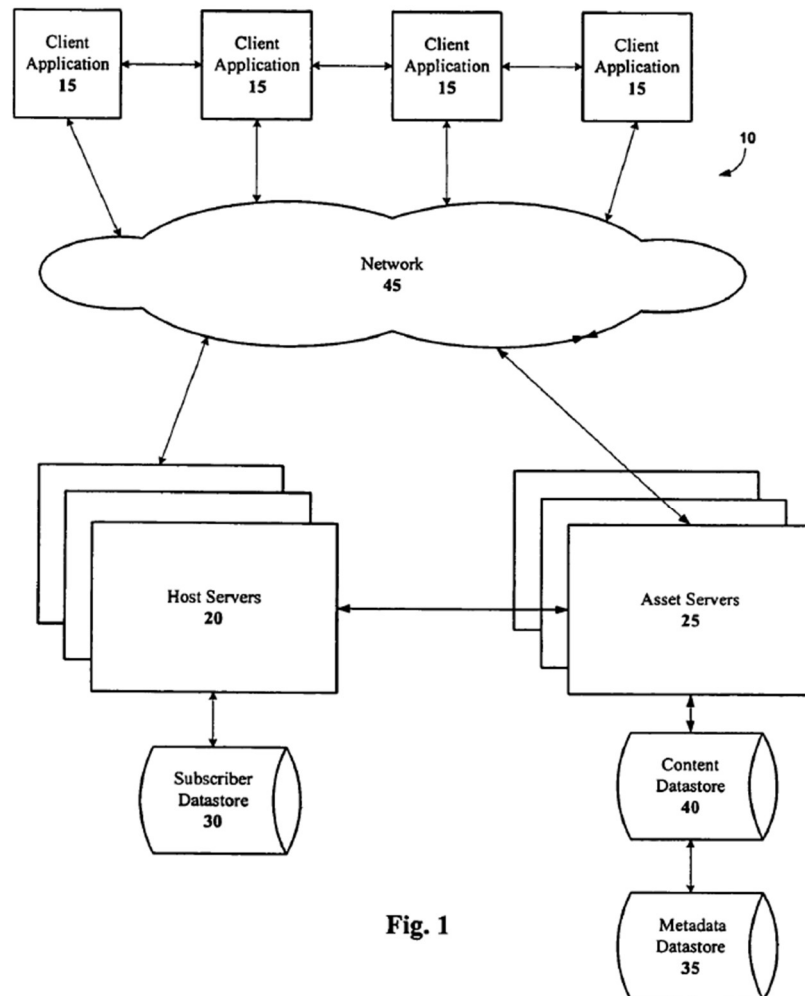


Fig. 1

Exhibit 1004 (Lewin), Figure 1

65. Lewin further describes a “client application” that downloads and executes complete game applications. Exhibit 1004 (Lewin), 3:26-36 (“Additional embodiments of the game delivery system describe using a client application residing on a computing device associated with the user that displays a listing of available games accepts a user request to play a game. In some described embodiments, the client application includes a plurality of emulators that are used to emulate the native platforms for which the plurality of games were originally intended to run. The emulators translate the native platform machine code blocks of the game code to functionally equivalent blocks of compiled instruction set code that run on the user device.”). The use of such a system allows the client application to execute an application that may not have initially been intended to run on the user’s device. *See id.*, 3:15-19 (“a game delivery system that allows a user to select from and receive delivery of at least one of a plurality of games via a network, the games include a first set of game code that is stored in a format to be run on a series of arcade game systems”), 3:7-9 (“With regard to the delivery of games, emulators execute game code that is written to run on various gaming platforms.”).

66. Lewin also describes that by using an application to manage the download of services, the system can adjust to “adapt to whatever hardware the user uses” including on “a set top box of a cable or satellite system, a personal digital

assistant (PDA) device, cell phone or other digital electronic device.” Exhibit 1004 (Lewin), 10:47-52.

C. Zilliacus

67. U.S. Patent No. 6,832,230 to Zilliacus (“Zilliacus”) (Exhibit 1006) was published on December 14, 2004, at least six months before the earliest possible priority date of the ’883 patent.

68. Like the ’883 patent, Zilliacus recognizes that the devices of the time had memory capacity constraints. *See* Exhibit 1006 (Zilliacus), 1:31-45 (“The same technological advances that have fostered the proliferation of the cell phone have also fostered great advancements in computing devices. Computing devices are becoming increasingly smaller, faster, and more mobile. Technological advancement in computing devices and mobile terminals is blurring the line between them. Mobile terminals are integrating more and more capability typically associated with conventional computing devices. For example, cell phones are now able to run software applications and provide Internet access.”), 3:16-19 (“Because of limited memory capacity, a user or the mobile terminal itself may have to eliminate an old application in memory to make room for a new application in memory.”), 3:2-4 (“For example, current mobile terminals typically have less memory available for applications than current PCs.”), 3:14-16.

69. Zilliacus provides an improvement over the distribution technology of mobile apps to address these memory constraints. Zilliacus discloses a system where, upon download, a user can specify the duration that an application will reside on their mobile device. Exhibit 1006 (Zilliacus), 4:6-19 (“Another embodiment of the present invention is a method for downloading an application to a mobile terminal from an application database containing at least a first application. The application has a selectable lifetime during which the application is executable. The lifetime may be preprogrammed in the application or programmed in the application after the lifetime is selected. When the user of the mobile terminal chooses an application to download, the user also selects the lifetime for that application during which the application is executable. The lifetime may be determined on a transaction or time basis. The application is then downloaded to the mobile terminal from an application database. The user’s identification and lifetime selection is also stored in an application-license database.”). At the end of this duration, the unused applications are deleted from the device to make room for new applications. *Id.*, 4:20-23 (“When the application is downloaded to the mobile terminal it may be configured to delete itself or uninstall itself from memory of the mobile terminal when the lifetime expires.”). Alternatively, users themselves could make these determinations. *See id.*, 7:54-59 (“If a user needs to download an application to his or her mobile terminal when there is no memory available to store the application,

the user will have to delete an existing application or an application may be deleted automatically to make room for the new application.”).

X. SPECIFIC GROUNDS FOR UNPATENTABILITY

70. Grounds 1-3 below detail my opinion that claims 1, 2, 5-9, 17-23, 26-30, 38-40, 81-84, 87-91, and 99-101 would have been obvious in light of Montagna, Lewin, Mathews, and Zilliacus.

A. Ground 1: Claims 20-23, 26-30, 38, 1, 2, 5-9, 17, 81-84, and 87-91 are Obvious Over Montagna Alone or in View of Lewin

71. For Ground 1, I considered Montagna alone, or in combination with Lewin. Montagna discloses a method of delivering parts of a game using a “UI Definition,” which is transmitted to a user device so that it can determine what assets it already has in local storage, and what assets it needs to retrieve from a server. Montagna discloses that its method may be implemented on a broad range of devices, which, in my opinion, at that time, would have included mobile devices. Lewin discloses a similar system that explicitly uses mobile devices to deliver games using a similar method. To better present my analysis, I start with independent claim 20 before addressing other claims that have overlapping limitations.

1. Independent Claim 20

a) 20[preamble]: “A method of implementing a consumer service with a server comprising:”

72. In my opinion, even if the preamble of independent claim 20 is considered limiting, Montagna discloses it. As I explained above, Montagna discloses a method of delivering a game to a user device in a way that conserves memory and network resources. Montagna implements this method (i.e., a consumer service) with a server. In the following subsection (i), I will explain how Montagna discloses a “consumer service” (i.e., gaming applications); and in subsection (ii), I will explain how Montagna implements the consumer service with a server.

i. “A method of implementing a consumer service”

73. Montagna discloses this limitation because, like the '883 patent, it describes a method of providing a gaming application, which is a consumer service. *See* Exhibit 1003 (Montagna), [0002] (“The present specification relates to user interfaces.”), [0003] (“In general, applications that interact with users do so through a user interface, which is typically but not necessarily graphical in nature. Applications that administer on-line auctions, for example, provide graphical user interfaces through which one or more users can place bids and be presented with the results of their bids. Casino style gaming applications (or simply, gaming applications) also provide graphical user interfaces, which typically provide users

with a gaming environment for a game and allow users to provide input and receive output from the game.”), Abstract, [0028]; Exhibit 1001 (’883 patent), 1:54-58 (“The present invention relates to the field of distributed entertainment. More specifically, the present invention relates to the field of distributed entertainment utilizing a computing device where the entertainment corresponds to an event.”), 7:41-47 (“[T]he “Activity Client” has the ability to receive and hold downloaded activity-specific data instructions from the server. Thus, when a specific activity is selected for play by the user, for example, Monday Night Football™, Wheel of Fortune™, or Major League Baseball™, the server downloads the activity-specific instructions....”). The system described in Montagna creates games of chance within an application, such as “craps, blackjack, roulette, baccarat, poker, keno, bingo, and progressive jackpots,” however, it contemplates that its method has broad applicability and use, including in “applications other than the described gaming and auction applications.” Exhibit 1003 (Montagna), [0004], [0107].

74. Montagna’s gaming application (i.e., a consumer service) is provided to users. Exhibit 1003 (Montagna), [0010] (“The system is configured to personalize user interfaces in real time. In response to *user* input, for example, a request to participate in on-line gaming, the system can generate a user *interface that is personalized for the specific user* and for the current request with little latency.”), [0058] (“To initialize a new game, the *user* begins by clicking a hyperlink to select

a game from a personalized game menu. The user's browser then loads game client 304. Game client 304 then requests a UI definition from GDE 302. GDE 302 builds and sends to game client 304 a UI definition based on specific user information and configurable criteria as described above with reference to FIGS. 1 and 2. The UI definition includes a list of selected resources. After receiving the UI definition, game client 304 obtains resources necessary to generate a user interface for the selected game. ***Game client 304 then generates a user interface for the game and the user can initiate game play.***”).

75. The method described by Montagna builds a personalized interface in the gaming application based on, among other things, the geographic location of the user, the user's preferences, and the casino games that are legal in the user's jurisdiction. Exhibit 1003 (Montagna), [0009] (“The system can provide a user interface that can be personalized for each user. Personalization can be based on, for example, laws to which a specific user is subject, groups of which the specific user is a member, preferences of the specific user, and previous actions of the specific user.”), [0010] (“If there are changes that occur between a first request and a second request, for example, the specific user changes residency from the United Kingdom to a jurisdiction that permit the above-mentioned illegal gaming options, the system can make the appropriate changes during its personalization process.”), [0007],

[0033], [0036], [0043], [0044]. Montagna provides an exemplary interface for a game of poker in Figure 11:

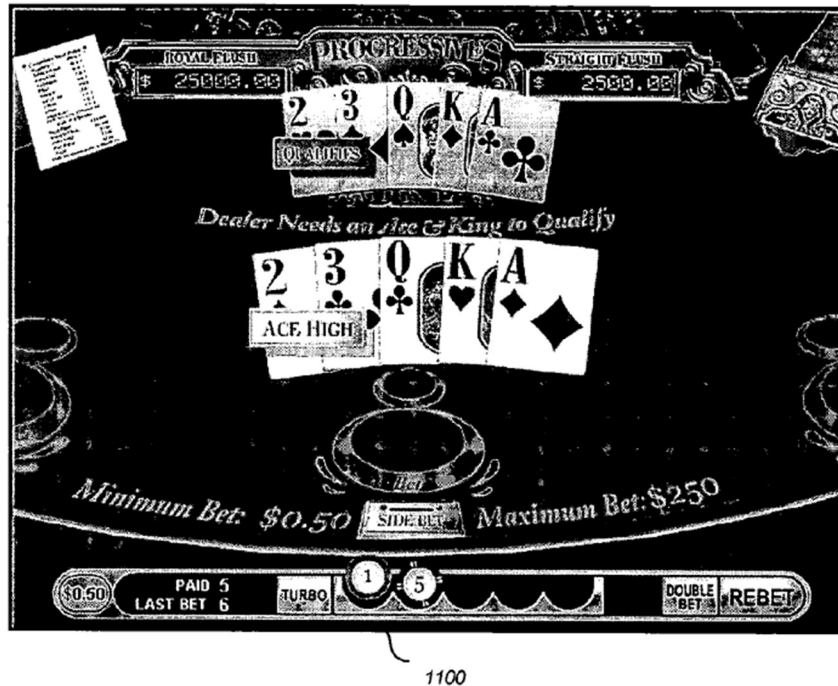


Exhibit 1003 (Montagna), Figure 11

76. When building a game interface, Montagna seeks to minimize the amount of information that needs to be transferred to the user's device. Exhibit 1003 (Montagna), [0045] ("In the instant example, the UI definition indicates that several user interface components are available at a certain location on a network accessible Web server.... [C]ertain resources can be available in local cached memory resulting from previous personalized user interfaces. UI interpreter 112 will not have to retrieve these resources."), [0014] ("Furthermore, resources from previous game

sessions can be reused in a current game session and the time needed to download resources to a client terminal is thus reduced”).

ii. “with a server”

77. Montagna discloses the use of a server to implement its services. For example, Montagna discloses that “[t]he mentioned assembling process can include retrieving user interface components from a networked resource, for example, a Web **server** or Networked File Structure.” Exhibit 1003 (Montagna), [0031]; *see also id.*, [0105] (“The invention can be implemented in a computing system that includes a back-end component, e.g., as a data server, or that includes a middleware component, e.g., an application server, or that includes a front-end component, e.g., a client computer having a graphical user interface or a Web browser through which a user can interact with an implementation of the invention, or any combination of such back-end, middleware, or front-end components.”).

78. The system described by Montagna is shown in Figure 3, annotated below. A user interacts with the consumer service through **game client 304**⁵, which resides on **client terminal 316**, which is in communication with **server 314**. As I

⁵ As I noted previously, Montagna’s Figure 3 contains a typo: the game client is shown labelled as 306, rather than 304, which is how it is labelled elsewhere in the specification.

will discuss in further detail below, the components used to retrieve game assets reside on **server 314**. Exhibit 1003 (Montagna), [0049].

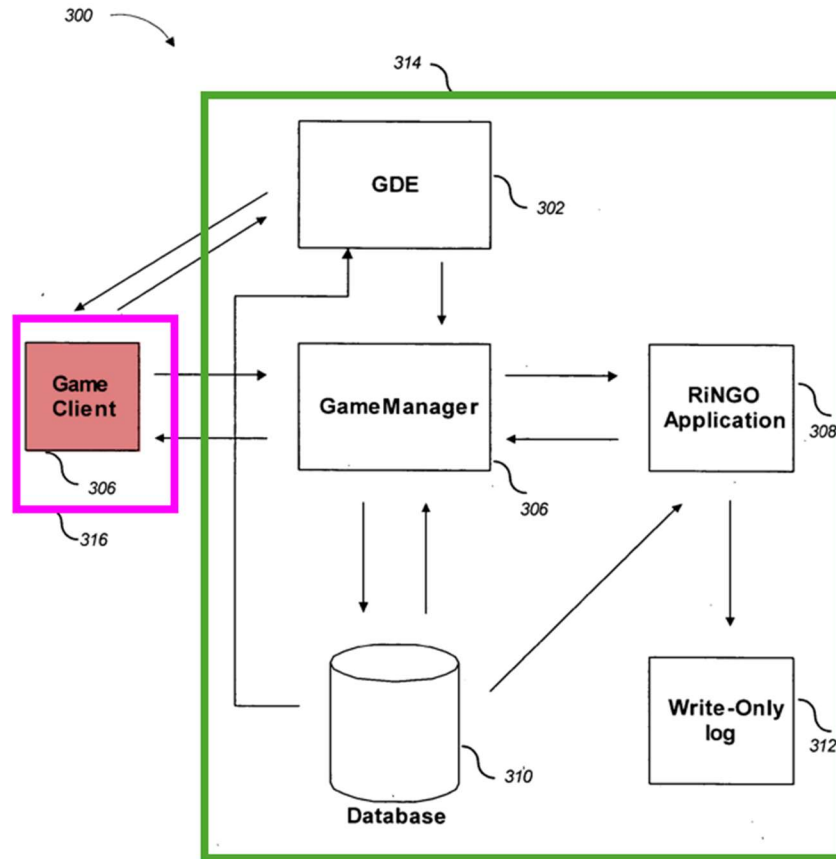


Exhibit 1003 (Montagna), Figure 3 (annotated)

b) 20[a]: “transmitting a set of service related information to an application;”

79. Montagna discloses limitation 20[a]. I will describe how Montagna discloses this in two sections below. In the following subsection (i), I explain how Montagna discloses a set of service related information (i.e., a menu of games). In subsection (ii), I explain how Montagna teaches that the set of service related information (i.e., a menu of games) is presented to an application (i.e., the game

client). In subsection (iii), I explain how Montagna transmits the set of service related information from the server to the game client residing on the client terminal.

i. “a set of service related information”

80. Montagna discloses a set of service related information because it teaches a list of game options related to the consumer service (i.e., the gaming application). Exhibit 1003 (Montagna), [0061] (“The user is provided a game menu that can include several versions of each game title, each using different denominations.”), [0058], [0061], [0063], [0065], [0067], [0068], [0070], [0072], [0073], [0074], [0083]. The ’883 patent describes that a similar menu of options is transmitted to an application (the activity client) on the user’s device. *See, e.g.*, Exhibit 1001 (’883 patent), 2:65-3:2 (“In one aspect, a method of implementing an entertainment service on computing devices comprises downloading an activity client to a computing device, **loading a set of event information to the activity client**, selecting an event to participate in from a list of possible events....); 3:14-21 (“The schedule of available events includes a name of a television event, identification of an activity to be played with the television event, a television network and local channel offering the television event, a duration of the television event and information relating to nature of a prize related to the television event. The method further comprises displaying the set of event information.”). The ’883 patent does not otherwise define “service related information.”

ii. “to an application”

81. Montagna discloses that service related information (i.e., a menu of games) is presented to the user on an application (i.e., the game client). Montagna describes a “game client,” which “collects user input” and “communicates it back to the server-side components of the game platform.” Exhibit 1003 (Montagna), [0051]. The game client includes a “master game page” that can be a “browser window that presents a game to a user.” *Id.*, [0051], [0028] (“The user interface is implemented as part of gaming applications.”), [0014], [0049], [0052], [0058], [0105], [0107]. Thus, like the ’883 patent, Montagna describes an application (i.e., game client) that receives a set of service related information (i.e., a game menu). *See* Exhibit 1003 (Montagna), [0051] (“Game client 304 includes a master game page, a loader, and a user interface. The master game page can be a browser window that presents a game to a user. The loader is the above-mentioned script that interprets the UI definition and generates the user interface.”); Exhibit 1001 (’883 patent), 4:15-17 (“In another aspect, a client device comprises a communication interface, an activity client for receiving activity information through the communication interface and a display for displaying the activity information received by the activity client.”), 6:30-32 (“The present invention utilizes a software application, referred to as an ‘Activity Client.’”).

82. Montagna describes that its game client is executed on a device. Exhibit 1003 (Montagna), [0049] (“Game platform 300 includes a game definition engine (‘GDE’) 302, a game client 304, a game manager 306.... GDE 302, game manager 306, RiNGO application 308, database 310, and write-only log 312 reside on a server 314 while the game client 304 resides on a client terminal 316.”), [0101], [0103]. As illustrated in Montagna’s Figure 3, annotated below, the **game client 304** (i.e., an application)resided on the **client terminal 316** (i.e., a user device),.

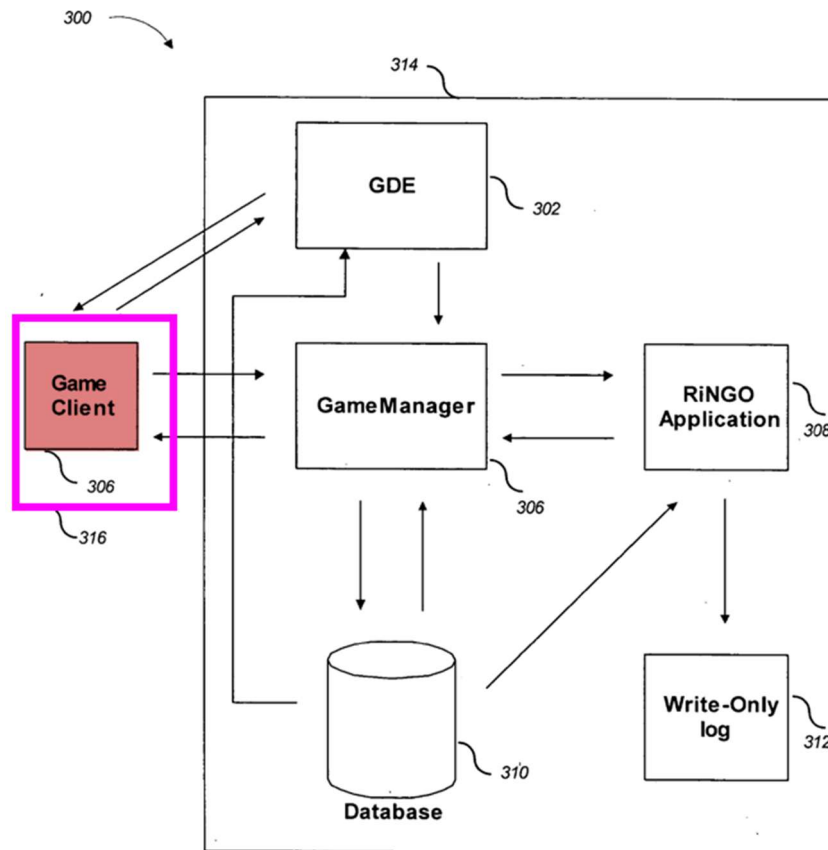


Exhibit 1003 (Montagna), Figure 3 (annotated)

iii. “transmitting”

83. Montagna describes that the set of service related information (i.e., a game menu) is transmitted from the server to the application (i.e., game client). Montagna describes that a “game definition engine” (“GDE”) 302 defines the game menu which is presented to the user. Exhibit 1003 (Montagna), [0065] (“GDE 302 delivers games into two phases. In the first phase, the GDE 302 defines a game menu.”), [0061] (“One example of a game definition process occurs as follows. A user logs on to an on-line casino. The user's profile indicates the user's currency preference is, for example, United Kingdom Pounds and language preference is, for example, English. GDE 302 accesses user specific information and rules similar to the above-described UI rules to determine which games are available to the user.”), [0068] (“When GDE 302 builds a game menu for a user, GDE 302 takes into account the game configurations, which specify what game titles, denominations, and pay models are available to users.”), [0058], [0105].

84. Because the GDE 302 is held on the server, for the service related information (i.e., game menu) to be presented to the user within the application (i.e., game client), the service related information must be transmitted from the server to the application (i.e., game client). Exhibit 1003 (Montagna), [0049] (“GDE 302, game manager 306, RiNGO application 308, database 310, and write-only log 312 reside on a server 314 ...”). Montagna’s Figure 3, annotated below, illustrates the

game client 304 (i.e., the application), the **client terminal 316** (i.e., the user device), the **server 314**, and the **GDE 302**, which defines the set of service specific information (i.e., game menu).

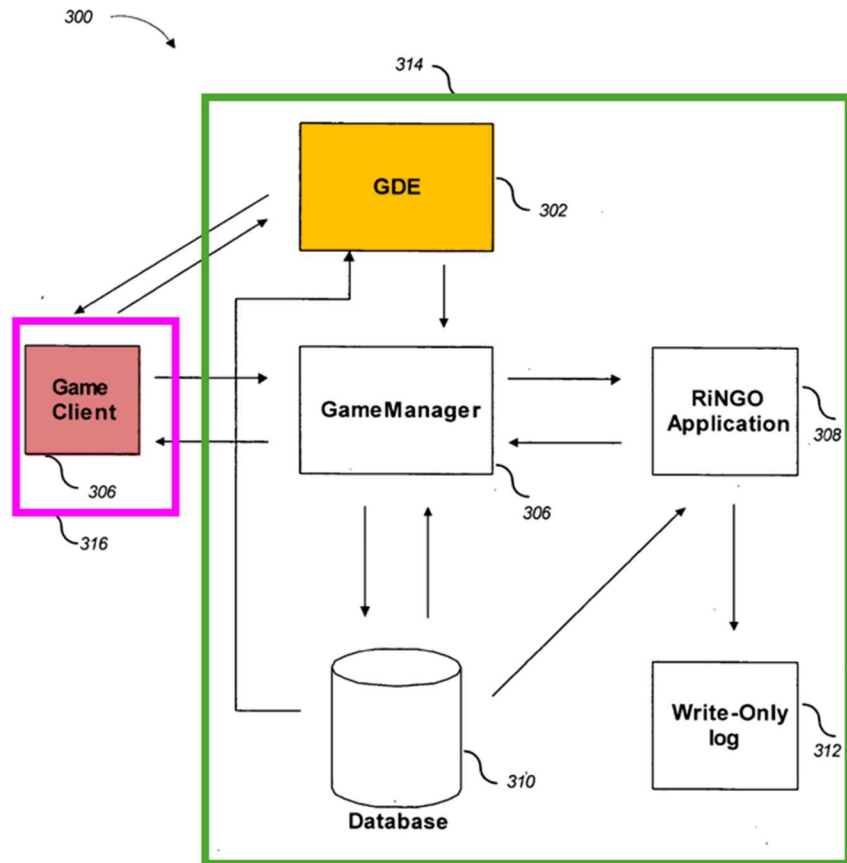


Exhibit 1003 (Montagna), Figure 3 (annotated)

85. Transmission between GDE 302 (held on the server 314), and the game client 304 (residing on client terminal 316) is illustrated by the arrows. Montagna describes that the communication between the GDE 302 on the server and the game client. Exhibit 1003 (Montagna), [0055] (“[G]ame platform 300 can be distributed over a network. The network can be, for example, any combination of: the Internet,

a network that includes Wireless Application Protocol ("WAP") clients, and a network that includes non-browser based clients.”), [0050] (“***GDE 302 is configured to personalized game delivery to users.*** Personalization takes into account branding, language, and currency, as well as game logic.”), [0051] (“Game client 304 collects user input (such as a wager) and ***communicates it back to the server-side components of the game platform 300.***”), [0052] (“Game manager 306 provides the server side management of an on-line game. Game manager 306 receives user requests from game client 304 and ensures the user receives appropriate responses.”).

c) 20[b]: “receiving a selection related to the consumer service;”

86. Montagna discloses that the server 314 receives a selection related to the consumer service. Montagna describes a selection (i.e., the user’s game selection). Exhibit 1003 (Montagna), [0006] (“The method includes receiving a request for a user interface for a game.”), [0058] (“To initialize a new game, the user begins by clicking a hyperlink to select a game from a personalized game menu. The user’s browser then loads game client 304.”), [0087] (“After logging into a casino system, a user selects a game to play from a game menu (step 702). . . .”), [0085] (“After a user selects a game from the Game Menu page. . . .”), Abstract, [0044],

[0051], [0052], [0058], [0071], [0087]. The selection (i.e., the user's game selection) relates to the consumer service because it is a game that the user would like to play.

87. Montagna describes that the user makes the selection at the game client 304 (i.e., the application) and the selection is then transmitted to the server, which receives the selection. Exhibit 1003 (Montagna), [0051] (“Game client 304 collects user input (such as a wager) and communicates it back to the server-side components of the game platform 300.”); [0052] (“Game manager 306 provides the server side management of an on-line game. *Game manager 306 receives user requests from game client 304* and ensures the user receives appropriate responses.”). As Montagna illustrates in Figure 3, annotated below, the **game client 304** receives information from and sends information to the **server 314**, including the **game manager 306**. *See id.*, [0049].

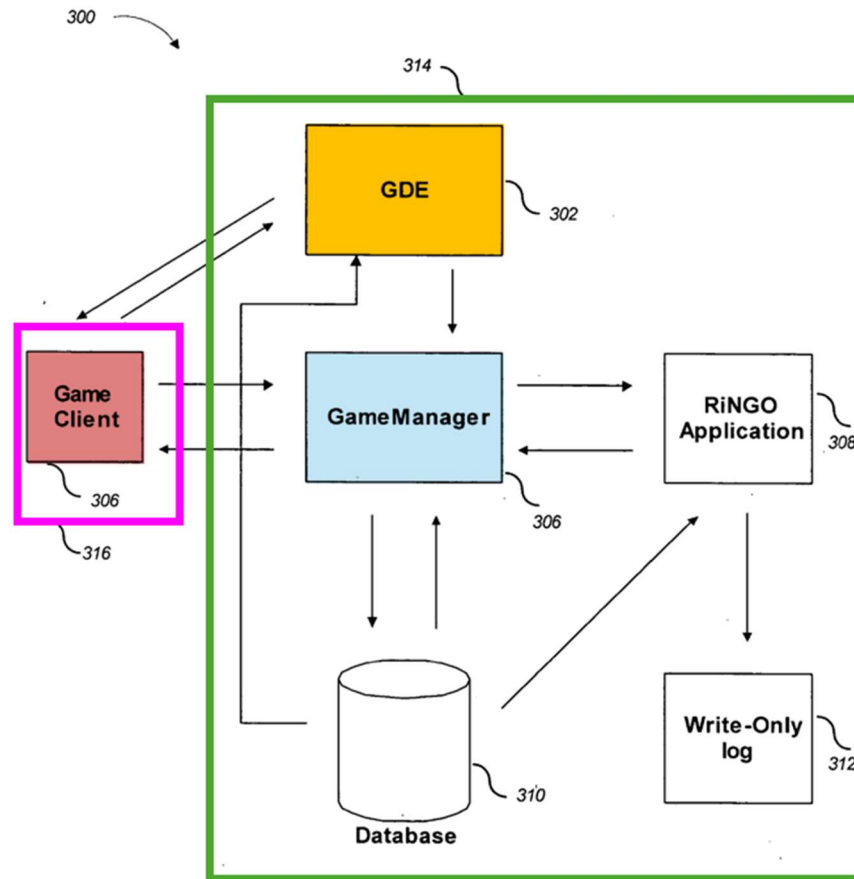


Exhibit 1003 (Montagna), Figure 3 (annotated)

d) 20[c]: “transmitting a set of selection information related to the selection from the server to a mobile Internet-connected computing device;”

88. Montagna alone, or in combination with Lewin, renders obvious this limitation. Below, in subsection (i), I explain how Montagna alone, or in combination with Lewin, teaches a set of selection information related to the selection (i.e., the choice of a game selected). In subsection (ii), I explain how Montagna and Lewin each discloses that its set of selection information is transmitted from a server. In subsection (iii), I explain how a POSITA would have

understood that Montagna teaches transmission to a mobile Internet-connected computing device and how Lewin explicitly discloses transmission to a mobile Internet-connected computing device.

i. “a set of selection information related to the selection”

89. The '883 patent does not explicitly define “selection information.” However, the '883 patent describes information that is sent after the selection. For example, the '883 patent describes that “instructions and options” are downloaded and displayed on a user’s device after the user makes a selection. *See, e.g.*, Exhibit 1001 ('883 patent), 11:56-62 (“In operation, a user is able to use his cell phone or other computing device to enjoy interactive entertainment. For example, the user scrolls through the menus and selects the interactive entertainment desired to be played. ***Thereafter, downloads occur, and instructions and options are displayed on the display*** so that a user is able to read about and determine which interactive entertainment he wants to play.”). Montagna and Lewin each teaches a set of selection information related to the user’s selection.

90. First, Montagna describes an error message that is transmitted and that tells the user if their selected game is improper because they already have a game in progress. As Montagna illustrates in Figure 6 (reproduced below), this error message includes a set of information: (1) a message that indicates “A Game is

already in progress,” (2) the title of the game in progress, and (3) the denomination of the game in progress, and (4) an error code. Exhibit 1003 (Montagna), Figure 6, [0085], [0086].

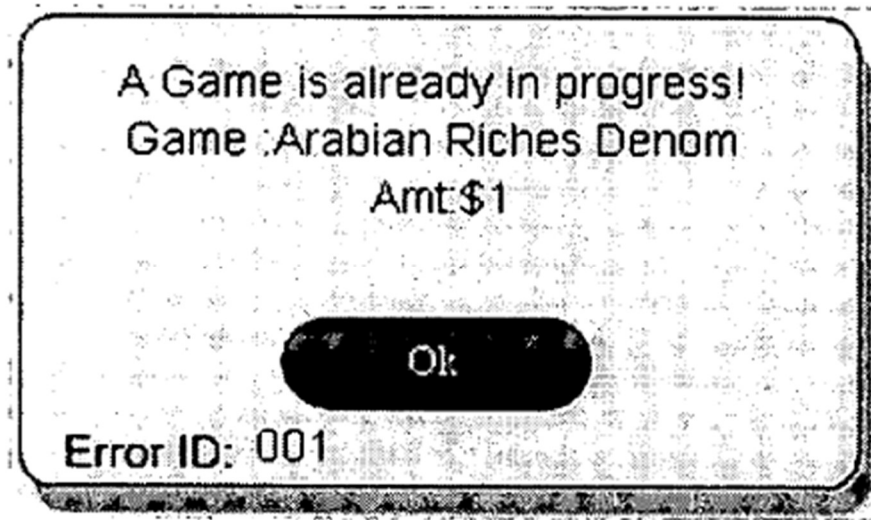


Exhibit 1003 (Montagna), Figure 6

91. After receiving the user’s selection, Montagna’s system recognizes whether the user had already started an instance of that game. If the user attempts to start another instance of the same game, the system will send the user the error message, which notifies them that their selection is against the rules. Exhibit 1003 (Montagna), [0085] (“In a different circumstance, game platform 300 uses an error message to ensure the user completes an incomplete game.... To prevent a user from beginning a new game, the system presents an error message indicating that the user has a game in progress that they must finish before beginning a new game.”), [0082],

[0084], [0085]. Montagna's error message is a set of selection information related to the game the user has selected that is sent after the selection is made.

92. Second, Montagna discloses the transmission of real-time updates, which alter the game that the user has selected after execution has begun. Exhibit 1003 (Montagna), [0056], [0095], [0096] ("The Game Definition XML updater can provide real time Game Definition XML updates with each change in game state."), [0097], [0098]. If the user is playing a "progressive game" which changes the jackpot based on the activity of other users, information about the progressive jackpot is updated during gameplay. Exhibit 1003 (Montagna), [0090] ("[T]he user interface can be generated to include information for progressive games such as a progressive jackpot total. A progressive game provides an opportunity for a user to win a progressive jackpot under defined circumstances. The progressive jackpot increases with game play until someone wins the progressive.... A separate progressive servlet can also be used to provide real time updates for Game Definition XML to the Flash Game Loader."), [0061] ("After sending the UI definition to the loader, updates to the UI definition are sent by the game manager 306 (i.e., a game servlet and/or a progressive game servlet)."), [0078] ("Most of the XML sub-components used to build the Game Definition XML are fixed, however certain portions are generated dynamically. Fixed components of the XML are retrieved from the database; dynamic portions are pulled from the XML Game Description,

user profile, the user's game state, and, for progressive games, a progressive gateway.”). Montagna’s real-time updates comprise a set of selection information.

93. In my opinion, between these two disclosures, Montagna teaches a set of selection information as claimed. However, if Montagna’s disclosures were to be found insufficient, Lewin discloses a set of selection information. Lewin describes “transitional content,” which is a set of selection information. Exhibit 1004 (Lewin), 10:66-11:1 (“Still another aspect of the system is the delivery of commercials or other transitional content between game levels or during download wait times”), 11:1-42, claim 39. Lewin’s transitional content is sent after a user makes a selection but before they play the selected game, and includes “mini-games, tips, game cheats, game *instructions* and a brief history about the game being downloaded.” Exhibit 1004 (Lewin), 11:40-42.

94. To determine this transitional content, Lewin describes that, after a user makes a selection of a game or content, the server determines information about the game or content selected. Exhibit 1004 (Lewin), 8:17-20 (“In response to a request from a client application 15 for a game or other content, the host server 20 retrieves information about the requested game from a metadata table or datastore 35.”). Lewin describes that this “metadata” includes information like the title, release date, number of players, and rating or other parental guidance information. *Id.*, 8:20-27 (“The metadata datastore 35 typically stores information about the available content

rather than storing the content itself. In the context of games content, the data stored in the metadata datastore 35 may include a game title, release date, genre, number of players, supported game controllers, minimum and recommended system requirements, original platform and an entertainment software rating board (ESRB) rating or other parental guidance indicia.”). This metadata is then used to determine what transitional content to deliver to the user’s device. *Id.*, 11:32-35 (“The host server 20 delivers any specific content specified in the metadata datastore 35, and if none is specified, delivers the advertising content in accordance with the priority list.”), 11:38-42 (“Examples of non-commercial forms of transitional content that can be delivered to a user as a requested game is downloading can include mini-games, tips, game cheats, game instructions and a brief history about the game being downloaded.”).

95. Lewin’s transitional content (i.e., tips and game instructions) closely resembles the selection information described in the ’883 patent. Exhibit 1001 (’883 patent), 11:57-63 (“For example, the user scrolls through the menus and selects the interactive entertainment desired to be played. Thereafter, downloads occur, and ***instructions*** and options are displayed on the display so that a user is able to read about and determine which interactive entertainment he wants to play. After selecting the activity, again further downloads occur.”).

*(a) Motivation to combine Montagna and
Lewin*

96. A POSITA would have found it obvious to combine the transitional content taught by Lewin with the game distribution system taught by Montagna for the following reasons.

97. As I discussed in the Technology Background section above, in 2005, a POSITA would already been well versed in both game development and development of games for mobile devices. Moreover, in and around 2005, a POSITA would have known that network bandwidth was limited and downloads of files (especially large game files) could be relatively slow. Accordingly, a POSITA would have been highly motivated to improve the user experience by entertaining (or otherwise distracting) the user during the long download time.

98. A POSITA would also have readily appreciated the benefits of adding a set of selection information (i.e., instructions about a game) to Montagna's system, as described by Lewin. As Montagna describes, and as a POSITA would have been well aware, gaming applications can be tailored to a wide variety of games, for example "craps, blackjack, roulette, baccarat, poker, keno, bingo, and progressive jackpots." Exhibit 1003 (Montagna), [0004]. Because of the wide variety of games offered in Montagna's method, a POSITA would have found it obvious to provide a user with instructions and other transitional content before the user receives the

game, as described by Lewin. For example, a POSITA would have readily appreciated that the user would be better able to enjoy the game (consumer service) once available on their device because they already have been provided with instructions on how to play the game. A POSITA would appreciate that these instructions and other information about the game (sets of selection information) improves the user experience. Lewin itself makes note of this: “In the context of a gaming network, the system 10 manages the user experience by offering commercial advertisements or other forms of audiovisual content between game levels and during download wait times.” Exhibit 1004 (Lewin), 9:67-10:3.

99. Lewin’s discussion of foreground and background processes would not have dissuaded a POSITA from combining Lewin’s teachings with Montagna’s system. For example, in the context of computer processes, “foreground” is generally understood to mean: “The condition of the program or document currently in control and affected by commands and data entry in a windowing environment.” Exhibit 1012 (Microsoft Computer Dictionary) at 221. In the same context, “background” is generally understood to mean: “In the context of processes or tasks that are part of an operating system or program, operating without interaction with the user while the user is working on another task. Background processes or tasks are assigned a lower priority in the microprocessor’s allotment of time than foreground tasks and generally remain invisible to the user unless the user requests

an update or brings the task to the foreground.” *Id.* at 47-48. Since foreground and background generally refer to whether the user is directly interacting with that program at that time, this distinction would not have dissuaded a POSITA from combining Lewin’s teachings with Montagna’s system.

100. Montagna and Lewin are also both from the same field, which is shared by the ’883 patent, providing a POSITA a motivation to combine them. Montagna and Lewin both relate to the provision of gaming on a user device. Exhibit 1003 (Montagna), Abstract (“A computer program product for generating a user interface definition includes instructions to cause a processor to receive a request for a user interface for a game.”); Exhibit 1004 (Lewin), Abstract (“A content delivery system that uses a graphical user interface and host avatar to introduce users to and allow them to select from available content. A game delivery system that uses emulators to execute software written to run on a plurality of game platforms.”). Both Montagna and Lewin also address the same problem: how to deliver a variety of games to a user’s device after a selection is made from a list of options. Exhibit 1003 (Montagna), [0010] (“The system is configured to personalize user interfaces in real time. In response to user input, for example, a request to participate in on-line gaming, the system can generate a user interface that is personalized for the specific user and for the current request with little latency.”); Exhibit 1004 (Lewin), 9:49-63 (In Step 50, the asset server 25 retrieves the requested content from the content

datastore 40.... Depending on the size of the content file or files being transferred, the content may be delivered in segments or as a single file. For example, games 55 that were originally written for the Atari and Nintendo video console systems are typically less than a megabyte in size and require just a few seconds to download to the client application 15. In contrast, a medium-size game written for the Sony PlayStation console is about 100 megabytes in size, and a 60 game written for a personal computer typically ranges between several hundred megabytes to several gigabytes of data.), Abstract (“[A] combination of linear and on-demand content provides users with a managed gaming experience not unlike that of interactive television.”). Both describe similar solutions to this problem—by providing the user with a menu of options, and transmitting the user their selected game option from a server). Exhibit 1003 (Montagna), [0074] (“After the user selects a game title and denomination from the game menu, GDE 302 sends to the loader (which, in the instant implementation is a Flash Game Loader) a list of all the game assets needed to display the game user interface”), [0088] (“In one implementation, the resources are downloaded from a gaming server and can be in compressed or uncompressed form.”); Exhibit 1004 (Lewin), 3:14-16 (“An embodiment of the present invention herein described is a game delivery system that allows a user to select from and receive delivery of at least one of a plurality of games via a network....).

101. In my opinion, when designing the system of Montagna, there would have been a limited set of options regarding what occurs after a user selects a game to play. One option would be to immediately send the list of assets, which begins the process of transferring the game. Another option would be for the system to transmit a set of information beforehand (i.e., a set of selection information), for example an error message if an error were to occur (as discussed in Montagna), or an instruction or some content that allows a user to better pass the time (as discussed in Lewin). Given these limited options, and considering the benefits of the transitional content described by Lewin, a POSITA would have chosen to send transitional content. As I discuss next, I also believe a POSITA would have a reasonable expectation of success in making this combination.

102. A POSITA would have combined of Montagna's system with Lewin's set of selection information according to known methods and the combination would have yielded predictable results. A POSITA would have used basic and well-known computer programming principles to include Lewin's set of selection information in Montagna's system. Such a combination would have yielded the predictable results of providing a set of selection information to the user while the game is being downloaded. A POSITA would have possessed the knowledge and skillset to perform this combination.

103. A POSITA would have recognized that Montagna and Lewin disclose similar systems and thus including Lewin's set of selection information in Montagna's system would have merely amounted to implementing known techniques to improve similar systems. Montagna's system allows users to select a game from a plurality of games, just like Lewin's system. Exhibit 1003 (Montagna), Abstract; Exhibit 1004 (Lewin), Abstract. Therefore, a POSITA would have readily used basic and well-known computer programming principles to implement Lewin's set of selection information in Montagna's system to enhance the user experience "between game levels and during download wait times." Exhibit 1004 (Lewin), 9:67-10:3.

***(b) Reasonable expectation of success in
combining Montagna and Lewin***

104. A POSITA would have had a reasonable expectation of success in combining Montagna and Lewin as I discussed above. Both Montagna and Lewin use a server and a client device to send information and games to a user. Exhibit 1003 (Montagna), [0105] ("The invention can be implemented in a computing system that includes a back-end component, e.g., as a data server... or that includes a front-end component, e.g., a client computer having a graphical user interface or a Web browser through which a user can interact with an implementation of the invention, or any combination of such back-end, middleware, or front-end

components. The components of the system can be interconnected by any form or medium of digital data communication, e.g., a communication network.”), [0101]; Exhibit 1004 (Lewin), 3:3-4 (“The present invention is directed to methods and systems of delivering entertainment content via a network.”), 3:21-25 (“The game delivery system includes an asset server that stores software game code associated with the three different sets of games, and a host server that directs the asset server to deliver a selected one of the games to the user.”). This type of client/server arrangement—where a client device requests information from a server, was well known as of 2005. *See, e.g.*, Exhibit 1036 (Kryzyanowski) at 4 (“The most common networking relationship is the client-server model. The model contains three components: a client, a server, and a service. A service is that task that a machine can perform (such as offering files over a network or the ability to execute a command). A server is the machine that performs the task (the machine that offers the service). A client is the machine that is requesting the service.”).

105. Both Montagna and Lewin use XML data structures to deliver information to user devices. Exhibit 1003 (Montagna), [0038]; Exhibit 1004 (Lewin), 12:22-29, 17:50-53. Thus, not only do Montagna and Lewin use the same methodology to deliver the service (a client/server network), but they use the same language to deliver the service over that network.

106. Thus, it would not require much experimentation or difficulty for a POSITA to integrate features from Lewin's disclosure to the system disclosed in Montagna.

ii. "transmitting...from the server"

107. Montagna and Lewin each describe transmitting a set of selection information from a server.

108. Montagna describes that its set of selection information (i.e., error messages, or updated game information) comes from the server. Exhibit 1003 (Montagna), [0052] ("Game manager 306 provides the server side management of an on-line game. Game manager 306 receives user requests from game client 304 and ensures the user receives appropriate responses. Game manager 306 is responsible for keeping track of game sessions, requesting a game outcome from RiNGO application 306, saving game states in the database 310 for record keeping and refresh, updating the user's account as appropriate for the game state (e.g., debit/credit money), and keeping track of progressive game play if applicable."), [0105], [0053], [0055].

109. Montagna explains that, when the error message (i.e., the set of selection information) warns the user of a game in progress, this information comes from the server. In particular, the **server 314** retrieves games in progress from the **database 316**, and in conjunction with the database, warns a user of an in-progress

game. Exhibit 1003 (Montagna), [0085] (“To prevent a user from beginning a new game, the system presents an error message indicating that the user has a game in progress that they must finish before beginning a new game.”); [0093] (“The INTERFACESTATE element is the data exchanged between client and server.”); Table 2 (“INTERFACESTATE: Determined by game state. May be initial game state determined by XML Game Description or saved game state retrieve[d] from database.”). Montagna’s Figure 3 depicts how saved game state is retrieved from the **database 310** by the **game manager 306**, and transmitted to the **game client 304**. Exhibit 1003 (Montagna), [0049].

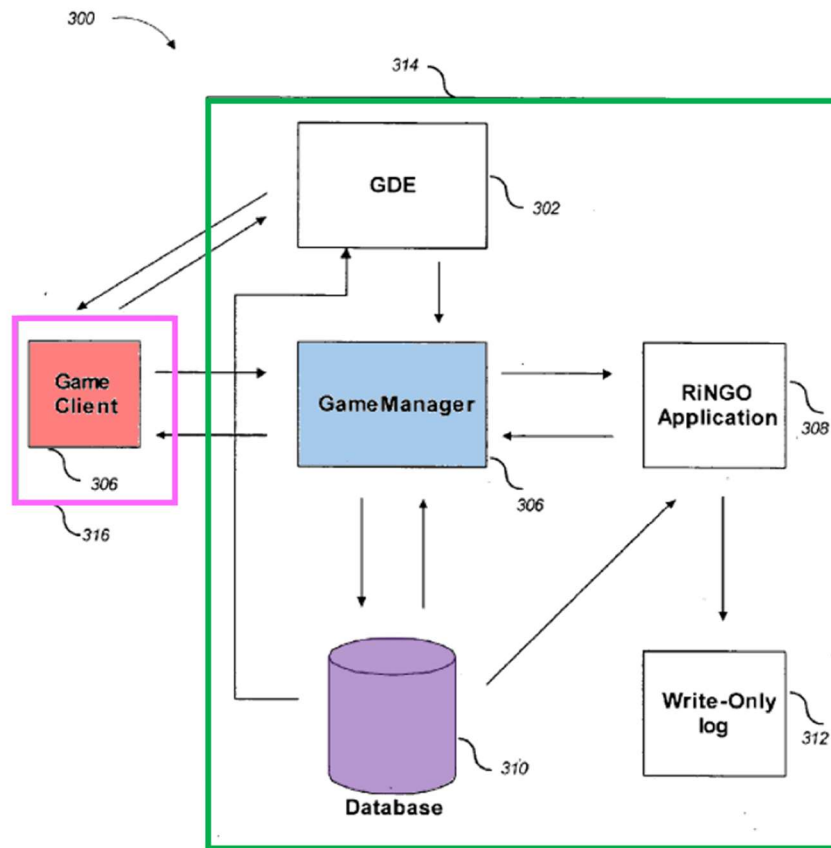


Exhibit 1003 (Montagna), Figure 3 (annotated)

110. Montagna also describes that game updates (i.e., the set of selection information) are sent from the **game manager 306**, which resides on the **server 314**, to a loader, which is part of the game client on the user device. Exhibit 1003 (Montagna), [0051] (“Game client 304 includes a master game page, a loader, and a user interface.”), [0061] (“After sending the UI definition to the loader, updates to the UI definition are sent by the game manager 306 (i.e., a game servlet and/or a progressive game servlet), [0074] (“GDE 302 sends to the loader (which, in the instant implementation is a Flash Game Loader)...”), [0090] (“A separate

progressive servlet can also be used to provide real time updates for Game Definition XML to the Flash Game Loader.”).

111. Lewin also describes that its transitional content (i.e., the set of selection information) is transmitted from a server to the user device. Lewin describes that after metadata⁶ is used to identify transitional content, that transitional content is transmitted to the user’s device. Exhibit 1004 (Lewin), 11:29-34 (“As part of the delivery process, the host server 20 checks the metadata to determine if specified commercials or 30 other transitional content has been specified for the requested game. The host server 20 delivers any specific content specified in the metadata datastore 35, and if none is specified, delivers the advertising content in accordance with the priority list.”), 11:32-35 (“The host server 20 delivers any specific content specified in the metadata datastore 35, and if none is specified, delivers the advertising content in accordance with the priority list.”), 27:57-60 (“[C]lient application is configured to deliver transitional content to said user device when said game file is being transferred and before said game play has begun.”).

⁶ Lewin’s metadata includes “game title, release date, genre, number of players, supported game controllers, minimum and recommended system requirements, original platform and an entertainment software rating board (ESRB) rating or other parental guidance indicia.” Exhibit 1004 (Lewin), 8:20-27.

iii. “to a mobile Internet-connected computing device”

112. Montagna describes that components on the server side (e.g., GDE 302 and game manager 306) are responsible for communication with the game client 304 (i.e., the application) on the user device. Exhibit 1003 (Montagna), [0052] (“Game manager 306 provides the server side management of an on-line game. Game manager 306 receives user requests from game client 304 and ensures the user receives appropriate responses. Game manager 306 is responsible for keeping track of game sessions, requesting a game outcome from RiNGO application 306, saving game states in the database 310 for record keeping and refresh, updating the user's account as appropriate for the game state (e.g., debit/credit money), and keeping track of progressive game play if applicable.”), [0051], [0053], [0058]. Annotated Figure 3 is shown again below. The **GDE 302** and **game manager 306** transmit sets of selection information to the **client terminal 316**, which is used to execute the **game client 304**. Exhibit 1003 (Montagna), [0049].

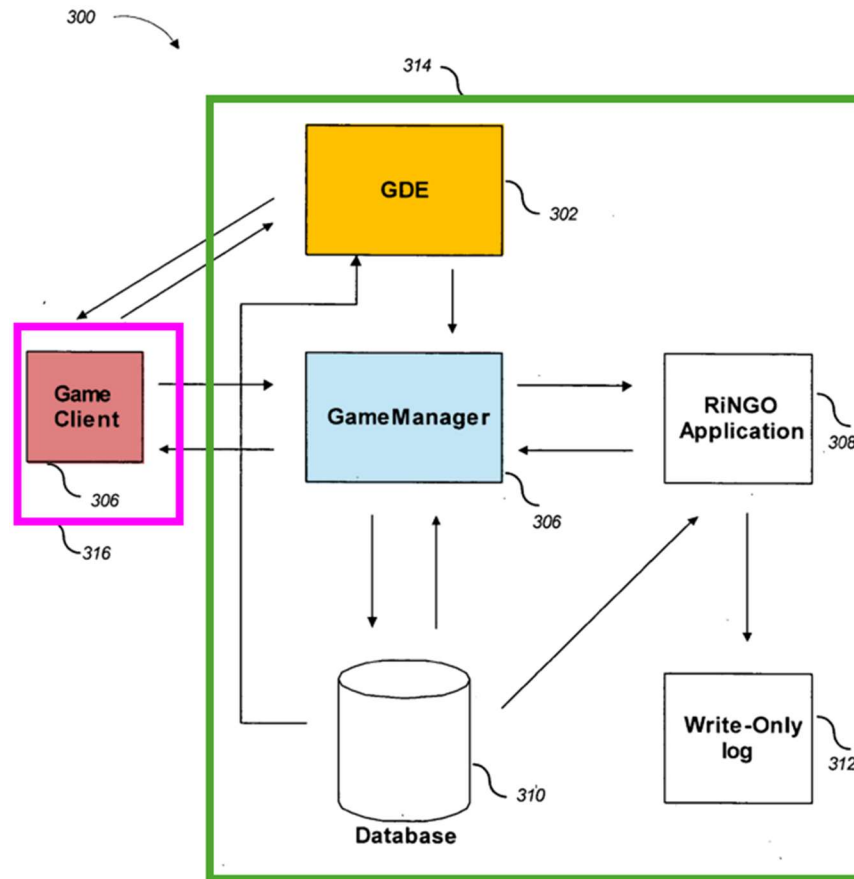


Exhibit 1003 (Montagna), Figure 3 (annotated)

113. Montagna describes that its system, which is designed to minimize memory storage and transmission of data, “can be implemented in digital electronic circuitry, or in computer hardware, firmware, software, or in combinations of them.” Exhibit 1003 (Montagna), [0101], [0103]. Given the advancements in handheld devices and mobile computing in the early 2000s, a POSITA would have found it obvious that Montagna’s combination of computer hardware, firmware, and software would include cellphones and other mobile Internet-connected computing devices. *See id.*, [0055] (“As shown in FIG. 4, game platform 300 can be distributed

over a network. The network can be, for example, any combination of: the *Internet*, a network that includes Wireless Application Protocol (“WAP”) clients, and a network that includes non-browser based clients.”), Figure 4, [105]; *see also* Section VI (Technology Background).

114. However, if Montagna’s disclosure were to be found to not include mobile Internet-connected computing devices, Lewin explicitly describes that the device used to interact with its game distribution system is a mobile Internet-connected computing device. Exhibit 1004 (Lewin), 6:38-40 (“In a preferred embodiment, the various users of the system are distributed over a large geographical area and the network 45 is a wide area network such as the *Internet*.”), 10:46-49 (“Additional users may run the client application 15 on a set top box of a cable or satellite system, a personal digital assistant (PDA) device, *cell phone* or other digital electronic device.”), 8:1-6 (“In alternative embodiments, communication may occur via a transmission control protocol/internet protocol (TCP/IP) socket connection or via some combination of HTTP and a socket connection. Socket connections are well known in the art as a means of continuous data transmission between computer systems.”), 3:57-63 (“the client application has multiple graphic user interface modules that are configured to interact with different end-user devices. In some cases, the user interface of the system is modified to accommodate the specific type of user device, and in other cases the actual game, or

at least the list of available games, are modified based on the specifications of the end-user device.”), 16:22-26 (“[B]ecause the client is not limited to computer systems, user input can be received from electronic devices such as *mobile phones*, wireless handheld devices and cable set top boxes, to name a few.”). As I discussed above, Lewin teaches an analogous system to that of Montagna, where an application on a user device is used to interact with a consumer service (i.e., gaming). *Id.*, 3:26-29 (“Additional embodiments of the game delivery system describe using a client application residing on a computing device associated with the user that displays a listing of available games accepts a user request to play a game.”), 12:22-32, 12:48-54, 7:65-8:16, 8:27-31, 8:40-47, 7:50-55, 16:14-25, 16:27-47, 17:3-15.

(a) Motivation to combine Montagna and Lewin

115. A POSITA would have been motivated to use the system described by Montagna on a mobile Internet-connected computing device, as taught by Lewin. As I described above, and as the ’883 patent notes, at the time of the ’883 patent, mobile devices and applications were already becoming commonplace. *See* Section VI (Technology Background); Exhibit 1001 (’883 patent), 1:62-63 (“In the United States alone there are over 170 million registered cell phones.”), 2:12-14 (“the ability to play games and enjoy other activities corresponding to current events on cell phones exists”). Similarly, mobile devices were becoming a widely used medium

for gaming. *See* Section VI (Technology Background); Exhibit 1004 (Lewin), 1:12-24 (“The U.S. entertainment software industry is almost a \$7 billion a year industry, according to the most recent data released by the Interactive Digital Software Association (now the Entertainment Software Association). More than 221 million computer and video games are sold each year, which equates to almost two games for every household in America. The bulk of the entertainment software market share has historically been devoted to games written for personal computers. But in recent years the introduction of specialized video game consoles by companies such as SonyTM, MicrosoftTM and NintendoTM has caused console software to capture a greater percentage of the market share.”). Thus, a POSITA would have had plenty of motivation to combine teachings that relate to the delivery of games with teachings that relate to the delivery of games on mobile Internet-connected computing devices.

116. A POSITA would also have appreciated that users of the system described by Montagna would benefit from being able to play on a mobile Internet-connected computing device as described by Lewin. Exhibit 1003 (Montagna), [0101] (“The invention and all of the functional operations described herein can be implemented in digital electronic circuitry, or in computer hardware, firmware, software, or in combinations of them. The invention can be implemented as a computer program product, i.e., a computer program tangibly embodied in an

information carrier, e.g., in a machine-readable storage device or in a propagated signal, for execution by, or to control the operation of, data processing apparatus, e.g., a programmable processor, a computer, or multiple computers.”); Exhibit 1004 (Lewin), 16:22-26 (“[B]ecause the client is not limited to computer systems, user input can be received from electronic devices such as *mobile phones*, wireless handheld devices and cable set top boxes, to name a few.”), 17:10-16 (“And a third GUI player 82 written for a handheld wireless device or mobile phone may limit the presentation to a text message. GUI players 82 can thus modify the presentation of the system 10 to meet the specific hardware requirements of a user device. Used this way, the GUI players 82 allow the content distribution system 10 to support a broad range of user devices.”).

117. Lewin itself provides the motivation to implement Montagna’s system on mobile Internet-connected computing devices. Lewin describes that its system is beneficial because it allows a user to play the same content, regardless of the device used to access it. Exhibit 1004 (Lewin), 16:22-26 (“[B]ecause the client is not limited to computer systems, user input can be received from electronic devices such as mobile phones, wireless handheld devices and cable set top boxes, to name a few.”).

118. Further, combining the disclosures of Montagna and Lewin merely amounts to implementing known techniques to improve similar systems. The

content delivery system disclosed by Lewin provides a client application that manages the information sent to a user's device. The client application ensures that the device is able to execute the game that is transferred. Exhibit 1004 (Lewin), 4:29-33 ("Again, the system can be configured to work with a plurality of different end-user devices and the client application can be configured to detect one or more hardware capabilities of the end-user system and to conform the presentation of the linear and n-demand content to the system."). It is this aspect of Lewin that allows the system to operate on devices with limited available memory, including mobile Internet-connected computing devices. Exhibit 1004 (Lewin), 3:26-29, 10:46-49, 3:57-63, 16:22-26. A POSITA would have recognized that Montagna discloses a very similar system. Montagna discloses a "game client" that manages and minimizes the data that is being transmitted to the user device. Thus, implementing Montagna's system on a mobile Internet-connected computing device, as taught by Lewin, would merely require using a different, well known type of device to execute a known compatible system.

119. A POSITA would have implemented Montagna's system on a mobile device, as taught by Lewin, according to known methods and the combination would have yielded predictable results. A POSITA would have used basic and well-known computer programming principles to implement Montagna's system on a mobile

device. A POSITA would have possessed the knowledge and skillset to perform this implementation.

120. Finally, as I described in Section X.A.1.d)(a), Montagna and Lewin are from the same field, which is shared by the '883 patent, making them obvious to combine.

***(b) Reasonable expectation of success in
combining Montagna and Lewin***

121. A POSITA would also have a reasonable expectation of success in combining Montagna's system with a mobile Internet-connected computing device, as described in Lewin. As I described above, Montagna and Lewin both teach a system for delivering content to a user's device. Both Montagna and Lewin describe systems of delivering content that can be used on any type of device. Exhibit 1004 (Lewin), 17:3-16 ("The choice of GUI player 82 determines the type of presentation that the user receives. Thus, a first GUI player 82 that is written to support a high-end computer system may produce a presentation that shows a user interface and host avatar as a full-screen, dynamic, three-dimensional image. While a second GUI player 82 that is written for a lower-end computer system may produce a two-dimensional presentation of the interface and avatar. And a third GUI player 82 written for a handheld wireless device or mobile phone may limit the presentation to a text message. GUI players 82 can thus modify the presentation of the system 10

to meet the specific hardware requirements of a user device. Used this way, the GUI players 82 allow the content distribution system 10 to Support a broad range of user devices.”); Exhibit 1003 (Montagna), [0101] (“The invention and all of the functional operations described herein can be implemented in digital electronic circuitry, or in computer hardware, firmware, software, or in combinations of them. The invention can be implemented as a computer program product, i.e., a computer program tangibly embodied in an information carrier, e.g., in a machine-readable storage device or in a propagated signal, for execution by, or to control the operation of, data processing apparatus, e.g., a programmable processor, a computer, or multiple computers”). Thus, a POSITA would readily be able to modify Montagna’s system to operate on a mobile device, as taught by Lewin. As I discussed above, the systems required to implement the methods described in Montagna and Lewin would have been known to a POSITA as of 2005. The ’883 patent does not provide an improvement of circuitry or processing of the time, and thus, the circuitry and processing would have been available to users of the system of Montagna and Lewin. It would thus have been obvious, when presented with the teaching of a mobile device of Lewin, to combine those teachings with the system taught by Montagna.

e) 20[d]: “receiving additional information based on a user’s execution of the application on the mobile Internet-connected computing device;”

122. Montagna alone, or in combination with Lewin, renders obvious that additional information is received at the server based on the user’s execution of the application on the user device.

123. Montagna discloses a method that includes receiving additional information based on the user’s execution of the game client 304 (i.e., the application). For example, in Montagna’s system, the server 314 receives a selection as to whether the user would like to launch their in-progress-game. Exhibit 1003 (Montagna), [0086] (“When the user clicks the OK button, the game client 304 loads the incomplete game, allowing the user to finish the game.”). As I discussed above, if the user attempted to begin a new game when they already had a game in progress, they would be presented with an error screen (i.e., a set of selection information). Montagna’s game client 304 collects additional information based on a user’s continued execution of the application: a selection from the user that indicates they would like to launch the game that they already have in progress. This additional information is received by the server 314 before the game client 304 receives the in-progress game from the database 310, which resides on the server 314. Exhibit 1003 (Montagna), Figure 3, [0093] (“The INTERFACESTATE element is the data exchanged between client and server.”); Table 2 (“INTERFACESTATE:

Determined by game state. May be initial game state determined by XML Game Description or saved game state retrieve[d] from database.”).

124. Montagna also discloses that, once the user begins playing a game within the application, additional input is collected, such as drawing cards or placing bets during a card game. Exhibit 1003 (Montagna), [0051] (“Game client 304 collects user input (such as a wager) and communicates it back to the server-side components of the game platform 300”), [0058] (“Game client 304 then generates a user interface for the game and the user can initiate game play.”), [0093] (“The DATAEXCHANGE element defines what functions the Flash Game Loader should perform in each game state. For example, in poker, a game state can be the ‘draw’ state in which a user can request cards.”), [0089] (“Once the user interface has been assembled, the user can then initiate the game by providing some input to the game manager 306 through the user interface (step 714).”), [0011], [0048], [0052], [0056], [0057]. Referring to Montagna’s Figure 3, annotated below, the execution of the application by the user is performed at the **game client 304** on **client terminal 316** and received at the **server 314**. *See, e.g.*, Exhibit 1003 (Montagna), [0051] (“Game client 304 collects user input (such as a wager) and communicates it back to the server-side components of the game platform 300.”).

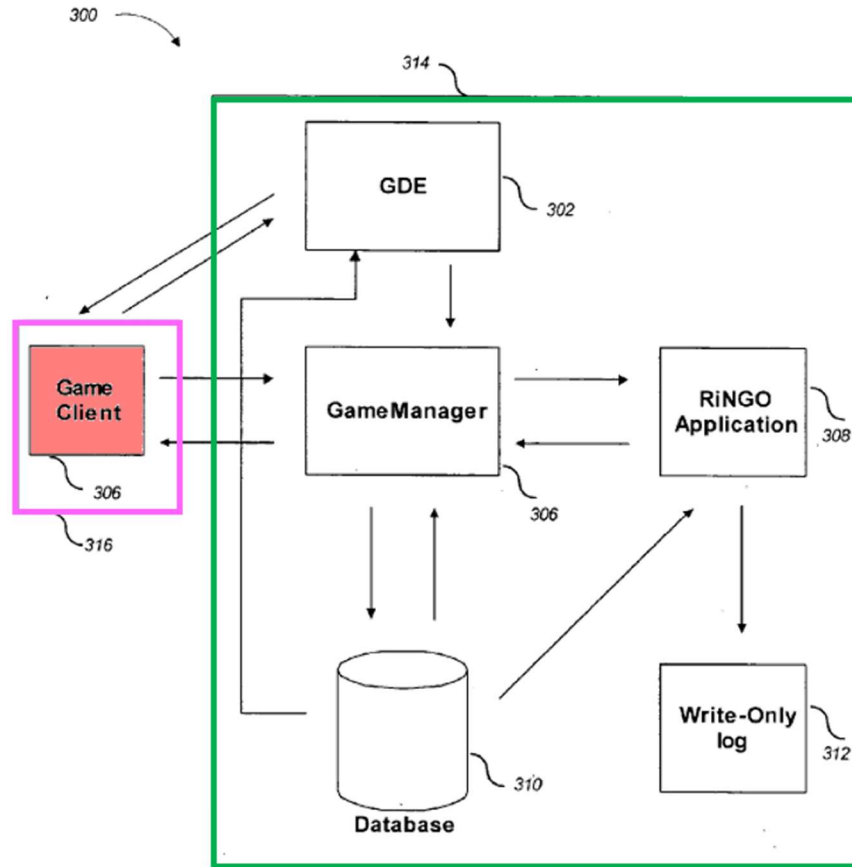


Exhibit 1003 (Montagna), Figure 3 (annotated)

125. Lewin also teaches additional information based on the user's execution of an application on a mobile Internet-connected computing device. As I discussed for element 20[c], Lewin teaches transitional content sent prior to a game being played, for example "mini-games, tips, game cheats, or game instructions." Exhibit 1004 (Lewin), 11:38-42. Lewin's transitional content collects user input. *Id.*, 10:3-6 ("If the requested content loads completely before the end of the commercial, **users can choose** to watch the rest of the commercial (or other transitional content) or to proceed to the game."). As I discussed in Sections

X.A.1.d).i.(a) and X.A.1.d).i.(b), a POSITA would have been motivated to combine Lewin's teaching of transitional content with Montagna's system and would have had a reasonable expectation of success in doing so.

126. As I also discussed above for limitation 20[c], if Montagna is found to not teach a mobile Internet-connected computing device, a POSITA would have found it obvious to implement Montagna's system on a mobile Internet-connected computing device, as described in Lewin. *See* Section X.A.1.d).iii.

f) 20[e]: “transmitting a list of assets necessary for executing the application, wherein the assets necessary for executing the application directly affect the operation of the application; and”

127. Montagna discloses this limitation. As I discussed above, Montagna discloses that the server 314 receives a selection of a game that a user would like to play. For the user to play the game, the client terminal 316 must first assemble the necessary assets to build the game interface. As I will describe next, Montagna discloses: a list of assets (UI definition), that the assets described by the list are necessary for executing the application, and directly affect the operation of the application, and the list is transmitted to the user's device.

i. “transmitting a list of assets”

128. Montagna describes a UI definition, which is a list of assets. Montagna describes that, after a user selects a game they would like to play from the game menu (i.e., the set of service related information), the selected game interface must

be generated on user's device. *See, e.g.*, Exhibit 1003 (Montagna), [0009] ("The system can provide a user interface that can be personalized for each user."). To determine the assets necessary for the game, a list of resources necessary for the activity is assembled by the server. Exhibit 1003 (Montagna), [0007] ("The product includes instructions to identify resources for inclusion in the user interface based on the customization data."). This list of resources is called a User Interface ("UI") definition. Exhibit 1003 (Montagna), [0029] ("The user interface can be generated from a user interface definition ('UI definition'), which can be implemented as an XML document.... The programmatic component can create the UI definition in real time in response to a user request. Moreover, the programmatic component can create UI definitions that are personalized for each user request."), [0074] ("After the user selects a game title and denomination from the game menu, GDE 302 sends to the loader (which, in the instant implementation is a Flash Game Loader) a list of all the game assets needed to display the game user interface."), [0029] ("In response to a user request and in real time, definition engine 108 creates a UI definition based on UI data 102, user interface resource list 104, and UI rules 106.").

129. Montagna's UI definition describes all the assets that need to be retrieved and assembled to build the gaming interface at the user's device. Exhibit 1003 (Montagna), [0060] ("GDE 302 dynamically builds, in real time, a UI definition from which the loader can execute to request the appropriate resources

(e.g., games assets) and generate a user interface.”), [0057] (“GDE 302 builds and sends to game client 304 a UI definition based on specific user information and configurable criteria as described above with reference to FIGS. 1 and 2. The UI definition includes a list of selected resources.”), [0034] (“Resource list 104 describes resources for generating components of user interfaces. Components of user interfaces can include Static user interface objects, dynamic user interface objects, or both. Resource list 104 can include a collection of disparate user interface instructions, referred to in the instant specification as user interface metadata. The user interface metadata can, for example, include any combination of the names of user interface objects, a position of objects in the user interface; the user interface objects required controls, functions, and physical appearance (for example, colors); as well as other descriptive identifiers. The user interface metadata can be implemented as XML elements stored in a relational database.”), [0035] (“User interface components can be heterogeneous objects of any type, number, or property. User interface components can be stateful and include, for example, buttons, images, or executable code.”), [0014], [0029], [0030], [0032], [0033], [0036], [0038], [0039], [0042], [0043], [0044], [0058], [0061], [0064], [0074], [0076], [0077], [0078].

130. The UI definition describes the asset that must be assembled, but itself does not provide the necessary assets for execution, just like the list of assets that the

'883 patent describes. *See* Exhibit 1003 (Montagna), [0029] (“The user interface can be generated from a user interface definition (‘UI definition’), which can be implemented as an XML document”), [0036] (“Rules 106 can specify, for example, that given a particular user and a particular enterprise, the UI definition should include instructions that identify certain user interface resources such that the user interface will include components generated by the resources.”), [0040] (“A user interface definition interpreter (‘UI interpreter’)... interprets a UI definition to construct a user interface 114”; “The UI interpreter 112 locates resources (e.g., resources 116) in order to assemble the user interface.”); Exhibit 1001 ('883 patent), 4:34-36 (“A list of assets are utilized to determine which assets are downloaded to the activity client. Only a set of necessary assets are downloaded to the activity client.”), 4:65-67 (“The server device sends only assets in the list of assets that are not in the list of client assets for each client device of the client devices.”), 9:41-46 (“The server having previously generated the manifest for the event, then compares the list of resident assets sent with the request in the step 302, and then downloads a single packet customized for the client containing only the assets missing on the client, in the step 304.”), 9:18-20 (“FIG. 2 illustrates a flowchart of an embodiment of implementing a manifest, which is a list of assets, to determine needed assets for a client computing device.”), 9:26-34 (“In the step 204, the client then compares the received manifest to the assets currently resident in the computing device. The client

then requests from the server the assets it needs to have resident in the client to obtain a full compliment of required software necessary to participate in the activity, in the step 206. The server then sends a package containing only the assets necessary to complete the manifest resident on the client computing device in the step 208.”).

131. Montagna’s Table 2, reproduced below, includes the elements that can be included in its UI Definition. Exhibit 1003 (Montagna), [0076] (“GDE 302 passes the list obtained from the database query along to the Flash Game Loader in the ASSETS element of the UI definition (an XML document that defines the user interface for a requested game, referred to as a Game Definition XML).”), [0079] (“Table 2 provides an example of high-level elements of a Game Definition XML and, furthermore, describes the source of each of the high-level elements.”), [0080].

TABLE 2

High-Level Element	Source
PAYTABLE	Built from XML Game Description.
ASSETS	Stored in database. A unique set of assets is available for every casino, game title, and language.
HELP	Contains URL for an HTML game help page.
ERRORS	Stored in database. A unique set of assets is available for every casino, game title, and language.
CONTROLDEF	Stored in database. A unique set of assets is available for every casino, game title, and language.
DATAEXCHANGE	Constant for each game title.
INTERFACESTATE	Determined by game state. May be initial game state determined by XML Game Description or saved game state retrieve from database. A user is always presented with their last game state, even if it describes a complete game.
USERPREFS	Determined by saved user preferences
JACKPOTSTATE	Determined both by dynamic values and static data progressive game and jackpot instance, update interval rate, and update request timeout period are obtained from the Progressive Display Manager.

Montagna, Table 2

132. Montagna describes that the UI definition (i.e., the list of assets) is transmitted from the **GDE 302**, which resides on the **server 314**, to the **game client 304**, which resides on the **client terminal 316** (i.e., a user device). Exhibit 1003 (Montagna), [0058] (“GDE 302 builds and sends to game client 304 a UI definition based on specific user information and configurable criteria as described above with reference to FIGS. 1 and 2. The UI definition includes a list of selected resources.”).

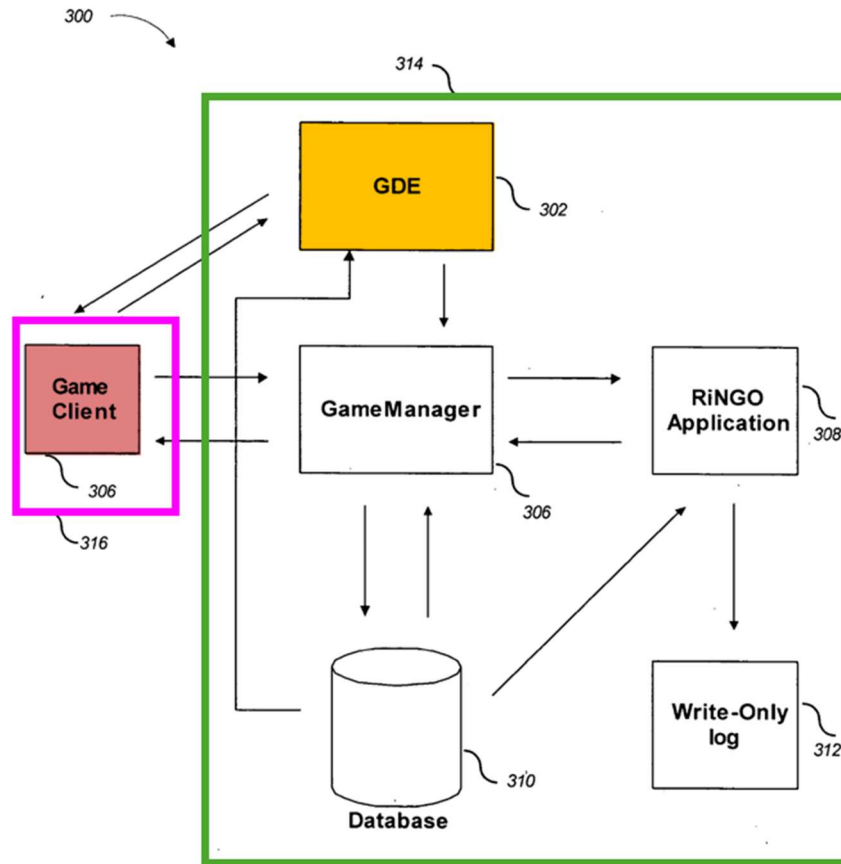


Exhibit 1003 (Montagna), Figure 3 (annotated)

- ii. ***“necessary for executing the application, wherein the assets necessary for executing the application directly affect the operation of the application;”***

133. Montagna describes that the assets listed in the UI definition are necessary for executing the game client (i.e., the application). Without the assets described by the UI definition, the user game interface could not be built. Exhibit 1003 (Montagna), [0077] (“The Game Definition XML describes what assets are to be included in the game and how the game UI works.”), [0046] (“Once the user interface component resources are successfully received, UI interpreter 112

continues the execution of the UI definition, which includes control definitions that specify the parameters by which the user interface components should be displayed on the users' display device.”), [0060] (“GDE 302 dynamically builds, in real time, a UI definition from which the loader can execute to request the appropriate resources (e.g., games assets) and generate a user interface.”), [0076] (“To determine the appropriate list of game assets, the GDE 302 queries a database for the list of game assets needed for a specific combination of casino, game title, and language.”).

134. As Montagna explains, the purpose of its method is to build a game interface that the user can interact with, and thus, the assets in the UI Definition are necessary to execute the game client (i.e., the application). Exhibit 1003 (Montagna), [0003] (“In general, applications that interact with users do so through a user interface...”), [0003] (“Casino style gaming applications (or simply, gaming applications) also provide graphical user interfaces, which typically provide users with a gaming environment for a game and allow users to provide input and receive output from the game.”), [0058] (“To initialize a new game, the user begins by clicking a hyperlink to select a game from a personalized game menu. The user's browser then loads game client 304. Game client 304 then requests a UI definition from GDE 302. GDE 302 builds and sends to game client 304 a UI definition based on specific user information and configurable criteria as described above with reference to FIGS. 1 and 2. The UI definition includes a list of selected resources.

After receiving the UI definition, game client 304 obtains resources necessary to generate a user interface for the selected game. Game client 304 then generates a user interface for the game and the user can initiate game play.”), [0077] (“The Game Definition XML describes what assets are to be included in the game and how the game UI works.”), [0088] (“Assembling the Game Definition XML includes retrieving necessary resources”), [0094] (“For example, the Flash Game Loader can read the ASSETS element contained with the Game Definition XML in order to determine which resources are needed to play the game.”).

135. Montagna also describes that the assets described by the UI definition directly affect the operation of the game client (i.e., the application). A POSITA reading the '883 patent would understand that, when the '883 patent describes that different assets correspond to different types of game or event (for example, the difference between soccer and football), these assets directly affect the operation of the application. Exhibit 1001 ('883 patent), 8:56-59, 6:7-17, 7:44-52. Similarly, Montagna describes that the assets described by the UI definition directly affect the operation of the application, because different assets are used to build different games. Exhibit 1003 (Montagna), [0004], [0075], [0061]. For example, an interface for an auction would operate differently than an interface for poker, just like an interface for Jeopardy would operate differently than an interface for NFL football. *See* Exhibit 1003 (Montagna), [0003] (“Applications that administer on-line

auctions, for example, provide graphical user interfaces through which one or more users can place bids and be presented with the results of their bids.... A graphical user interface for poker, for example, can include a representation of a gaming table and cards, buttons, and controls for choosing a wager and selecting cards....”); Exhibit 1001 (’883 patent), 8:56-59 (“Complexity is further compounded by the fact that the assets change in minor or major part from activity to activity. For example, from football to baseball to Jeopardy™, there are changes to these assets.”).

g) 20[f]: “transmitting only a second set of assets within the list of assets that are not already resident on the mobile Internet-connected computing device.”

136. Montagna alone, or in combination with Lewin, teaches this limitation.

137. Montagna describes a second set or portion of the assets described by UI definition (i.e., the list of assets) that are not already present on the user device. After the UI definition is sent to game client (i.e., the application), the instance of the game must be built for the user. Exhibit 1003 (Montagna), [0058] (“After receiving the UI definition, game client 304 obtains resources necessary to generate a user interface for the selected game. Game client 304 then generates a user interface for the game and the user can initiate game play.”). To do so, a User Interface interpreter (“UI Interpreter”) 112 is used to locate the resources that make up the game. Exhibit 1003 (Montagna), [0045] (“In the instant example, the UI definition indicates that several user interface components are available at a certain

location on a network accessible Web server. The UI interpreter 112 uses internal programmatic logic (e.g., provided by the above mentioned script) to make HTTP requests for the resources needed to generate user interface components.”).

138. Before requesting the necessary resources (i.e., assets), Montagna describes that the UI interpreter 112 determines what resources are already resident on the user device, for example, in the local cached memory of the device. Exhibit 1003 (Montagna), [0045] (“[C]ertain resources can be available in local cached memory resulting from previous personalized user interfaces. UI interpreter 112 will not have to retrieve these resources. The UI interpreter 112 can use its own internal programmatic logic to search the local cached memory for available resources of user interface components.”), [0089] (“In another implementation, the Flash Game Loader first checks for locally cached resources before requesting a download of the resources from the gaming Server. If a cached resource exists, it is retrieved from the cache.”), cl. 16 (“The method of claim 1, further comprising: checking a local cached memory for any of the identified resources.”), [0094] (“The Flash Game Loader checks local cached memory for the presence of any of the resources specified in the Game Definition XML (step 806).”). As shown in Montagna’s Figure 4, annotated below, the **Cached Game UI Resources** are located on the **user device**, along with the **UI Definition Interpreter** (shown in grey). Exhibit 1003 (Montagna), Figure 4, [0055].

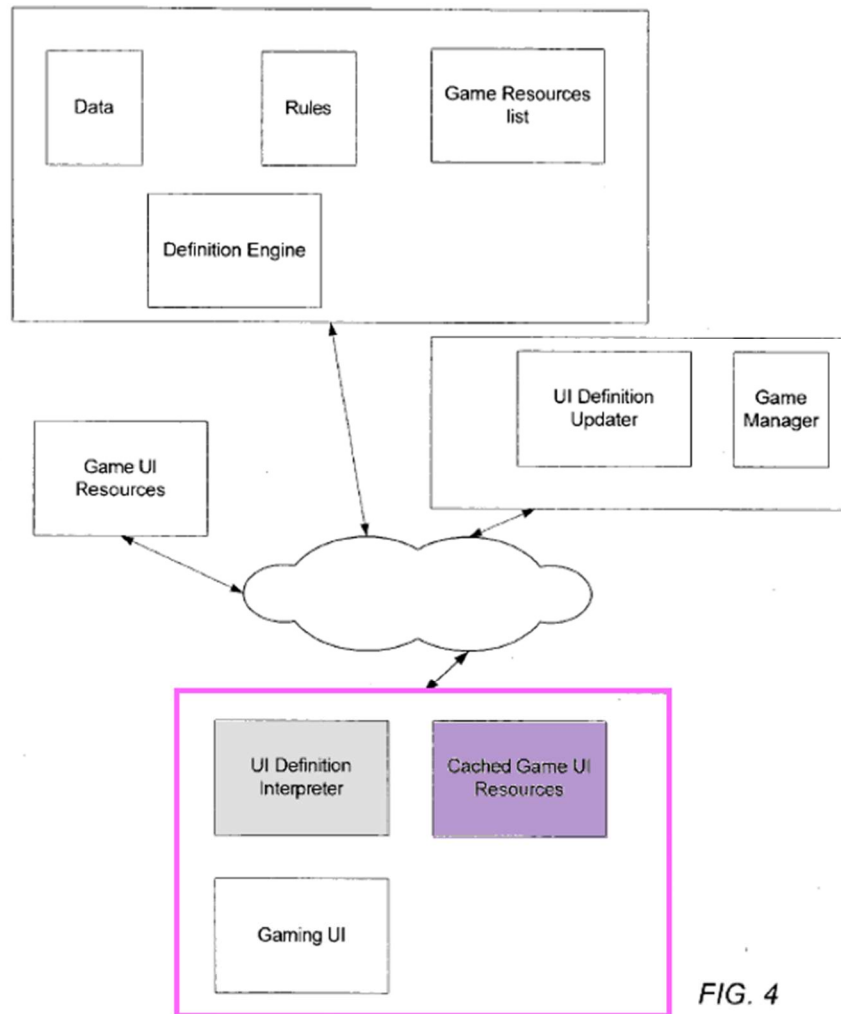


Exhibit 1003 (Montagna), Figure 4 (annotated)

139. Because the **Cached Game UI Resources** are already present on the device, Montagna states that the system does not have to request them from the server. Exhibit 1003 (Montagna), [0094] (“The Flash Game Loader checks local cached memory for the presence of any of the resources specified in the Game Definition XML (step 806). The Flash Game Loader retrieves the required resources *not located in local memory* (step 808).”), [0089] (“In another implementation, the

Flash Game Loader first checks for locally cached resources *before requesting a download of the resources from the gaming server*”), [0014] (“Furthermore, resources from previous game sessions can be reused in a current game session and the time needed to download resources to a client terminal is thus reduced.”), [0094] (“The Flash Game Loader checks local cached memory for the presence of any of the resources specified in the Game Definition XML (step 806). The Flash Game Loader retrieves the required resources *not located in local memory* (step 808).”); *see also* Exhibit 1001 (’883 patent), 4:34-36 (“A list of assets are utilized to determine which assets are downloaded to the activity client. Only a set of necessary assets are downloaded to the activity client.”), 4:65-67 (“The server device sends only assets in the list of assets that are not in the list of client assets for each client device of the client devices.”), 9:41-46 (“The server having previously generated the manifest for the event, then compares the list of resident assets sent with the request in the step 302, and then downloads a single packet customized for the client containing only the assets missing on the client, in the step 304.”), 9:18-20 (“FIG. 2 illustrates a flowchart of an embodiment of implementing a manifest, which is a list of assets, to determine needed assets for a client computing device.”), 9:26-34 (“In the step 204, the client then compares the received manifest to the assets currently resident in the computing device. The client then requests from the server the assets it needs to have resident in the client to obtain a full compliment of required software

necessary to participate in the activity, in the step 206. The server then sends a package containing only the assets necessary to complete the manifest resident on the client computing device in the step 208.”).

140. After determining what assets are already locally stored, Montagna describes transmitting only the second set of assets (i.e., the assets that are required to complete the requested game) to the user device. Exhibit 1003 (Montagna), [0094] (“The Flash Game Loader retrieves the required resources not located in local memory (step 808).”). These assets are located on a server, and are transmitted to the user device after being requested. Exhibit 1003 (Montagna), [0040] (“The UI interpreter 112 locates resources (e.g., resources 116) in order to assemble the user interface. The UI interpreter 112 can request resources from an external server or can retrieve the resources locally.”), [0045] (“In the instant example, the UI definition indicates that several user interface components are available at a certain location on a network accessible Web server. The UI interpreter 112 uses internal programmatic logic (e.g., provided by the above mentioned script) to make HTTP requests for the resources needed to generate user interface components.”), [0006], [0007], [0045], [0058], [0061], [0064], [0089].

141. Once the necessary missing assets are retrieved, the UI interpreter then assembles the user interface, allowing the user to play the selected game. Exhibit 1003 (Montagna), [0046] (“Once the user interface component resources are

successfully received, UI interpreter 112 continues the execution of the UI definition, which includes control definitions that specify the parameters by which the user interface components should be displayed on the users' display device.”).

142. Elements 20[e] and 20[f] are clearly disclosed by Montagna's Figure 8 below. At step 802, the user device receives the Game Definition, which it describes as a version of the UI definition (i.e., the list of assets). At step 806, the user device determines what assets are in cache and thus not already resident on the device. At step 808, the user device retrieves resources not located in cache; i.e., only a second set of assets not already resident on the device are retrieved or transmitted from the server to the user device.

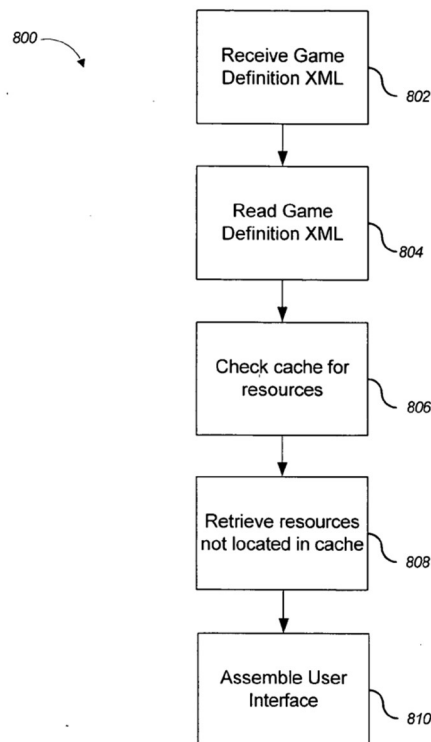


Exhibit 1003 (Montagna), Figure 8

143. As I discussed above for limitation 20[c], if Montagna is found to not teach a mobile Internet-connected computing device, a POSITA would have found it obvious to implement Montagna's system on a mobile Internet-connected computing device, as described in Lewin. *See* Section X.A.1.d).iii.

144. In my opinion, Montagna alone, or in combination with Lewin, renders obvious independent claim 20.

2. Dependent Claim 21

“The method of claim 20 wherein transmitting only the second set of assets within the list of assets that are not already resident on the mobile Internet-connected computing device is by determining a last activity participated in by the user or a genre of the last activity participated in by the user.”

145. Montagna alone, or in combination with Lewin, renders obvious claim 20, as I discussed above. Montagna alone, or in combination with Lewin, also renders obvious the additional limitation of claim 21 (transmitting only the second set of assets within the list of assets that are not already resident on the mobile Internet-connected computing device is by determining a last activity participated in by the user or a genre of the last activity participated in by the user).

146. Montagna describes that assets from previous game sessions (i.e., activities participated in) of the user can be used to determine the second set of assets. Exhibit 1003 (Montagna), [0014] (“Furthermore, resources from previous game sessions can be reused in a current game session and the time needed to

download resources to a client terminal is thus reduced”). By determining the last activity participated in by the user, the second set of assets do not need to include the assets already resident on the user device. *Compare id., with* Exhibit 1001 (’883 patent), 8:59-63 (“Some cellular phones are able to retain some or all of the assets in the phone’s memory if the activity to be played was the last activity partaken by a user. Other phones retain some or part of the assets which are common to different genres of activities.”).

147. As I discussed above for limitation 20[c], if Montagna is found to not teach a mobile Internet-connected computing device, a POSITA would have found it obvious to implement Montagna’s system on a mobile Internet-connected computing device, as described in Lewin. *See* Section X.A.1.d).iii.

148. Thus, Montagna alone, or in combination with Lewin, renders obvious dependent claim 21.

3. Dependent Claim 22

“The method of claim 20 further comprising comparing the list of assets with a first set of assets on the mobile Internet-connected computing device.”

149. Montagna alone, or in combination with Lewin, renders obvious claim 20, as I discussed above. Montagna alone, or in combination with Lewin, also renders obvious the additional limitation of claim 22 (comparing the list of assets with a first set of assets on the mobile Internet-connected computing device).

150. As I discussed in detail above, Montagna teaches that the UI definition (i.e., the list of assets) is compared against a first set of assets already resident on the device (the locally cached assets). *See also* Exhibit 1003 (Montagna), [0045] (“[T]he UI definition indicates that several user interface components are available at a certain location on a network accessible Web server. The UI interpreter 112 uses internal programmatic logic (e.g., provided by the above mentioned script) to make HTTP requests for the resources needed to generate user interface components. Optionally, certain resources can be available in local cached memory resulting from previous personalized user interfaces. UI interpreter 112 will not have to retrieve these resources.”). Without comparing the UI definition (list of assets) with a first set of assets on the device (the locally cached assets), the system would not know what assets to transmit. Thus, Montagna teaches comparing the list of assets (UI definition) with a first set of assets (locally cached assets) on a device.

151. As I discussed above for limitation 20[c], if Montagna is found to not teach a mobile Internet-connected computing device, a POSITA would have found it obvious to implement Montagna’s system on a mobile Internet-connected computing device, as described in Lewin. *See* Section X.A.1.d).iii.

152. Thus, Montagna alone, or in combination with Lewin, renders obvious dependent claim 22.

4. Dependent Claim 23

“The method of claim 20 further comprising comparing the list of assets with a first set of assets on the server.”

153. Montagna alone, or in combination with Lewin, renders obvious claim 20, as I discussed above. Montagna teaches the additional limitation of claim 23 (comparing the list of assets with a first set of assets on the server). Montagna discloses that the UI definition (i.e., the list of assets) is used to retrieve necessary assets to build a game. Exhibit 1003 (Montagna), [0031] (“The mentioned assembling process can include retrieving user interface components from a networked resource, for example, a Web server or Networked File Structure.”), [0040] (“The UI interpreter 112 locates resources (e.g., resources 116) in order to assemble the user interface.”), [0031]. Montagna describes that, to retrieve the necessary assets, the list of assets (i.e., UI definition) is compared against the assets that are available at various locations across the server, and the available assets are retrieved. Exhibit 1003 (Montagna), [0045] (“In the instant example, the UI definition indicates that several user interface components are available at a certain location on a network accessible Web server. The UI interpreter 112 uses internal programmatic logic (e.g., provided by the above mentioned script) to make HTTP requests for the resources needed to generate user interface components.”).

154. A POSITA would understand that to retrieve the necessary missing assets from the server, the UI Definition would need to be compared with the assets

on the server. Like a shopping list, the UI Definition contains the unique components for a given instance of the game. To retrieve those components on the list, the list is compared with the assets that are available on the server to determine what to retrieve.

155. Thus, Montagna alone, or in combination with Lewin, renders obvious dependent claim 23.

5. Dependent Claim 26

“The method of claim 20 wherein the geographic location of the mobile Internet-connected computing device determines the set of service related information to be transmitted, wherein the set of service related information includes service related information which is relative to and dependent on the geographic location of the mobile Internet-connected computing device.”

156. Montagna alone, or in combination with Lewin, renders obvious claim 20, as I discussed above. Montagna alone, or in combination with Lewin, also renders obvious the additional limitations of claim 26 (the geographic location of the mobile Internet-connected computing device determines the set of service related information to be transmitted, wherein the set of service related information includes information that is relative and dependent on the geographic location of the mobile Internet-connected computing device).

i. “The method of claim 20 wherein the geographic location of the mobile Internet-connected

***computing device determines the set of service
related information to be transmitted”***

157. Montagna discloses customizing a game menu (i.e., a set of service related information) based on the geographic location of the user device. Exhibit 1003 (Montagna), [0008] (“Personal data can indicate, for example, a jurisdiction of law to which a user is subject, a geographic location where the user has been verified to be currently located, the user's place of residence, and any combination of the aforementioned. Configurable criteria can specify, for example, which online games or which options are available to a user based on where the user resides and, furthermore, the jurisdiction of law to which the user is subject.”), [0009] (“The system can, for example, provide to a user in the United Kingdom a user interface that includes only gaming options that comply with the laws of the United Kingdom. The system does not include, in the user interface, gaming options that are available in other jurisdictions but illegal in the United Kingdom.”), [0061] (“The user’s profile indicates the user’s currency preference is, for example, United Kingdom Pounds and language preference is, for example, English. GDE 302 accesses user specific information and rules similar to the above-described UI rules to determine which games are available to the user.”), [0087] (“Based on the existing user profile, the user can be provided with a game menu tailored to the user preferences. The game menu can include a number of different games and denominations from which

the user can select. The availability of games can vary depending on the user profile. For example, some jurisdictions may allow some games to be played through online casinos and not others, thus the game menu can vary depending on the location of the user.”), [0010].

158. As I discussed above for limitation 20[c], if Montagna is found to not teach a mobile Internet-connected computing device, a POSITA would have found it obvious to implement Montagna’s system on a mobile Internet-connected computing device, as described in Lewin. *See* Section X.A.1.d).iii.

ii. “wherein the set of service related information includes service related information which is relative to and dependent on the geographic location of the mobile Internet-connected computing device.”

159. Montagna alone, or in combination with Lewin, also describes that a portion of the service related game menu includes service related information which is relative to and dependent on the geographic location of the mobile Internet-connected computing device. Montagna describes that, in the set of service related information (i.e., game menu), there are certain options that are relative to and depend on the geographic location of the user device. For example, Montagna teaches that depending on the laws where the user device is located, certain games may or may not be included in the game menu. Exhibit 1003 (Montagna), [0009], [0010], [0033], [0036], [0043], [0044], [0090]. Montagna describes that, if the user

changes the geographic location of the user device from a first to a second request, the game menus (i.e., set of service related information), which depend on that geographic location, will change. Exhibit 1003 (Montagna), [0010].

160. As I discussed above for limitation 20[c], if Montagna is found to not teach a mobile Internet-connected computing device, a POSITA would have found it obvious to implement Montagna's system on a mobile Internet-connected computing device, as described in Lewin. *See* Section X.A.1.d).iii.

161. Thus, Montagna alone, or in combination with Lewin, renders obvious dependent claim 26.

6. Dependent Claim 27

“The method of claim 26 wherein as the geographic location of the mobile Internet-connected computing device changes, different service related information is presented within the application based on the geographic location.”

162. Montagna alone, or in combination with Lewin, renders obvious claim 20, as I discussed above. Montagna alone, or in combination with Lewin, also renders obvious the additional limitation of claim 27 (as the geographic location of the mobile Internet-connected computing device changes, different service related information is presented within the application based on the geographic location).

163. Montagna discloses that as the geographic location of the user device changes, different game menu options (i.e., service related information) are presented. Exhibit 1003 (Montagna), [0009], [0010] (“The system can provide a

user interface that can be personalized for each user. Personalization can be based on, for example, laws to which a specific user is subject.... The system can, for example, provide to a user in the United Kingdom a user interface that includes only gaming options that comply with the laws of the United Kingdom.... If there are changes that occur between a first request and a second request, for example, the specific user changes residency from the United Kingdom to a jurisdiction that permit the above-mentioned illegal gaming options, the system can make the appropriate changes during its personalization process.”), [0107] (“[T]he personalized user interface can prevent, for example, auctions of items that are illegal in a particular jurisdiction while allowing the same items to be auctioned in jurisdiction where they are legal.”).

164. As I discussed above for limitation 20[c], if Montagna is found to not teach a mobile Internet-connected computing device, a POSITA would have found it obvious to implement Montagna’s system on a mobile Internet-connected computing device, as described in Lewin. *See* Section X.A.1.d).iii.

165. Thus, Montagna alone, or in combination with Lewin, renders obvious dependent claim 27.

7. Dependent Claim 28

“The method of claim 27 wherein the geographic location of the mobile Internet-connected computing device is determined

using one or more cellular networks, GPS, GSM or other related technologies.”

166. Montagna alone, or in combination with Lewin, renders obvious claim 20, as I discussed above. Montagna alone, or in combination with Lewin, also renders obvious the additional limitation of claim 28 (the geographic location of the mobile Internet-connected computing device is determined using one or more cellular networks, GPS, GSM, or other related technologies).

167. As I discussed above, Montagna discloses that the location of the user determines the game options that are presented to the user. Montagna further describes that the location of the user can be determined by a variety of means, including geographic location where the user is presently located. Exhibit 1003 (Montagna), [0008] (“Personal data can indicate, for example, a jurisdiction of law to which a user is subject, a geographic location where the user has been verified to be currently located, the user's place of residence, and any combination of the aforementioned. Configurable criteria can specify, for example, which online games or which options are available to a user based on where the user resides and, furthermore, the jurisdiction of law to which the user is subject.”). For methods to determine the user’s location, Montagna looks to U.S. Patent Application No. 10/280,895 (issued as U.S. Patent No. 6,853,973; Mathews (Exhibit 1007)), which Montagna incorporates by reference in its entirety. Exhibit 1003 (Montagna), [0091]

(“The verification module is generally an application that verifies, among other things, a user's identity and place of residence. ... The verification module is further described in commonly assigned U.S. patent application Ser. No. 10/280,895, which is hereby incorporated by reference in its entirety.”). Mathews specifies that the geographic location of the user can be determined by GPS. Exhibit 1007 (Mathews), 25:37-41 (“The verification module can use any data source providers and is not limited to the examples described. For example, the verification module can obtain longitude and latitude information from a data source that uses GPS to locate a user.”), 5:41-48, 8:21-37.

168. As I discussed above for limitation 20[c], if Montagna is found to not teach a mobile Internet-connected computing device, a POSITA would have found it obvious to implement Montagna's system on a mobile Internet-connected computing device, as described in Lewin. *See* Section X.A.1.d).iii.

169. Thus, Montagna alone, or in combination with Lewin, renders obvious dependent claim 28.

8. Dependent Claim 29

“The method of claim 20 wherein the information based on the geographic location utilizes past usage of a user in connection with the geographic location.”

170. Montagna alone, or in combination with Lewin, renders obvious claim 20, as I discussed above. Montagna discloses the additional limitation of claim 29

(the information based on the geographic location utilizes past usage of a user in connection with the geographic location).

171. As I explained above for claim 26, Montagna discloses a menu of game options (i.e., service related information) that is based on geographic location. Montagna further discloses that its menu of game options is personalized to a user based on location and previous actions. Montagna, [0009] (“The system can provide a user interface that can be personalized for each user. Personalization can be based on, for example, laws to which a specific user is subject, groups of which the specific user is a member, preferences of the specific user, and previous actions of the specific user. The system can, for example, provide to a user in the United Kingdom a user interface that includes only gaming options that comply with the laws of the United Kingdom.”).

172. Thus, Montagna alone, or in combination with Lewin, renders obvious dependent claim 29.

9. Dependent Claim 30

“The method of claim 20 wherein the server is located remotely from the mobile Internet-connected computing device.”

173. Montagna alone, or in combination with Lewin, renders obvious claim 20, as I discussed above. Montagna alone, or in combination with Lewin, also

renders obvious the additional limitation of claim 30 (the server is located remotely from the mobile Internet-connected computing device).

174. Montagna discloses and illustrates in Figure 4 (below) that the server is separate from the user device. Exhibit 1003 (Montagna), [0106] (“The computing system can include clients and servers. A client and server are generally remote from each other and typically interact through a communication network.”), [0105], *see also* [0019] (“FIG. 4 shows the game platform implemented over a network.”).

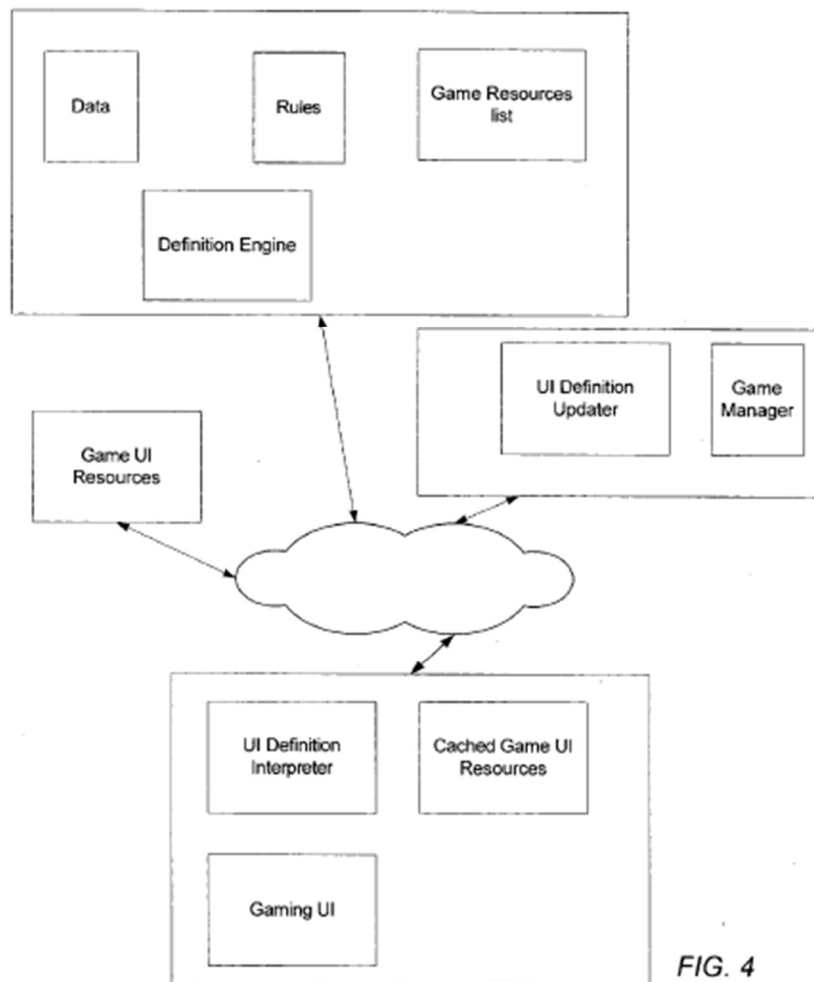


FIG. 4

Exhibit 1003 (Montagna), Figure 4

175. As I discussed above for limitation 20[c], if Montagna is found to not teach a mobile Internet-connected computing device, a POSITA would have found it obvious to implement Montagna's system on a mobile Internet-connected computing device, as described in Lewin. *See* Section X.A.1.d).iii.

176. Thus, Montagna alone, or in combination with Lewin, renders obvious dependent claim 30.

10. Dependent Claim 38

“The method of claim 20 wherein the assets are delivered based on a current state of the consumer service.”

177. Montagna alone, or in combination with Lewin, renders obvious claim 20, as I discussed above. Montagna discloses the additional limitation of claim 38 (the assets are delivered based on a current state of the consumer service).

178. In particular, Montagna discloses that the user game interface is generated based on the state of user interface components. Exhibit 1003 (Montagna), [0048] (“The UI definition can also include logic for determining the states of its user interface components. A user interface icon such as a button, for example, can have multiple states. Once the UI interpreter 112 has finished generating a user interface having components in their appropriate states, the user interface is ready to function without further input from the system 100”), [0078], [0093], [0096], [0097], [0098]. These user interface components make up the consumer service. Montagna also teaches that, when the user begins a new game,

they are presented a “beginning game state.” Exhibit 1003 (Montagna), [0056] (“In a typical gaming session, game platform 300 presents a user with a beginning game state, then updates the game state based on user input. The initial game state can be a new game. All subsequent game states reflect the random game outcome determined for a particular user. Game transactions move a user from one game state to the next. Most games consist of either two or three games states. The game platform facilitates the user's request for the next game state, logs each state determination, and updates the user's account as necessary after each state is reached.”). Therefore, the set of assets delivered would be different based on that game state.

179. Thus, Montagna alone, or in combination with Lewin, renders obvious claim 38.

11. Independent Claim 1

a) 1[preamble]: “A method of implementing a consumer service on a mobile Internet-connected computing device comprising:”

180. In my opinion, even if the preamble of independent claim 1 is considered limiting, Montagna alone, or in combination with Lewin, renders it obvious.

181. The preamble of claim 1 only differs from the preamble of claim 20 in that it specifies that the consumer service is on a mobile Internet-connected

computing device. For at least the reasons that I provided for the preamble of claim 20, Montagna discloses a method of implementing a consumer service.

182. As I discussed above for limitation 20[c], if Montagna is found to not teach a mobile Internet-connected computing device, a POSITA would have found it obvious to implement Montagna's system on a mobile Internet-connected computing device, as described in Lewin. *See* Section X.A.1.d).iii.

183. Thus, Montagna alone, or in combination with Lewin, renders obvious a method of implementing a consumer service on a mobile Internet-connected computing device, as recited by independent claim 1.

b) 1[a]: “loading a set of service related information located on a server to an activity client;”

184. The combination of Montagna and Lewin renders obvious loading a set of service related information located on a server to an activity client. Limitation 1[a] differs from limitation 20[a] in three ways: (i) it specifies that set of service related information is “loaded” rather than “transmitted”; (ii) it specifies that the set of service related information is located on a server; and (iii) it recites an “activity client” rather than an “application.” I discuss each of these differences below. For the reasons set forth for limitation 20[a] discussed above, as well as the additional reasons set forth below, the combination of Montagna and Lewin teaches this limitation.

i. “loading” versus “transmitting”

185. As I explained above for limitation 20[a], Montagna discloses that service related information (a game menu) is transmitted from the server to the user device. Once there, it is presented to the user. The distinction between “loading” a set of service related information (in limitation 1[a]) and “transmitting” a set of service related information (in claim 20[a]) is just a difference in perspective. Independent claim 20 is written from the perspective of the server, and thus the server “transmits” a set of service related information (i.e., a game menu) to the user device. Claim 1 is written from the perspective of the device, and thus the user device “loads” the same set of service related information (i.e., a game menu) received from the server. Thus, because Montagna describes that the set of service related information (game menu) is transmitted from the server to the user device, it is true that it also describes the same user device “loading” the service related information received from the server.

186. The same can be said of the distinction between “transmitting” and “downloading,” which is discussed later in the claims. A POSITA would understand “transmitting” to be a broad category, including a plethora of different means by which information gets from point A to point B. The act of downloading is merely a subset of this genus.

ii. service related information located on server

187. Limitation 1[a] also differs from limitation 20[a] because it states that the service related information is located on the server. As I explained for limitation 20[a], Montagna teaches transmitting service related information (i.e., a game menu). As I also explained, Montagna discloses that the game menu is generated by the GDE. The GDE resides on a server and, thus, the service related information (game menu) is located on a server before it is loaded to the user device. Exhibit 1003 (Montagna), [0049].

iii. “activity client” versus “application”

188. Lastly, limitation 1[a] differs from limitation 20[a] because it recites an “activity client” rather than an “application.” As I explained above for limitation 20[a], Montagna teaches an application. The activity client described in the ’883 patent is an application that allows other applications to be transmitted, stored, and executed on a user device. Exhibit 1001 (’883 patent), 6:30-32 (“The present invention utilizes a software application, referred to as an “Activity Client.” The “Activity Client” is retained in a user’s computing device’s memory”); 3:5-6 (“The activity client is a program for managing activity information and data.”); (“[T]he “Activity Client” has the ability to receive and hold downloaded activity-specific data instructions from the server.”); 3:25-27 (“An activity interpreter resident in the activity client is used to execute the application.”). This use of an activity client is

nothing new, but is merely a difference in implementation, and one that would have been well known in the art. Lewin discloses such an activity client.

189. Lewin discloses a “client application” which resides on a “computing device associated with the user that displays a listing of available games,” and includes a “plurality of emulators that are used to emulate the native platforms for which the plurality of games were originally intended to run.” Exhibit 1004 (Lewin), 3:26-33. An emulator is a particular type of application or system that allows a user device to run software programs or code that it was not initially intended to run.

190. Once a game is downloaded, it is the emulators within Lewin’s client application that allow the game to run. Exhibit 1004 (Lewin), 9:8-11 (“When the game file is delivered to a client application 15, an emulator software component in the client application 15 emulates the original platform to allow the game to play on 10 the user system much like it did on its original platform.”). By using the client application, Lewin’s system allows a variety of devices to run the same gaming application, including a mobile device. Exhibit 1004 (Lewin), 10:46-49 (“Additional users may run the client application 15 on a set top box of a cable or satellite system, a personal digital assistant (PDA) device, cell phone or other digital electronic device.”); 16:22-25 (“And because the client is not limited to computer systems, user input can be received from electronic devices such as mobile phones, wireless handheld devices and cable set top boxes, to name a few.”), 3:57-63.

191. The use of a client application (i.e., activity client) like that described in Lewin differs from the use of a single application to play a game (as described by Montagna), primarily in the amount of information that is required to download and execute a game. A system like that described in Montagna builds a different interface for each game within the application (i.e., game client), but the execution of the game still occurs directly within the game client. The use of a client application like that described by Lewin allows for a completely separate application to be downloaded and executed inside of the client application. Exhibit 1004 (Lewin), 3:4-11 (“With regard to the delivery of games, emulators execute game code that is written to run on various gaming platforms. A dynamic interface launches and manages emulators in a manner that is largely transparent to the user.”), 2:16-18, 9:53-63, 12:6-12, 22:24-35. The client application is specially designed to execute the original code of the other applications. Exhibit 1004 (Lewin), 3:29-36 (“In some described embodiments, the client application includes a plurality of emulators that are used to emulate the native platforms for which the plurality of games were originally intended to run. The emulators translate the native platform machine code blocks of the game code to functionally equivalent blocks of compiled instruction set code that run on the user device.”), 3:57-63. Lewin’s client application corresponds to the “activity center” claimed by the ’883 patent.

(a) Motivation to combine Montagna and Lewin

192. A POSITA would have been motivated to combine the client application described in Lewin with the system disclosed by Montagna. First, as I discussed above, the system taught by Montagna is adept at minimizing the information transfer and memory storage required when building a customized gaming interface (a consumer service). See Sections IX.A (Overview of Montagna) and X.A.1.f) (limitation 20[e]). The transfer-minimization benefits of Montagna's system would have motivated a POSITA to use it on larger game transfers, up to, and including entire game applications, as taught by Lewin. Lewin itself states that the transfer of entire applications can become unwieldy. Exhibit 1004 (Lewin), 9:59-63 ("In contrast, a medium-size game written for the Sony PlayStation console is about 100 megabytes in size, and a game written for a personal computer typically ranges between several hundred megabytes to several gigabytes of data."), 11:43-52 ("[T]he game content that is delivered to users can involve large amounts of data...."). Thus, a POSITA would have appreciated the benefit of pairing of the systems of Montagna and Lewin.

193. Lewin itself provides the motivation to apply Montagna's system to larger games. Lewin states that, by using the client application, a user is able to download applications that may not have been initially built for the user's device.

Exhibit 1004 (Lewin), 3:26-33 (“Additional embodiments of the game delivery system describe using a client application residing on a computing device associated with the user that displays a listing of available games accepts a user request to play a game. In some described embodiments, the client application includes a plurality of emulators that are used to emulate the native platforms for which the plurality of games were originally intended to run.”), 2:1-7 (“Emulation software (sometimes referred to herein as ‘emulators’) arose, in part, as a response to a perceived need among gamers [sic] for a cross-platform gaming capability. Generally, a software emulator is a computer program that runs on a target platform (often a personal computer) and uses software to supply native platform capabilities that are not present in the target platform.”). Thus, by using a client application as taught by Lewin in the system of Montagna, additional options become available to the user. A POSITA would have appreciated the benefit of such a change to Montagna’s system.

194. Finally, as I discussed in Section X.A.1.d).i.(a), Montagna and Lewin are from the same field, which is shared by the ’883 patent, providing a POSITA motivation to combine them.

***(b) Reasonable expectation of success in
combining Montagna and Lewin***

195. A POSITA would also have a reasonable expectation of success in using a client application as described in Lewin in a system as described in Montagna. Montagna describes a method where individual assets are used to build a user interface, including executable code. *See* Exhibit 1003 (Montagna), [0035]. These same principles can be used to build a complete application to operate within client application, as described in Lewin. As I discussed above, at the time of the '883 patent, a POSITA would have already been well versed in the practice of downloading applications and, thus, would have readily been able to apply the teachings of Montagna to the known practice of downloading applications to a device. *See* Section IV (Technology Background).

***c) 1[b]: “selecting an option from the activity client from a
list of available options;”***

196. The combination of Montagna and Lewin teaches selecting an option from the activity client. Limitation 1[b] only differs from limitation 20[b] in that it specifies “selecting an option from the activity client from the list of available options” rather than “receiving a selection related to the consumer service” (a difference in perspective from the server to the user device). As I discussed for limitation 20[b], Montagna discloses that the user of the game client selects an option from the game menu, which is a list of available options. Thus, when using client

application (as taught by Lewin; i.e., an activity client) instead of a game client in Montagna's system, a user selection would still occur.

197. Regardless, Lewin also describes the selection of an option (i.e., a game that the user would like to play) from the list of available options within its client application, and thus, when using a client application (i.e., activity client) rather than a game client (i.e., application), there is no difference in selecting an option. Exhibit 1004 (Lewin), 22:53-54 ("The entertainment content distribution system 10, which comprises an ordered listing of selectable services."), 3:4-7 ("An embodiment of the delivery system uses a graphical user interface and host avatar to introduce users to and allow them to select from available content."). Thus, the combination of Montagna and Lewin teaches this limitation 1[b].

d) 1[c]: "downloading a set of service-specific information related to a selected option from the server to the mobile Internet-connected computing device;"

198. Montagna alone, or in combination with Lewin, teaches downloading a set of service-specific information related to a selected option (game to play). Limitation 1[c] differs from limitation 20[c] because (i) it uses the term "service-specific information" rather than "selection information," and (ii) it uses the term "downloading" rather than "transmitting." I address these two differences below. Otherwise, limitation 1[c] is the similar to limitation 20[c], which, as I already explained above, is taught by Montagna alone, or in combination with Lewin.

i. “Service-specific information” versus “selection information”

199. As I explained for limitation 20[c], Montagna alone, or in combination with Lewin, teaches “selection information.” Montagna and Lewin also describes “service-specific information.” A POSITA would understand that the same selection information (i.e., error messages, game updates, and instructions) that Montagna and Lewin disclose is service-specific in that it applies only to the specific gaming chosen. As I discussed above, when the ’883 patent discusses “selection information” that is sent before the transmission of the assets that make up the game, the ’883 patent refers to “instructions and options” that are displayed following a user selection. *See* Exhibit 1001 (’883 patent), 11:57-63. Thus, when Montagna and Lewin discuss sending information that is related to the game selection, whether that be an error message, information that populates the selected game, or instructions about the game that the user has selected, such information correspond to both “selection information” and “service-specific information.” This makes sense, as any “selection information” about a user’s selection would be “service specific” in that it relates to the service that the user has selected. For example, a set of information about a selection for a game (for example, instructions to play Pacman or information about a game of Pacman in progress) would make little sense in the context of a different service, like restaurant recommendations.

ii. “downloading” versus “transmitting”

200. Montagna and Lewin each teach “downloading” service-specific information. As I discussed in limitation 20[c], transmitting is taught by Montagna and Lewin. *See* ¶ 107-111. As I discussed for limitation 1[a], “downloading” is a subset within “transmitting” and thus the disclosure of Montagna also includes “downloading” as included in limitation 1[c].

e) 1[d]: “executing an application related to the selected option within the activity client on the mobile Internet-connected computing device;”

201. The combination of Montagna and Lewin teaches executing an application related to the selected option within the activity client on the mobile Internet-connected computing device.

202. As I discussed above, by using a client application (i.e., activity client) that can download different applications, a user is able to download and execute complete games within the client application (activity center). As I also discussed above for limitation 1[a], it would have been obvious to combine the teachings of Montagna with the teachings of the system of Lewin that not only allows games to be played, but also allows for entire gaming applications to be downloaded and run within another application (i.e., client application or activity client). The client application of Lewin executes a game. Exhibit 1004 (Lewin), 9:8-11 (“When the game file is delivered to a client application 15, an emulator software component in

the client application 15 emulates the original platform to allow the game to play on 10 the user system much like it did on its original platform.”), 13:39-41 (“When that portion of the game required to start play has finished downloading, the game manager 90 launches a game player and play begins.”), 3:29-36 (“In some described embodiments, the client application includes a plurality of emulators that are used to emulate the native platforms for which the plurality of games were originally intended to run. The emulators translate the native platform machine code blocks of the game code to functionally equivalent blocks of compiled instruction set code that run on the user device.”).

203. As I discussed above for limitation 20[c], if Montagna is found to not teach a mobile Internet-connected computing device, a POSITA would have found it obvious to implement Montagna’s system on a mobile Internet-connected computing device, as described in Lewin. *See* Section X.A.1.d).iii.

f) 1[e]: “receiving a list of assets necessary for executing the application from the server, wherein the assets necessary for executing the application directly affect the operation of the application;”

204. The combination of Montagna and Lewin teaches this limitation. The difference between limitation 1[e] and limitation 20[e] is that limitation 1[e] recites “receiving” a list of assets, rather than “transmitting” a list of assets, as stated in limitation 20[e]. This is simply a difference in perspective—whether it is from the

perspective of the sender or recipient. For example, to receive a piece of mail, it must also be sent. When a piece of mail is sent from A to B, that means that it is also received at B from A. Similarly, transmitting a list of assets from a server also includes receipt of this list at the user device.

205. As I discussed for limitation 20[e], Montagna teaches transmitting a UI definition (i.e., a list of assets). Montagna's system transmits the UI definition from the server to the user device, where it is received. *See* Exhibit 1003 (Montagna), [0045] ("The definition engine 108 sends the UI definition across a network as an HTTP response object to the UI interpreter 112 (step 208), which executes on an end-user personal computer."). With respect to the difference between the use of a client application as taught by Lewin and the game client taught by Montagna, the only difference is that the assets would now pertain to the application (i.e., gaming application) to be played within the client application, rather than only the assets to build a game in the game client. As I discussed above, a POSITA would have been motivated to combine the teachings of Lewin to the teachings of Montagna to allow a user to download and play entire gaming applications, and would have a reasonable expectation of success in doing so. Thus, Montagna and Lewin teach this limitation.

g) 1[f]: “comparing the list of assets with a first set of assets on the mobile Internet-connected computing device; and”

206. Montagna alone, or in combination with Lewin, teaches this limitation. Limitation 1[f] recites the same limitations contained in dependent claim 22. Thus, as I explained with respect to claim 22, limitation 1[f] is taught by Montagna alone, or in combination with Lewin.

h) 1[g]: “receiving only a second set of assets within the list of assets that are not already resident on the mobile Internet-connected computing device.”

207. Montagna alone, or in combination with Lewin, teaches this limitation. Limitation 1[g] differs from limitation 20[f] only in that it recites *receiving* a second set of assets (1[g]) rather than *transmitting* a second set of assets (20[f]). As I discussed above for limitation 20[f], the second set of assets are *transmitted* from a server to the user device, where they are *received*, and thus, this difference is only one of perspective. Exhibit 1003 (Montagna), [0089] (“In another implementation, the Flash Game Loader first checks for locally cached resources before requesting a download of the resources from the gaming server.”). Thus, Montagna alone, or in combination with Lewin, teaches this limitation.

208. In my opinion, the combination of Montagna and Lewin renders obvious independent claim 1.

12. Dependent Claim 2

“The method of claim 1 wherein receiving only the second set of assets within the list of assets that are not already resident on the mobile Internet-connected computing device is by determining a last activity participated in by the user or a genre of the last activity participated in by the user.”

209. The combination of Montagna and Lewin renders obvious claim 1, as I discussed above. Montagna alone, or in combination with Lewin, renders obvious the additional limitation of claim 2. Claim 2 differs from claim 21 because it depends from claim 1 instead of claim 20, and it states “receiving” a second set of assets, rather than “transmitting” only a second set of assets. Aside from this difference in perspective, for at least the same reasons that I provided above for claim 21, the additional limitation of claim 2 is taught by Montagna alone, or in combination with Lewin.

210. Thus, the combination of Montagna and Lewin renders obvious claim 2.

13. Dependent Claim 5

“The method of claim 1 wherein a geographic location of the mobile Internet-connected computing device determines the set of service related information to be loaded, wherein the set of service related information includes service related information which is relative to and dependent on the

geographic location of the mobile Internet-connected computing device.”

211. The combination of Montagna and Lewin renders obvious claim 1, as I discussed above. Montagna alone, or in combination with Lewin, renders obvious the additional limitations of claim 5. Claim 5 recites the same limitations as claim 26, and only differs because it depends from claim 1 rather than claim 20, and recites “loaded” instead of “transmitted.” Aside from this difference in perspective, for at least the same reasons I provided above for claim 26, the additional limitations of claim 5 is also taught by Montagna alone, or in combination with Lewin.

212. Thus, the combination of Montagna and Lewin renders obvious claim 5.

14. Independent Claim 81

a) 81[preamble]: “A method of implementing a consumer service with a server comprising:”

213. In my opinion, even if the preamble of independent claim 81 is considered limiting, Montagna discloses it. The preamble of claim 81 is identical to the preamble of independent claim 20. Therefore, for the reasons that I provided for the preamble of claim 20, Montagna discloses the preamble of claim 81.

b) 81[a]: “transmitting a set of service related information to an application on a mobile Internet-connected computing device;”

214. Montagna alone, or in combination with Lewin, renders this limitation obvious. This limitation differs from limitation 20[a] only in that it states that the application is on a mobile Internet-connected computing device. The remaining language of this limitation is identical to limitation 20[a], and thus is disclosed by Montagna as I discussed above. As I discussed with respect to limitation 20[c], if Montagna is found to not teach a mobile Internet-connected computing device, a POSITA would have found it obvious to implement Montagna’s system on a mobile Internet-connected computing device, as described in Lewin. *See* Section X.A.1.d).iii. Thus, Montagna alone, or in combination with Lewin, renders obvious limitation 81[a].

c) 81[b]: “receiving additional information based on a user’s execution of the application on the mobile Internet-connected computing device;”

215. Montagna alone, or in combination with Lewin, renders this limitation obvious. Limitation 81[b] is identical to limitation 20[d], and thus is taught by Montagna alone, or in combination with Lewin, as I discussed above for limitation 20[d].

d) 81[c]: “transmitting a list of assets necessary to execute the application to the mobile Internet-connected computing device, wherein the assets necessary for executing the

***application directly affect the operation of the application;
and”***

216. Montagna alone, or in combination with Lewin, renders this limitation obvious. Limitation 81[c] differs from limitation 20[e], in that it states that the list of assets is transmitted to the mobile Internet-connected computing device. As I discussed for limitation 20[c], if Montagna is found to not teach a mobile Internet-connected computing device, a POSITA would have found it obvious to implement Montagna’s system on a mobile Internet-connected computing device, as described in Lewin. *See* Section X.A.1.d).iii. The remainder of this limitation is identical to limitation 20[e], and thus is taught by Montagna as I discussed above for limitation 20[e].

e) 81[d]: “transmitting to the mobile Internet-connected computing device only a second set of assets within the list of assets that are not already resident on the mobile Internet-connected computing device.”

217. This limitation is rendered obvious by Montagna alone, or in combination with Lewin. This limitation differs from limitation 20[f] in that it recites that the second set of assets are transmitted to the mobile Internet-connected computing device. As I discussed for limitation 20[f] above, Montagna alone, or in combination with Lewin, teaches that the second set of assets is sent to the application, which resides on the mobile Internet-connected computing device.

Thus, this limitation is similarly rendered obvious by Montagna alone, or in combination with Lewin.

218. In my opinion, Montagna alone, or in combination with Lewin, renders obvious independent claim 81.

15. Dependent Claims 6-9, 17, 82-84, 87-91, 99

219. Dependent claims 6-9, 17, 82-84, 87-91, and 99 only differ from the claims that I already discussed above because they depend from claim 1 or 81 rather than claim 20. The language of these claims is otherwise identical. Thus, for the reasons that I already provided above for claims that depend from claim 20, Montagna alone, or in combination with Lewin, renders obvious claims 6-9, 17, 82-84, 87-91, and 99 according to the mapping in the following table.

Prior discussed claim depending from claim 20	Corresponding claims depending from:	
	Claim 1	Claim 81
21	-	82
22	-	83
23	-	84
26	-	87
27	6	88
28	7	89
29	8	90
30	9	91
38	17	99

B. Ground 2: Claims 28, 7, and 89 are Obvious Over Montagna Alone or in View of Lewin, and Further in View of Mathews

220. For Ground 2, I considered Montagna alone, or in combination with Lewin, further combined with Mathews. My analysis of claims 28, 7, and 89 is similar to that in Ground 1, except that here, in Ground 2, I consider Mathews as a standalone reference, rather than as incorporated by reference in its entirety into Montagna.

1. Dependent Claim 28

“The method of claim 27 wherein the geographic location of the mobile Internet-connected computing device is determined using one or more cellular networks, GPS, GSM or other related technologies.”

221. As I explained for claim 28 in Ground 1, Montagna alone, or in combination with Lewin, discloses using GPS or other related technologies to determine the geographic location of the mobile Internet-connected computing device because Montagna incorporates Mathews by reference in its entirety.

222. However, even if Mathews were found to not be incorporated by reference into Montagna, it would have been obvious to combine Mathews’ teachings with that of Montagna and Lewin. As I discussed with respect to claim 28 in Ground 1, Montagna teaches the use of a verification module to determine the games that are available to the user based on the user’s location. Exhibit 1003 (Montagna), [0091] (“In one implementation, the above described user interface data

can include a result of a verification check performed by a verification module. The verification module is generally an application that *verifies*, among other things, *a user's* identity and *place of residence*.... In the instant implementation, the above described rules can specify which games are available to a user based on results of the verification.”). Mathews teaches a variety of means that a verification module can use to determine the location of a user, including IP address and GPS. *See* Exhibit 1007 (Mathews), 5:41-48 (“Given an IP address a player is using to access the Internet, GeoPoint can determine the following information: *Physical location (i.e., continent, country, state, city, and postal or zip code), latitude and longitude*....”), 25:37-41 (“The verification module can use any data source providers and is not limited to the examples described. For example, *the verification module can obtain longitude and latitude information from a data source that uses GPS to locate a user*.”).

(a) Motivation to combine Montagna and Mathews

223. A POSITA would have been motivated to combine the teachings of GPS location verification as taught by Mathews with the system taught by Montagna. Montagna describes that the game menu provided to a user can vary based on the user's location. *See, e.g.*, Exhibit 1003 (Montagna), [0008] (“A system in accordance with the invention can define user interfaces based on user specific

data and configurable criteria. Personal data can indicate, for example, a jurisdiction of law to which a user is subject, a geographic location where the user has been verified to be currently located, the user's place of residence, and any combination of the aforementioned.”), [0009], [0010]. Montagna also describes the use of a verification module to determine the user’s location. *Id.*, [0091] (“The verification module is generally an application that verifies, among other things, a user's identity and place of residence.”). Thus, in using the system taught by Montagna, a POSITA would have looked for ways to implement Montagna’s verification module and would have readily considered Mathews because Montagna itself explicitly states that Mathews describes its verification module. *Id.*, [0091] (“The verification module is further described in commonly assigned U.S. patent application Ser. No. 10/280,895, which is hereby incorporated by reference in its entirety.”). U.S. Patent Application No. 10/280,895 issued as U.S. Patent No. 6,853,973, which is Mathews. Thus, Montagna itself provides explicit motivation to combine its system with Mathews’s verification module and GPS location verification functionality.

224. Moreover, a POSITA would have used basic and well-known computer programming principles to implement Mathew’s verification module in Montagna’s system. This implementation would have yielded the predictable results of allowing Montagna’s system to determine the user’s location. A POSITA would have possessed the knowledge and skillset to perform this implementation.

***(b) Reasonable expectation of success in
combining Montagna and Mathews***

225. A POSITA would also have had a reasonable expectation of success in combining Montagna and Mathews. Montagna and Mathews are both from the same field, which is shared by the '883 patent—providing online gaming to users. Exhibit 1003 (Montagna), Abstract; Exhibit 1007 (Mathews), Abstract (“Methods and apparatus, including computer program products, for verifying a player's eligibility for participation in on-line gaming. A method for verifying a player's eligibility to participate in on-line gaming includes performing one or more verification checks for the player.”). Mathews is meant to provide a complementary service to systems like that of Montagna by verifying the user’s location so that systems can determine what services to offer. *See, e.g.*, Exhibit 1007 (Mathews), 1:55-65. Mathews specifies that its services can be integrated into gaming systems like that described in Montagna. *Id.*, 4:46-52. Mathews also specifies that its verification module can “accommodate limitations” of various data sources that have “limited capabilities to verify the information of users residing in a certain geographic location.” *Id.*, 2:53-61 (“The importance of information from a particular data source can be varied, advantageously allowing the verification module to accommodate limitations of a particular data source. For example, if a data source has limited capabilities to verify the information of users residing in a certain geographic location, the verification

module can be configured to give less weight to data from the data source when verifying users from the geographic location. Consequently, the risk of erroneously rejecting an eligible user because of the limitations of the data source is reduced. ”). This functionality of Mathew’s verification module can thus be provided to Montagna’s system, which includes requesting information from and sending information to different servers.

226. Therefore, a POSITA would have no trouble finding motivation, or success in integrating the teachings of Mathews into Montagna’s system or the Montagna-Lewin system.

227. Thus, Montagna alone, or in combination with Lewin, further combined with Mathews also renders obvious claim 28.

2. Dependent Claims 7 and 89

228. Claims 7 and 89 recite the same limitations as claim 28, and only differ because they depend from claim 6 and 81, respectively, rather than claim 27. For the reasons that I provided above for claim 28, Montagna alone, or in combination with Lewin, further combined Mathews also renders obvious claims 7 and 89.

C. Ground 3: Claims 39, 40, 18, 19, 100, and 101 are Obvious Over Montagna Alone or in View of Lewin, and Further in View of Zilliacus

229. In Ground 1, I describe how Montagna alone, or in combination with Lewin, teaches a method of delivering a list of assets to a mobile Internet-connected computing device so that a user can quickly and efficiently receive the assets

necessary to play a game. In Ground 3, I consider the teachings of Zilliacus (Exhibit 1006), which is directed towards methods of managing memory on a mobile device by automatically replacing or purging unused assets. As I will describe below, Montagna alone, or in combination Lewin, further combined with Zilliacus renders obvious claims 39, 40, 18, 19, 100, and 101.

1. Dependent Claim 39

“The method of claim 20 further comprising replacing unused assets to manage memory on the mobile Internet-connected computing device.”

230. Montagna alone, or in combination with Lewin, renders obvious claim 20, as I discussed above in Gound 1. Montagna alone, or in combination with Lewin, further combined with Zilliacus renders obvious the additional limitation of claim 39 (replacing unused assets to manage memory on the mobile Internet-connected computing device).

231. Zilliacus discloses a method of managing applications on a mobile Internet-connected computing device. *See* Exhibit 1006 (Zilliacus), 3:39-42 (“An embodiment of the present invention also provides a method by which an application with limited lifetime on the mobile phone can uninstall itself and thus free memory space for new applications.”). To free memory, Zilliacus describes that either the user can replace applications, or the system can automatically replace unused applications on behalf of the user. *See id.*, 4:20-23 (“When the application is

downloaded to the mobile terminal it may be configured to delete itself or uninstall itself from memory of the mobile terminal when the lifetime expires.”), 7:54-59 (“If a user needs to download an application to his or her mobile terminal when there is no memory available to store the application, the user will have to delete an existing application or an application may be deleted automatically to make room for the new application.”). Because an application is made up of assets, when an application is replaced, the assets that make up that application are also replaced.

i. Motivation to combine Montagna, Lewin, and Zilliacus

232. A POSITA would have found it obvious to replace unused assets as taught by Zilliacus in Montagna’s system or the Montagna-Lewin system. As I discussed in the Section IV (Technology Background), a POSITA would have been well versed in issues of memory management, and the concerns and problems that arose with transferring and storing large amounts of data. At the time, a POSITA would have understood that Montagna’s system or the Montagna-Lewin system would not store information in perpetuity, nor would any computing system. Thus, a POSITA using the methods as taught by Montagna alone, or in combination with Lewin, would be actively searching for means to manage the end of life of the assets and other information transferred as part of Montagna’s methods. A POSITA would have found it obvious to replace unused assets in light of the teachings of Zilliacus,

and a POSITA would have found combining Zilliacus with Montagna's system or the Montagna-Lewin system obvious as I detail below.

233. Montagna, Lewin, and Zilliacus are from the same field, which is shared by the '883 patent. All three relate to the provision of applications, including games, to users. *See* Section IX.A (overview of Montagna); Section IX.B (overview of Lewin); Section IX.C (overview of Zilliacus); *see also* Exhibit 1006 (Zilliacus) 2:49-53 (“Games are often available in a limited dataset where only the first level is available to the user. The limited dataset application is generally freeware, however, use of the complete dataset—the additional levels—requires payment of the purchase price.”). Both Montagna and Zilliacus relate to issues that arise with the desire to minimize the amount of data that is maintained in storage of a user device, and Lewin describes that the files used in its system can be quite large. Montagna minimizes the amount of stored data by ensuring that duplicative data is not stored in device memory. Exhibit 1003 (Montagna), [0089] (“[T]he Flash Game Loader first checks for locally cached resources before requesting a download of the resources from the gaming server.”), [0010]. Zilliacus does so by ensuring that information that is no longer necessary is removed. Exhibit 1006 (Zilliacus), 3:16-19 (“Because of limited memory capacity, a user or the mobile terminal itself may have to eliminate an old application in memory to make room for a new application in memory.”). Thus, both Montagna and Zilliacus seek to improve the operation of

a device that a user operates when interacting with entertainment services. However, as I discussed above, the use of Montagna's system or the Montagna-Lewin system does not obviate the need for a system like that taught by Zilliacus—a POSITA would still need to determine how to remove assets that are no longer in use by the game client.

234. Further, implementing the replacement of unused assets as taught by Zilliacus in Montagna's system or the Montagna-Lewin system merely amounts to implementing known techniques to improve similar systems. In any computing system, there exists a need to remove old information. As taught by Zilliacus, one method to do so is by replacing unused assets when new assets arrive at the user device. Exhibit 1006 (Zilliacus), 3:16-19, 7:54-59 ("If a user needs to download an application to his or her mobile terminal when there is no memory available to store the application, the user will have to delete an existing application or an application may be deleted automatically to make room for the new application."). This known technique would thus have improved Montagna's system or the Montagna-Lewin system, as old assets are no longer stored on the device.

235. There would have been finite options for the removal of old assets. One option would be to wait until new assets have arrived (as I discussed above). Another option would be to simply remove those unused assets proactively to ensure

that excess memory is not being used (as I will discuss below). Of the two options for removing assets, a POSITA would have been motivated to try both.

236. Finally, a POSITA would have been motivated to combine Zilliacus with Montagna alone, or in combination with Lewin, because it would have simply involved implementing Zilliacus's method of removing unused assets in Montagna's system or the Montagna-Lewin system according to known methods. Such combination would have yielded predictable results. A POSITA would have used basic and well-known computer programming principles to implement this combination. This combination would have yielded the predictable results of removing unused assets in Montagna's system or the Montagna-Lewin system. A POSITA would have possessed the knowledge and skillset to perform this implementation.

*ii. Reasonable expectation of success in combining
Montagna, Lewin, and Zilliacus*

237. A POSITA would also have had a reasonable expectation of success in combining Zilliacus with Montagna's system or the Montagna-Lewin system. Both Montagna and Zilliacus describe methods of managing assets on a user device. Both use standard computing devices to operate their respective programs. Neither requires specialized circuitry to implement their systems. Exhibit 1003 (Montagna), [0101] ("The invention and all of the functional operations described herein can be

implemented in digital electronic circuitry, or in computer hardware, firmware, software, or in combinations of them. The invention can be implemented as a computer program product, i.e., a computer program tangibly embodied in an information carrier, e.g., in a machine-readable storage device or in a propagated signal, for execution by, or to control the operation of, data processing apparatus, e.g., a programmable processor, a computer, or multiple computers.”), [0103]; Exhibit 1006 (Zilliacus), 1:40-45 (“Accordingly, the term ‘mobile terminal’ shall herein refer to a broad range of devices (e.g. Personal Digital Assistants (‘PDA’), Communicators, SmartPhones, and Personal Computers (‘PC’)) and not solely devices operable merely to communicate in a cellular communication system.”); 1:46-49 (“Mobile terminals utilize a range of processing circuitry to perform the functions described herein. The processing circuitry is generally capable of running, i.e., executing, content, also referred to herein as an application program.”). The systems disclosed by Montagna, Lewin, and Zilliacus would all have included mechanisms and capabilities for managing the memory of the devices. Thus, to implement the techniques disclosed in Zilliacus would require no additional capabilities over what is disclosed by Montagna and Lewin, and a POSITA would have had a reasonable expectation of success in combining the replacement of unused assets as taught by Zilliacus with Montagna’s system or the Montagna-Lewin system.

238. Thus, Montagna alone, or in combination with Lewin, further combined with Zilliacus renders obvious dependent claim 39.

2. Dependent Claim 40

“The method of claim 20 further comprising purging unused assets to manage memory on the mobile Internet-connected computing device.”

239. Montagna alone, or in combination with Lewin, renders obvious claim 20, as I discussed above in Ground 1. Montagna alone, or in combination with Lewin, further combined with Zilliacus renders obvious the additional limitation of claim 40 (purging unused assets to manage memory on the mobile Internet-connected computing device).

240. Zilliacus teaches a method where unused applications delete (i.e., purge) themselves automatically at the end of a pre-determined period. Exhibit 1006 (Zilliacus), 3:52-61 (“The application(s) in the application database have a selectable lifetime during which the application is executable. In addition, the applications may be configured to delete or uninstall themselves from memory of the mobile terminal when their lifetime expires. The applications may be preprogrammed to delete themselves according to a preset lifetime or configured to delete themselves after the lifetime is selected. The lifetime may be calculated on time or transaction basis.”), 6:53-56 (“Deletion comprises freeing all memory space used by the application and corresponding data or only freeing some specified portion of the memory space used

by the application.”), 8:44-45 (“Second, the user may choose to remove an existing application from his or her mobile terminal”), 6:64-67 (“At block 320, the application determines if its lifetime has expired. If so, the application deletes itself at block 322. If not, the application does not delete itself and allows the user to continue using the application.”), 4:20-23. Zilliacus describes that this deletion is done to manage memory on the mobile device. *Id.*, 3:39-42 (“An embodiment of the present invention also provides a method by which an application with limited lifetime on the mobile phone can uninstall itself and thus free memory *space* for new applications.”), Abstract, 3:16-19, 6:25-29, 6:44-48, 8:56-60.

241. As I discussed for dependent claim 39, a POSITA would have been motivated to combine the asset deletion features of Zilliacus with Montagna’s system or the Montagna-Lewin system and would have had a reasonable expectation of success in doing so. As I also discussed, a POSITA could choose to wait for new assets to arrive to purge them or could proactively remove assets when they are no longer necessary. Both methods are discussed in Zilliacus, and a POSITA would have likely tried both in Montagna’s system or the Montagna-Lewin system, if not provide both options to users.

242. Thus, Montagna alone, or in combination with Lewin, further combined with Zilliacus renders obvious dependent claim 40.

3. Dependent Claims 18, 19, 100, and 101

243. Claims 18, 19, 100, and 101 only differ from claims that I already discussed above because they depend from claim 1 or 81 rather than claim 20. For the reasons that I already provided above for the claims that depend from claim 20, Montagna alone, or in combination with Lewin, further combined with Zilliacus also renders obvious claims 18, 19, 100, and 101 according to the mapping in the following table.

Prior discussed claim depending from claim 20	Corresponding claims depending from:	
	Claim 1	Claim 81
39	18	100
40	19	101

XI. CONCLUSION

244. Based on the foregoing, it is my opinion that claims 1, 2, 5-9, 17-23, 26-30, 38-40, 81-84, 87-91, and 99-101 of the '883 patent are invalid.

XII. AVAILABILITY FOR CROSS-EXAMINATION

245. In signing this declaration, I recognize that the declaration will be filed as evidence in a contested case before the Patent Trial and Appeal Board of the United States Patent and Trademark Office. I also recognize that I may be subject to cross-examination in the case and that cross-examination will take place within the United States. If cross-examination is required of me, I will appear for cross-examination within the United States during the time allotted for cross-examination.

XIII. RIGHT TO SUPPLEMENT

246. I reserve the right to supplement my opinions in the future to respond to any arguments that the Patent Owner raises and to take into account new information as it becomes available to me.

XIV. JURAT

247. I declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code.

Dated: January 14, 2026



Dr. Sandeep Chatterjee

APPENDIX A

Professional Summary

Sandeep Chatterjee, Ph.D. is a seasoned technology expert and business professional with almost two decades of hands-on contributions as a thought leader, technologist, consultant, entrepreneur, and author. He is an expert in computer software and hardware systems, with particular emphasis on distributed systems and architectures, service-oriented architectures (SOAs), software-as-a-service (SaaS) and Web Services, end-to-end security, quality-of-service (QoS), communications and telecommunications systems, location-based services, social network services and social media, as well as mobile and wireless systems and applications. He is the co-author of a book on developing enterprise computing systems, which has been adopted by over 100 universities worldwide.

Dr. Chatterjee also has extensive experience with providing expert testimony for intellectual property and commercial litigation, including for high stakes patent litigation, copyright and trade secret misappropriation, contract disputes and patent licensing cases. He has testified at trial, and has had his deposition taken more than 100 times. Dr. Chatterjee combines strong experience with expert testimony within the context of litigation together with worldwide experience in designing, architecting and implementing complex computing systems.

Honors & Achievements

- Listed in the **IAM Patent 1000** of the world's leading patent professionals.
- Named a **Young Global Leader** by the **World Economic Forum** for professional accomplishments, commitment to society and potential to contribute to shaping the future of the world.
- Doctoral dissertation at the Massachusetts Institute of Technology (MIT) was selected as one of the **most important inventions in computing**, and the invention is preserved and showcased in a time-capsule at the Museum of Science in Boston, Massachusetts. Other recipients of this honor include **Bill Gates** (founder of Microsoft) and **Tim Berners-Lee** (inventor of the World Wide Web).
- Technology solution designed by Dr. Chatterjee was identified as a Bloomberg Innovation, and the company he co-founded to commercialize that technology solution was featured on Bloomberg TV's "**Bloomberg Innovators**" program.

- Member of the **JSR 172 Expert Group** that defined the worldwide standard for J2ME mobile Web services.

Professional Experience

From: 2013
To: Present
Organization: Experantis LLC
Title: Chief Executive Officer
Summary: Experantis LLC provides consultation and services in all facets of commercial litigation.

From: 2016
To: 2019
Organization: International Institute of Digital Technologies
Title: Dean, Mobility Center of Excellence
Summary: International Institute of Digital Technologies (IIDT) is a leading institute of higher education, focused on digital technologies.

From: 2013
To: Present
Organization: S3G Technology LLC
Title: Member
Summary: S3G Technology LLC provides consultation, products and services related to mobile and wireless solutions that deliver critical services to semi-urban, rural and remote populations across the world.

From: 2011
To: Present
Organization: World Economic Forum
Title: Young Global Leader; Member of the Expert Network
Summary: The World Economic Forum is an international institution for public-private cooperation, and its mission is cited as "committed to improving the state of the world by engaging business, political, academic, and other leaders of society to shape global, regional, and industry agendas."

From: 2007
To: 2013
Organization: Shuv Gray LLC
Title: Chief Executive Officer
Summary: Shuv Gray LLC provides information technology and intellectual property consultation and services.

From: 2007
To: 2012
Organization: SourceTrace Systems, Inc.
Title: Chief Technology Officer & Executive Vice President
Summary: SourceTrace Systems is a leading vendor of end-to-end mobile and wireless distributed transactional systems for the financial services, agricultural commodities, and product distribution markets around the world.

From: 2004
To: 2007
Organization: Cyndeo LLC
Title: Chief Executive Officer
Summary: Founded Cyndeo to be a leading provider of enterprise integration and mobilization solutions as well as technology strategy consulting and outsourcing services to global corporations, government agencies, and major not-for-profit organizations. Cyndeo provides consulting, software engineering development, and management services to companies and organizations of all sizes, from Fortune Global 100 corporations to small organizations in developing countries.

From: 2001
To: 2002
Organization: Hewlett-Packard (Palo Alto, CA)
Title: Senior Member of Technical Staff
Summary: Independently identified a need in the mobile and e-Business marketplace for a more flexible Web Services platform. Invented a patent-pending (applied) solution, and led the development as well as the initial sales and marketing efforts.

- This next-generation J2EE (HP/Bluestone app server) Web Services solution ties together the emerging Web Services standards, end-to-end transactions, and optimized mobile services & applications.
- Led and managed the entire lifecycle of the product from conception, architectural design, product development, QA, documentation, senior management approval, patent filing, preparation for marketing launch at JavaOne, as well as interfacing with early adopter customers and strategic partners.
- Managed relationships with multiple HP organizations and teams that contributed to the overall solution.
- Positioned the product, targeted key early adopters and partners, and leveraged my own contacts to build successful strategic relationships.
- Selected to provide support to HP/Bluestone's top sales manager, and helped land a strategic account in the Asia-Pac market.
- Selected as the representative for Hewlett Packard to the JSR 172 Expert Group that is defining the worldwide standard for J2ME mobile Web Services.

- From: 1999
To: 2000
Organization: Satora Networks
Title: Founder & Chief Technology Officer
Summary: Founded Satora Networks to create wireless platforms and services for Internet devices based on the technologies invented by Dr. Chatterjee as part of his Ph.D. research at MIT Lab for Computer Science.
- Raised venture funding, recruited the management and core engineering teams, set corporate and product strategy, handled customer and partner relations, and managed overall corporate affairs.
 - Architected and led the development of the StrongARM-based mobile hardware platform, Linux-based mobile OS, client-side services, and server-side J2EE environment.
 - Landed key customer and partner wins, managed revenue and burn rate to become profitable early on, and oversaw the growth of the company.
- From: 1995
To: 2001
Organization: MIT Lab for Computer Science
Title: Doctoral Researcher
Summary: Designed a modular system architecture that supports the cost-effective development of network devices and services. The system consists of Lego-like commodity hardware and standards-based software components that are quickly composed together through a development environment into optimized end-user devices as well as client-side services.
- Architected and implemented the entire system: a Linux-based client software environment, a wireless network and StrongARM processor-based device hardware platform, as well as a Web-based design and optimization environment and tool.
 - Successfully mentored and managed a research group of 6 Masters and Bachelors students.
 - The technology underlying Dr. Chatterjee's thesis was selected as one of the top inventions in thirty-five year history of MIT Laboratory for Computer Science.
- From: 1997
To: 1999
Organization: FidelityCAPITAL (Boston, MA)
Title: Entrepreneur-In-Residence
Summary: Worked with Fidelity senior management and FidelityCAPITAL partners to source, analyze, and fund technology start-ups strategic to Fidelity interests.
- Evaluated business plans submitted to Fidelity for venture funding, and analyzed emerging market and technology trends to identify opportunities for internal spin-offs in the telecom and eBusiness spaces.
 - Founded and served as President & CTO of a wireless devices and services company (Satora Networks) that was seed funded by Fidelity.

Litigation Support Experience

Expert Engagement:

Type of Matter: Enterprise Software Patent Infringement
Case Name: Software AG and Software AG v. BEA Systems

Expert Engagement:

Type of Matter: Mobile software intellectual property litigation.
Case Name: Esmertec v. Tao Group

Expert Engagement:

Type of Matter: Enterprise and mobile software intellectual property litigation.
Case Name: Metrologic Instruments v. Symbol Technologies

Expert Engagement:

Type of Matter: Distributed and mobile computing trade secret litigation.
Case Name: Modular Mining Systems (Komatsu Ltd.) v. Jigsaw Systems et al.

Expert Engagement:

Type of Matter: Distributed computing and network routing systems intellectual property litigation.
Case Name: WebXChange v. Dell

Expert Engagement:

Type of Matter: Communications systems intellectual property litigation.
Case Name: Motorola, Inc. et al. v. Rembrandt Technologies, LP

Expert Engagement:

Type of Matter: Database and object mapping intellectual property litigation.
Case Name: DataTern v. United Airlines

Expert Engagement:

Type of Matter: Control systems intellectual property litigation.
Case Name: Finisar v. Source Photonics

Expert Engagement:

Type of Matter: Mobile systems intellectual property litigation.
Case Name: OpenWave v. 724 Solutions

Expert Engagement:

Type of Matter: Mobile systems intellectual property litigation.
Case Name: Ganas v. Charles Schwab et al

Expert Engagement:

Type of Matter: Control systems intellectual property litigation.
Case Name: Finisar v. Oplink

Expert Engagement:

Type of Matter: Mobile systems intellectual property litigation.
Case Name: Motorola Mobility (Google) v. Microsoft

Expert Engagement:

Type of Matter: Wireless networking & management intellectual property litigation.
Case Name: Netgear v. Ruckus Wireless

Expert Engagement:

Type of Matter: Mobile systems intellectual property litigation.
Case Name: XpertUniverse v. Cisco Systems

Expert Engagement:

Type of Matter: Mobile systems intellectual property litigation.
Case Name: Aeritas v. US Airways, Inc., Delta Airlines, Inc., Alaska Air Group

Expert Engagement:

Type of Matter: Distributed systems intellectual property litigation.
Case Name: Wellogix v. ADP et. al.

Expert Engagement:

Type of Matter: Trade secret misappropriation litigation.
Case Name: Bradford Technologies v. NCV Software et al.

Expert Engagement:

Type of Matter: Mobile systems intellectual property litigation.
Case Name: Ericsson Inc. v. Samsung Electronics Co., Ltd. et al.

Expert Engagement:

Type of Matter: Contract dispute litigation.
Case Name: AMC Technologies v. Cisco Systems

Expert Engagement:

Type of Matter: Distributed systems and security intellectual property litigation.
Case Name: Media Rights Technologies, Inc. v. Capital One Financial Corporation

Expert Engagement:

Type of Matter: Distributed computing intellectual property litigation.
Case Name: American Airlines, Inc. et al. v. Loyalty Conversion Systems Corporation

Expert Engagement:

Type of Matter: Mobile computing and security intellectual property litigation.
Case Name: Tierra Intellectual Borinquen, Inc. v. Toshiba Corporation et. al.

Expert Engagement:

Type of Matter: Distributed computing systems and software intellectual property litigation.
Case Name: Kroy IP Holdings, LLC v. Safeway, Inc. and The Kroger Co.

Expert Engagement:

Type of Matter: Secure mobile computer hardware and software intellectual property litigation.
Case Name: Maxim Integrated Products, Inc. v. JPMorgan Chase & Co.

Expert Engagement:

Type of Matter: Computer hardware and software antitrust litigation.
Case Name: In Re: Keurig Green Mountain Single Serve Coffee Antitrust Litigation.

Expert Engagement:

Type of Matter: Mobile computer and location-based systems intellectual property litigation.
Case Name: Unwired Planet LLC v. Square, Inc.

Expert Engagement:

Type of Matter: Distributed computing systems intellectual property litigation.
Case Name: AirWatch LLC v. Good Technology Corporation et al.

Expert Engagement:

Type of Matter: Mobile computer and location-based systems intellectual property litigation.
Case Name: Unwired Planet LLC v. Square, Inc.

Expert Engagement:

Type of Matter: Distributed computing systems intellectual property litigation.
Case Name: Good Technology Corporation et al. v. AirWatch LLC

Expert Engagement:

Type of Matter: Mobile computer and location-based systems intellectual property litigation.
Case Name: Square, Inc. v. Unwired Planet LLC

Expert Engagement:

Type of Matter: Software systems intellectual property litigation.
Case Name: American Express Company et al. v. Signature Systems, LLC

Expert Engagement:

Type of Matter: Software and backup systems intellectual property litigation.
Case Name: Farstone Technology, Inc. v. Apple, Inc.

Expert Engagement:

Type of Matter: Distributed computing systems intellectual property litigation.
Case Name: MobileIron, Inc. v. Good Technology Corporation

Expert Engagement:

Type of Matter: Data management intellectual property litigation.
Case Name: Olivistar v. Regions Bank et al.

Expert Engagement:

Type of Matter: Cloud computing and data migration intellectual property litigation.
Case Name: BitTitan v. SkyKick

Expert Engagement:

Type of Matter: Breach of contract, fraud litigation involving large scale enterprise software.
Case Name: State of Oregon v. Oracle et al.

Expert Engagement:

Type of Matter: Document processing intellectual property litigation.
Case Name: Capital Security Systems v. NCR et al.

Expert Engagement:

Type of Matter: Web page intellectual property litigation.
Case Name: Tele-Publishing, Inc. v. Facebook et al.

Expert Engagement:

Type of Matter: Intellectual property litigation.
Case Name: Sound View Innovations, LLC v. Facebook, Inc.

Expert Engagement:

Type of Matter: Intellectual property litigation.
Case Name: Zkey Investments, LLC v. Facebook, Inc.

Expert Engagement:

Type of Matter: Intellectual property litigation.
Case Name: Vaporstream, Inc. v. Snap, Inc.

Expert Engagement:

Type of Matter: Intellectual property litigation.
Case Name: Symantec Corporation v. Zscaler, Inc.

Expert Engagement:

Type of Matter: Intellectual property litigation.
Case Name: American Express Company et al. v. Signature Systems, LLC

Expert Engagement:

Type of Matter: Intellectual property litigation.
Case Name: Apogee Telecom, Inc. v. Brian Rosenblatt

Expert Engagement:

Type of Matter: Intellectual property arbitration.
Case Name: Trupanion, Inc. v. Veterinary Data Services, Inc.

Expert Engagement:

Type of Matter: Intellectual property litigation.
Case Name: Leftsnrights, Inc. v. 33Across, Inc.

Expert Engagement:

Type of Matter: Intellectual property litigation.
Case Name: Ameranth, Inc. v. Mobo Systems, Inc. (Olo)

Expert Engagement:

Type of Matter: Intellectual property litigation.

Case Name: Facebook, Inc. and Instagram, LLC v. Search and Social Media Partners, LLC

Expert Engagement:

Type of Matter: Intellectual property litigation.

Case Name: Facebook, Inc. v. Hyper Search LLC

Expert Engagement:

Type of Matter: Intellectual property litigation.

Case Name: Touchstream Technologies, Inc. v. Vizbee, Inc.

Expert Engagement:

Type of Matter: Intellectual property litigation.

Case Name: Facebook, Inc., Instagram, LLC and WhatsApp, Inc. v. Blackberry Limited

Expert Engagement:

Type of Matter: Litigation.

Case Name: Roger Hogan et al. v. Toyota Motor Sales, U.S.A., Inc.

Expert Engagement:

Type of Matter: Intellectual property litigation.

Case Name: Blue Spike LLC v. DISH Network Corp.

Expert Engagement:

Type of Matter: Intellectual property litigation.

Case Name: NCR Corp. v. Pendum, LLC and Burroughs, Inc.

Expert Engagement:

Type of Matter: Intellectual property litigation.

Case Name: Snap, Inc. v. Blackberry Limited

Expert Engagement:

Type of Matter: Intellectual property litigation.

Case Name: Lufthansa Technik and Certified Aviation Services, LLC v. EcoServices, LLC

Expert Engagement:

Type of Matter: Intellectual property litigation.

Case Name: Rovi Guides, Inc. v. Comcast Corp. et al., ITC Inv. No. 337-TA-1158 (2019)

Expert Engagement:

Type of Matter: Intellectual property litigation.

Case Name: Microsoft Corporation, Google LLC, Asustek Computer Inc., Acer Incorporated, HTC Corporation v. Mimzi, LLC

Expert Engagement:

Type of Matter: Intellectual property litigation.
Case Name: Rain Computing, Inc. v. Samsung Electronics Co., Ltd.; Samsung Electronics America, Inc.; and Samsung Research America, Inc.

Expert Engagement:

Type of Matter: Intellectual property litigation.
Case Name: JBF Interlude 2009 Ltd. - Israel v. Quibi Holdings LLC

Expert Engagement:

Type of Matter: Intellectual property litigation.
Case Name: Warner Records, Inc., et al. v. Charter Communications, Inc.

Expert Engagement:

Type of Matter: Intellectual property litigation.
Case Name: UMG Recordings, Inc., et al. v. Bright House Networks, LLC

Expert Engagement:

Type of Matter: Intellectual property litigation.
Case Name: Dolby Laboratories, Inc. v. Intertrust Technologies Corporation

Expert Engagement:

Type of Matter: Intellectual property litigation.
Case Name: Document Dynamics, LLC v. Xerox Corporation

Expert Engagement:

Type of Matter: Intellectual property litigation.
Case Name: Thaddeus Gabara v. Facebook, Inc.

Expert Engagement:

Type of Matter: Intellectual property litigation.
Case Name: NexStep, Inc. v. Comcast Cable Communications, LLC

Expert Engagement:

Type of Matter: Intellectual property litigation.
Case Name: Facebook, Inc. v. USC IP Partnership, L.P.

Expert Engagement:

Type of Matter: Intellectual property litigation.
Case Name: Cloudfchange, LLC v. NCR Corp.

Expert Engagement:

Type of Matter: Intellectual property litigation.
Case Name: CommVault Systems, Inc. v. Rubrik Inc.

Expert Engagement:

Type of Matter: Intellectual property litigation.
Case Name: Google LLC v. Express Mobile, Inc.

Expert Engagement:

Type of Matter: Intellectual property litigation.
Case Name: Uber Technologies, Inc. v. AGIS Software Development LLC

Expert Engagement:

Type of Matter: Intellectual property litigation.
Case Name: Broadband iTV, Inc. v. AT&T Services, Inc. and AT&T Communications LLC

Expert Engagement:

Type of Matter: Intellectual property litigation.
Case Name: Broadband iTV, Inc. v. DIRECTV, LLC

Expert Engagement:

Type of Matter: Intellectual property litigation.
Case Name: Samsung Electronics Co., Ltd. v. Blaze Mobile, Inc. and Michelle Fisher

Expert Engagement:

Type of Matter: Intellectual property litigation.
Case Name: Proven Networks, LLC v F5 Networks, Inc.

Expert Engagement:

Type of Matter: Intellectual property litigation.
Case Name: WSOU Investments, LLC v F5 Networks, Inc.

Expert Engagement:

Type of Matter: Intellectual property litigation.
Case Name: People.AI, Inc. v SetSail Technologies, Inc.

Expert Engagement:

Type of Matter: Intellectual property litigation.
Case Name: IGT and IGT Canada Solutions ULC v Zynga Inc.

Expert Engagement:

Type of Matter: Intellectual property litigation.
Case Name: Express Mobile, Inc. v Booking Holdings, Inc.

Expert Engagement:

Type of Matter: Intellectual property litigation.
Case Name: Wapp Tech Limited Partnership v. Wells Fargo Bank, N.A.

Expert Engagement:

Type of Matter: Intellectual property litigation.
Case Name: Trident Holdings, Inc. v. HubSpot, Inc.

Expert Engagement:

Type of Matter: Intellectual property litigation.
Case Name: Hafeman v. LG Electronics, Inc.

Expert Engagement:

Type of Matter: Securities litigation.
Case Name: In re Cloudera, Inc. Securities Litigation

Expert Engagement:

Type of Matter: Intellectual property litigation.
Case Name: Fintiv Inc. v. Paypal Holdings Inc.

Expert Engagement:

Type of Matter: Intellectual property litigation.
Case Name: Lyft, Inc. v. AGIS Software Development LLC

Expert Engagement:

Type of Matter: Intellectual property litigation.
Case Name: Groove Digital, Inc. v. Jam City, Inc.

Expert Engagement:

Type of Matter: Intellectual property litigation.
Case Name: Consumeron, LLC v. Maplebear Inc. D/B/A Instacart

Expert Engagement:

Type of Matter: Intellectual property litigation.
Case Name: Apple Inc. v. Speir Technologies Ltd.

Expert Engagement:

Type of Matter: Intellectual property litigation.
Case Name: Smart Path Connections, LLC v. Nokia Corp. et al.

Expert Engagement:

Type of Matter: Antitrust litigation.
Case Name: In re Google Play Store Antitrust Litigation

Expert Engagement:

Type of Matter: California Invasion of Privacy Act (CIPA) litigation.
Case Name: Gutierrez v. Converse, Inc.

Expert Engagement:

Type of Matter: Intellectual property litigation.
Case Name: Kove IO, Inc. v. Google LLC

Expert Engagement:

Type of Matter: Intellectual property litigation.
Case Name: Dynapass IP Holdings LLC v. Bank of America Corp et al.

Expert Engagement:

Type of Matter: Intellectual property litigation.
Case Name: Entropic Communications, LLC v Comcast Corporation et al.

Expert Engagement:

Type of Matter: Intellectual property litigation.
Case Name: BMG Rights Management LLC, et al. v. Altice USA, Inc., et al.

Expert Engagement:

Type of Matter: Intellectual property litigation.
Case Name: Multifold International Incorporated Pte. Ltd. v. Google LLC

Expert Engagement:

Type of Matter: Intellectual property litigation.
Case Name: Scale Video Coding LLC v. Cisco Systems, Inc.

Expert Engagement:

Type of Matter: Intellectual property litigation.
Case Name: Dialect LLC v. Bank of America N.A.

Expert Engagement:

Type of Matter: Intellectual property litigation.
Case Name: Warner Records, Inc. et al. v. Altice USA, Inc., et al.

Expert Engagement:

Type of Matter: California Invasion of Privacy Act (CIPA) litigation.
Case Name: Hughes v. The Neiman Marcus Group LLC

Expert Engagement:

Type of Matter: Intellectual property litigation.
Case Name: KP Innovations 2, LLC v. Samsung Electronics Co., Ltd. et al.

Expert Engagement:

Type of Matter: Intellectual property litigation.
Case Name: Sportscaster Inc. (d/b/a PANDA Interactive) v. Sportradar Group AG

Expert Engagement:

Type of Matter: Class action litigation.
Case Name: Attila Csupo et al. v. Google LLC

Expert Engagement:

Type of Matter: Intellectual property litigation.
Case Name: Sportscaster Inc. (d/b/a PANDA Interactive) v. Genius Sports Ltd.

Expert Engagement:

Type of Matter: Intellectual property litigation.
Case Name: BrowserKey, LLC v. Bank of America Corporation, et al.

Expert Engagement:

Type of Matter: California Invasion of Privacy Act (CIPA) litigation.
Case Name: Heiting et al. v. I Am Beyond LLC

Expert Engagement:

Type of Matter: Class action litigation.
Case Name: Taylor v. Google LLC

Expert Engagement:

Type of Matter: Intellectual property litigation.
Case Name: Softex LLC v. Absolute Software Corp. and Absolute Software, Inc.

Expert Engagement:

Type of Matter: Intellectual property litigation.
Case Name: Internet Payments Patents Ltd. v. PayPal, Inc.

Expert Engagement:

Type of Matter: Intellectual property litigation.
Case Name: Promptu Systems Corporation v. Comcast Corporation and Comcast Cable Communications, LLC

Expert Engagement:

Type of Matter: Intellectual property litigation.
Case Name: ResMed Corp. v. Cleveland Medical Devices, Inc.

Expert Engagement:

Type of Matter: Class action litigation.
Case Name: *In Re Social Media Adolescent Addiction*

Expert Engagement:

Type of Matter: Class action litigation.
Case Name: *In Re Social Media*

Expert Engagement:

Type of Matter: Intellectual property litigation.
Case Name: Google LLC v. Telcom Ventures LLC

Expert Engagement:

Type of Matter: Intellectual property litigation.
Case Name: Mobile Data Technologies LLC v. Samsung Electronics America, Inc. and Samsung Electronics Co. Ltd.

Expert Engagement:

Type of Matter: Intellectual property litigation.
Case Name: Netflix, Inc. v. Broadcom Inc., VMware LLC

Non-Litigation Consulting Projects

Consulting Engagement:

Type of Matter: Enterprise software architecture and development
Client: Sevak Solutions, Inc.
Services Provided: Architect, specify, and develop enterprise-class software for secure financial transactioning in multiple markets and countries around the world
Date: 2005-2006

Consulting Engagement:

Type of Matter: Enterprise software architecture and development
Client: United States Agency for International Development (USAID)
Services Provided: Specify and develop financial transactioning and tracking software
Date: 2004-2005

Consulting Engagement:

Type of Matter: Training services
Client: South Korean Ministry of Information and Communications
Services Provided: Conducted a workshop to train selected university professors and corporate researchers on properly using mobile and distributed technology for strategic and financial growth.
Date: 2004

Consulting Engagement:

Type of Matter: Product and service strategy and development
Client: Hewlett-Packard Company
Services Provided: Analyze market opportunities for new, market-specific technology product and service offerings
Date: 2004

Consulting Engagement:

Type of Matter: Mobile enterprise software
Client: ACCION International
Services Provided: Specified, architected, and developed a mobile enterprise software system for optimizing wireless transactioning.
Date: 2003

Consulting Engagement:

Type of Matter: Market, technology and services strategy development
Client: Hewlett-Packard Laboratories

Services Provided: Analyzed multiple country-specific market needs and opportunities, and proposed technology and service offerings

Date: 2003

Publications

1. Chatterjee, Sandeep and Webber, James, Developing Enterprise Web Services: An Architect's Guide (Pearson Education Korea Ltd. and Hong Reung Science Pub. Co. 2005), Korean Trans.
2. Chatterjee, Sandeep and Webber, James, Developing Enterprise Web Services: An Architect's Guide (Prentice Hall 2004).
3. Chatterjee, Sandeep and Webber, James, Developing Enterprise Web Services: An Architect's Guide (Pearson Education Singapore Pvt. Ltd. 2004), Indian Reprint.
4. Chatterjee, Sandeep, "Enterprise Technology: Web Services: The Next Revolution in IT," in *Dataquest* (March 2004).
5. Chatterjee, Sandeep, "Enterprise 2004 Trends: On A Cautious Note," *Dataquest* (Feb. 2004).
6. Chatterjee, Sandeep, "A Real-World Web Services-Based Application," *JavaBoutique* (http://javaboutique.internet.com/articles/WSApplications/realWorld3_1.html).
7. Chatterjee, Sandeep, "Write Once, Run Anywhere Web Services," *JavaBoutique* (http://javaboutique.internet.com/articles/WSApplications/realWorld2_1.html).
8. Chatterjee, Sandeep, "Developing Real World Web Services-Based Applications," *JavaBoutique* (<http://javaboutique.internet.com/articles/WSApplications/>).
9. Chatterjee, Sandeep, Doctoral Thesis Dissertation, "Composable System Resources As An Architecture For Networked Systems," Massachusetts Institute of Technology (2001).
10. Keckler, Stephen W., Chang, Andrew, Lee, Whay Sing, Chatterjee, Sandeep, and Dally, William J., "Concurrent Event Handling Through Multithreading," *IEEE Trans. Computers* 48(9), pages 903-916 (1999).
11. Chatterjee, Sandeep and Devadas, Srinivas, "MASC Composable Computing Infrastructure For Intelligent Environments," *Proceedings of the Industrial Electronics Conference*, pages 132-138 (1999).
12. Chatterjee, Sandeep, "SANI: A Seamless and Non-Intrusive Framework and Agent for Creating Intelligent Interactive Homes," *Second International Conference on Autonomous Agents*, pages 436-440 (1998).
13. Chatterjee, Sandeep, "Towards A MASC Appliances-Based Educational Paradigm," *ACM Symposium on Applied Computing*, pages 112-116 (1998).
14. Chatterjee, Sandeep, "Towards Rapidly Deployable Intelligent Environments," *American Association for Artificial Intelligence Symposium on Intelligent Environments*, pages 31-36 (1998).
15. Chatterjee, Sandeep, "The ModuleC Network Architecture: A Novel Approach Of Computing Through Information Appliances," *IEEE International Symposium on Consumer Electronics* (1997).
16. Chatterjee, Sandeep, Masters Thesis, "Asynchronous Event Handling," Massachusetts Institute of Technology Technical Report (May 1997).

17. Tennenhouse, David L. and Chatterjee, Sandeep, "The First 10 Feet: The Missing Story For Encouraging User Investment In Universal Broadband Connectivity" (October 1996).
18. Chatterjee, Sandeep and Faraboschi, Paolo, "The VLIW Trace Scheduling Compiler Visual Analysis System," Hewlett-Packard Laboratories Internal Technical Report (September 1995).
19. Chatterjee, Sandeep and Donohue, Richard J., "Electron Gamma Shower Windows 2," *International Conference on Monte Carlo Simulations in Nuclear & High Energy Physics* (February 1993).