

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

TWITCH INTERACTIVE, INC.

Petitioner,

v.

RAZDOG HOLDINGS LLC

Patent Owner.

PTAB Case No. IPR2025-00307 and IPR2025-00308

U.S. Patent Nos. 9,729,658 and 10,334,068

**DECLARATION OF PAUL D. MARTIN, PH.D. IN SUPPORT OF
PETITIONS FOR *INTER PARTES* REVIEW OF U.S. PATENT NOS.
9,729,658 AND 10,334,068**

TABLE OF CONTENTS

	Page
EXHIBIT LIST	viii
I. INTRODUCTION	1
II. BACKGROUND AND QUALIFICATIONS.....	2
A. Education.....	2
B. Career	4
C. Research	7
D. Teaching	12
E. Patents.....	13
F. Publications and Professional Associations	13
G. Compensation.....	13
III. MATERIALS CONSIDERED	14
IV. RELEVANT LEGAL STANDARDS	14
V. THE '658 AND '068 PATENTS.....	20
A. Background	20
B. Specification.....	22
C. The Patent Claims	26
1. The '658 Patent.....	26
2. The '068 Patent.....	29
D. Prosecution History of the '658 Patent	31
E. Prosecution History of the '068 Patent	32

TABLE OF CONTENTS
(continued)

	Page
F. Priority Date	33
VI. OVERVIEW OF THE TECHNOLOGY	34
A. Cloud Computing	34
1. Moving from Local Apps to Cloud Apps.....	36
B. RSS Feed Readers	39
1. RSS Readers Moved To The Cloud.....	50
VII. LEVEL OF ORDINARY SKILL IN THE ART	52
VIII. CLAIM CONSTRUCTION.....	54
IX. PRIOR ART	56
A. Summary of Darnell (Ex. 1004).....	56
B. Summary of Kim (Ex. 1005).....	59
C. Summary of Dasgupta (Ex. 1006).....	63
X. SUMMARY OF OPINIONS.....	66
A. '658 Patent.....	66
B. '068 Patent.....	67
XI. '658 PATENT: DETAILED EXPLANATION OF THE GROUNDS	67
A. GROUND 1A: DARNELL AND KIM RENDER CLAIMS 1-5 AND 7-11 OBVIOUS.	67
1. Motivation to Combine Darnell with Kim.....	67
2. Independent Claim 1.....	73

TABLE OF CONTENTS
(continued)

	Page
3. Independent Claim 7.....	108
4. Claim 2: “The cloud-based computer system of claim 1, wherein the content sources or service providers include one or more personal content account and at least one or more social media account belonging to the user.”	114
5. Claims 3 & 11: [Claim 3] “wherein the personal content account is an e-mail account”; [Claim 11] “wherein one of the content sources or service providers includes an e-mail web-site belonging to the user.”	118
6. Claim 4: “The cloud-based computer system of claim 1, wherein the master application includes tools for enabling security features that limits access to one or more of the content sources or service providers based a user log-in belonging to the user.”	120
7. Claims 5 & 8: [Claim 5] “wherein the master server collects and stores history analytics based content data and content applications viewed by the user”; [Claim 8] “wherein the master server collects and stores history analytics based on usage of the remote computer.”	122
8. Claim 9: “The computer system of claim 8, wherein the master server uses the history analytics to select content data or the content applications based on a user log-in belonging to the user.”	125
9. Claim 10: “The computer system of claim 7, wherein the user interface is an icon-based master web-page.”	127
B. GROUND 1B: DARNELL, KIM, AND DASGUPTA RENDER CLAIMS 6 AND 12-18 OBVIOUS	130

TABLE OF CONTENTS
(continued)

	Page
1. Motivation to Combine Dasgupta with Darnell and Kim	130
2. Claim 6: “The cloud-based computer system of claim 5, the master application includes a purchase tool that allows the user to purchase goods or services from the multiple content sources or service providers from master web-page.”	135
3. Independent Claims 12	141
4. Dependent Claim 13: “The method of claim 12, wherein setting up the account includes one or more of enabling security features and selecting personal settings.”	147
5. Dependent Claim 14: “The method of claim 12, wherein one of the multiple content sources or service providers is an c-mail [sic] content source or service provider.”	147
6. Dependent Claims 15: “The method of claim 12, wherein the user interface is an icon-base web-page.”	148
7. Dependent Claim 16: “The method of claim 15, further comprising logging the cloud-based master server from a remote computer, wherein the cloud-based master server initiates a down load of software on to the remote computer that enables the slide tool program.”	148
8. Dependent Claim 17: “The method of claim 16, further comprising selectively viewing the content data or content applications from the multiple content sources or service providers by scrolling or paging through the content data or content applications using the slide tool program.”	150

TABLE OF CONTENTS
(continued)

	Page
9. Dependent Claim 18: “The method of claim 12, further comprising storing the analytics in the memory of the cloud-based master server.”	150
C. GROUND 2A: DASGUPTA AND DARNELL RENDER CLAIMS 1 & 4-10 OBVIOUS	151
1. Motivation to Combine Dasgupta and Darnell	151
2. Independent Claim 1	153
3. Independent Claim 7	177
4. Claim 4: “The cloud-based computer system of claim 1, wherein the master application includes tools for enabling security features that limits access to one or more of the content sources or service providers based a user log-in belonging to the user.”	183
5. Claims 5 & 8: [Claim 5] “wherein the master server collects and stores history analytics based content data and content applications viewed by the user”; [Claim 8] “wherein the master server collects and stores history analytics based on usage of the remote computer.”	185
6. Claim 6: “The cloud-based computer system of claim 5, the master application includes a purchase tool that allows the user to purchase goods or services from the multiple content sources or service providers from master web-page.”	187
7. Claim 9: “The computer system of claim 8, wherein the master server uses the history analytics to select content data or the content applications based on a user log-in belonging to the user.”	189

TABLE OF CONTENTS
(continued)

	Page
8. Claim 10: “The computer system of claim 7, wherein the user interface is an icon-based master web-page.”	190
D. GROUND 2B: DASGUPTA, DARNELL, AND KIM FURTHER RENDER CLAIMS 2, 3, AND 11 INVALID AS OBVIOUS	191
1. Motivation to Combine Kim with Dasgupta and Darnell.....	192
2. Claim 2: “The cloud-based computer system of claim 1, wherein the content sources or service providers include one or more personal content account and at least one or more social media account belonging to the user.”	195
3. Claims 3 & 11: [Claim 3] “wherein the personal content account is an e-mail account”; [Claim 11] “wherein one of the content sources or service providers includes an e-mail web-site belonging to the user.”	198
XII. '068 PATENT: DETAILED EXPLANATION OF THE GROUNDS	200
A. GROUND 1: DARNELL AND KIM RENDER CLAIM 1 OBVIOUS.	200
1. Motivation to Combine Darnell with Kim.....	200
2. Independent Claim 1.....	206
B. GROUND 2: DASGUPTA AND DARNELL RENDER CLAIM 1 OBVIOUS	240
1. Motivation to Combine Dasgupta and Darnell.....	240
2. Independent Claim 1.....	244

TABLE OF CONTENTS
(continued)

	Page
XIII. CONCLUSION	268
XIV. DECLARATION	269

EXHIBIT LIST

Exhibit	Description
1001	U.S. Patent No. 9,729,658 (“the ’658 patent,” IPR2025-00307) U.S. Patent No. 10,334,068 (“the ’068 patent,” IPR2025-00308)
1002	Declaration of Dr. Paul Martin, Ph.D. (“Martin”)
1003	File History of the ’658 (IPR2025-00307) File History of the ’068 (IPR2025-00308)
1004	U.S. Patent Publication No. 2008/0086484, filed Oct. 9, 2007, published April 10, 2008, to Benjamin G. Darnell et al. (“Darnell”)
1005	U.S. Patent Publication No. 2007/0016609, filed July 12, 2005, published Jan. 18, 2007, to Jane T. Kim et al. (“Kim”)
1006	U.S. Patent No. 8,640,052, filed Dec. 31, 2009, published June 30, 2011, issued January 28, 2014, to Mousumi Dasgupta et al. (“Dasgupta”)
1007	Executed Summons, Dkt. No. 14, <i>RazDog Holdings LLC v. Twitch Interactive, Inc.</i> , Case No. 24-cv-03638-JSC (N.D. Cal. filed Jun. 17, 2024)
1008	U.S. Patent Publication No. 2011/0078584, filed September 27, 2010, published March 31, 2011, to Daniel Ben Winterstein, et al. (“Winterstein”)
1009	Excerpt of File History of U.S. Patent Publication No. 2008/0086484 (“Darnell”) (Non-Final Office Action, dated Oct. 1, 2010)
1010	U.S. Patent Publication No. 2005/0165615, filed December 31, 2003, published July 28, 2005, to Nelson Minar (“Minar”)
1011	U.S. Patent Publication No. 2004/0093327, filed February 26, 2003, published May 13, 2004, to Darrell Anderson, et al. (“Anderson”)
1012	Amended Pretrial Order No. 1, <i>RazDog Holdings LLC v. Twitch Interactive, Inc.</i> , No. 24-cv-03638-JSC (N.D. Cal., Sept. 17, 2024), ECF No. 39

Exhibit	Description
1013	Complaint, <i>RazDog Holdings LLC v. Twitch Interactive, Inc.</i> , No. 24-cv-03638-JSC (N.D. Cal., June 17, 2024), ECF No. 1
1014	Excerpt of Preliminary Infringement Contentions from Razdog, Inc., <i>RazDog Holdings LLC v. Twitch Interactive, Inc.</i> , Case No. 24-cv-03638-JSC (N.D. Cal.), served Nov. 22, 2024
1015	Curriculum Vitae of Paul D. Martin, Ph.D.
1016	<i>Amazon EC2 Beta</i> , AMAZON WEB SERVICES BLOG (Aug. 29, 2006), [https://web.archive.org/web/20060829110501/http://aws.typepad.com/]
1017	<i>About Microsoft</i> , https://news.microsoft.com/about/ (last visited Dec. 15, 2024)
1018	<i>Reflecting on our ten year App Engine journey</i> , GOOGLE CLOUD BLOG, https://cloud.google.com/blog/products/gcp/reflecting-on-our-ten-year-app-engine-journey/ (last visited Dec. 15, 2024)
1019	Peter Mell & Tim Grance, <i>The NIST Definition of Cloud Computing</i> (National Institute of Standards and Technology, Version 15, Oct. 7, 2009), https://www.nist.gov/system/files/documents/itl/cloud/cloud-def-v15.pdf
1020	<i>The Benefits of Cloud Computing</i> (IBM Dynamic Infrastructure, July 2009), https://www.ibm.com/ibm/files/H300444G23392G14/13Benefits_of_Cloud_Computing_634KB.pdf
1021	<i>Cloud Computing and Sustainability</i> (Accenture Whitepaper, November 2010)
1022	RDF Site Summary (RSS) 1.0 (Oct. 11, 2010), [https://web.archive.org/web/20101011123954/https://web.resource.org/rss/1.0/spec]
1023	W3C RSS 1.0 News Feed Creation How-To (May 28, 2010), [https://web.archive.org/web/20100528134008/https://www.w3.org/2001/10/g lance/doc/howto]

Exhibit	Description
1024	<i>How Do RSS Feeds Work?</i> , RSS.com, https://rss.com/blog/how-do-rss-feeds-work/ (last visited Dec. 15, 2024)
1025	NetNewsWire.com (Sep. 7, 2009), [https://web.archive.org/web/20090907105218/https://netnewswire.com/]
1026	FeedDemon 3 – Free RSS Reader for Windows (Oct. 11, 2010), [https://web.archive.org/web/20101011081027/http://www.feeddemon.com/]
1027	NewsGator.com (Oct. 8, 2010), [https://web.archive.org/web/20101008110128/https://www.newsgator.com/]
1028	Google Reader Blog (Oct. 8, 2010), [https://web.archive.org/web/20101008110804/http://googlereader.blogspot.com/]
1029	RSS 0.90 Specification (Jul. 7, 2010), [https://web.archive.org/web/20100707223500/https://www.rssboard.org/rss-0-9-0]
1030	RDF Site Summary 1.0 Modules (Oct. 5, 2012), [https://web.archive.org/web/20121005062018/https://web.resource.org/rss/1.0/modules/]
1031	RDF Site Summary 1.0 Modules: Dublin Core (Sep. 28, 2010), [https://web.archive.org/web/20100928094047/https://web.resource.org/rss/1.0/modules/dc/]
1032	RDF Site Summary 1.0 Modules: Syndication (Oct. 11, 2010), [https://web.archive.org/web/20101011224813/https://web.resource.org/rss/1.0/modules/syndication/]
1033	RDF Site Summary 1.0 Modules: Content (Oct. 9, 2010), [https://web.archive.org/web/20101009185424/https://web.resource.org/rss/1.0/modules/content/]
1034	RSS 2.0 Specification (Oct. 11, 2010), [https://web.archive.org/web/20101011191039/https://www.rssboard.org/rss-specification]

Exhibit	Description
1035	N. Nottingham & R. Sayre, <i>The Atom Syndication Format, Request for Comments 4287</i> , NETWORK WORKING GROUP (Mar. 3, 2010), [https://web.archive.org/web/20100303155855/http://www.rfc-editor.org/rfc/rfc4287.txt]
1036	J. Gregorio & B. de Hora, <i>The Atom Publishing Protocol, Request for Comments 5023</i> , NETWORK WORKING GROUP (Feb. 20, 2010), [https://web.archive.org/web/20100220110543/https://www.rfc-editor.org/rfc/rfc5023.txt]
1037	Daniel Miessler, <i>RSS: Why We Should Just Call Them “Feeds” Instead of “RSS” Feeds</i> (Sep. 26, 2010), [https://web.archive.org/web/20100220110543/https://www.rfc-editor.org/rfc/rfc5023.txt]
1038	FEED Validator: Introduction to Atom (Jan. 23, 2010), [https://web.archive.org/web/20100123064931/https://validator.w3.org/feed/docs/atom.html]
1039	<i>Best Practices for XML sitemaps and RSS/Atom feeds</i> , Google Search Central Blog, https://developers.google.com/search/blog/2014/10/best-practices-for-xml-sitemaps-rssatom (last visited Dec. 15, 2024)
1040	<i>Awesome RSS</i> , GitHub, https://github.com/voidfiles/awesome-rss (last visited Dec. 15, 2024)
1041	Kyle Nazario, <i>How to Follow RSS Feeds in Chrome, Safari and Firefox</i> , TechNorms.com (Jul. 17, 2015), [https://web.archive.org/web/20150617232812/https://www.technorms.com/44712/follow-rss-feeds-in-chrome-safari-firefox]
1042	Sergey Tkachenko, <i>Google re-adds RSS support to the Chrome browser</i> , Winaero, https://winaero.com/google-re-adds-rss-support-to-the-chrome-browser/ (last visited Dec. 15, 2024)
1043	Chris Wetherell, <i>Google Reader Two Weeks</i> , Google Reader Blog (Dec. 15, 2005),

Exhibit	Description
	[https://web.archive.org/web/20051215032412/https://googlereader.blogspot.com/2005/10/google-reader-two-weeks.html]
1044	Andy Wibbels, <i>How to Use Google Reader</i> (Oct. 24, 2005), [https://web.archive.org/web/20051024012030/http://andywibbels.com/flash/google_reader.htm]
1045	Andrew Hitchcock Blog (Oct. 25, 2005), [https://web.archive.org/web/20051024012030/http://andywibbels.com/flash/google_reader.htm]
1046	<i>Analytics</i> , Cambridge Dictionary, https://dictionary.cambridge.org/us/dictionary/english/analytics (last visited Dec. 14, 2024)
1047	<i>Analytics</i> , Lexico powered by Oxford English Dictionary (Aug. 10, 2020) [https://web.archive.org/web/20200810063114/https://www.lexico.com/en/definition/analytics]
1048	SHELLEY POWERS, PRACTICAL RDF, 10-11 (O'Reilly & Associates, 1 st ed. 2003) (excerpts)
1049	WebTV, https://webtv.zone/ (last visited Dec. 15, 2024)
1050	<i>Instructions for using Web TV</i> , ChicagoNet (Jun. 21, 2010), [https://web.archive.org/web/20100621001111/http://www.chicagonet.net/webtv.htm]

I, Paul D. Martin , do hereby declare as follows:

I. INTRODUCTION

1. I am over the age of 18 and am competent to write this Declaration.

2. I make this declaration based upon my own personal knowledge and, if called upon to testify, would testify competently to the matters stated herein.

3. I have been asked to provide my expert opinion in connection with the above-referenced *inter partes* review proceedings in the United States Patent and Trademark Office for U.S. Patent Nos. 9,729,658 (“the ’658 patent”) and 10,334,068 (“the ’068 patent”) (together, “Challenged Patents”)¹ and prior art references relating to its subject matter. I have personal knowledge or have developed knowledge of these technologies based upon my education, training, or experience as set forth below.

4. This declaration is a statement of my opinions on issues relating to the patentability of claims 1–18 of the ’658 patent and claim 1 of the ’068 patent (together, “the Challenged Claims”). As I explain more fully below, it is my opinion

¹ I understand that each Challenged Patent has been filed as Exhibit 1001 in each respective proceeding, and their prosecution history has been filed as Exhibit 1003. For this reason, I will cite each Challenged Patent by its shorthand title, and the prosecution history as [Shorthand Patent No.] *Prosecution History*.

that all of the Challenged Claims would have been obvious to a person of ordinary skill in the art (“POSITA”) at the time of the alleged invention.

5. I am the Chief Scientist at Harbor Experts, Inc. and a part-time Lecturer in the Department of Computer Science at Johns Hopkins University. I hold B.S., M.S.E., and Ph.D. degrees from Johns Hopkins University, with all degrees, including my doctorate, being in computer science. My current curriculum vitae (CV) is attached as EX1015.

6. I am an expert in the technical subject matter areas relevant to this case. All opinions and facts stated in this report are true and correct to the best of my knowledge. If called upon to testify, I could and would testify to the truth of the following. The company for which I work, Harbor Experts, is being compensated at the rate of \$920 per hour for time that I spend on this case. Harbor Experts is also being reimbursed for my reasonable and customary expenses. My compensation is not dependent in any way on the results of the lawsuit or the substance of my testimony.

II. BACKGROUND AND QUALIFICATIONS

A. Education

7. I have seventeen (17) years of experience in electrical engineering and computer science and in cloud-based networking technology. I attended Johns Hopkins University from 2007 to 2016, earning a Bachelor of Science (B.S.) in

Computer Science, a Master of Science in Engineering (M.S.E.) in Computer Science, and a Doctor of Philosophy (Ph.D.) in Computer Science.

8. I also have thirteen (13) years of teaching experience. In 2011, I was a course assistant at Johns Hopkins University for the course “Practical Cryptographic Systems.” Between 2012 and 2014, I was a teaching assistant for the course “Security and Privacy” at Johns Hopkins University. In 2015, I became an Instructor at Johns Hopkins for the course “Introduction to Hardware Hacking.” In 2023, I became an Instructor for the course titled “Security and Privacy in Computing.”

9. While pursuing my masters and Ph.D. between 2011-2016, I was a research assistant at Johns Hopkins University.

10. I am currently a part-time lecturer in the Department of Computer Science at Johns Hopkins University. I currently teach classes on computer and network security and applied cryptography. In the past I have taught classes on the topic of hardware hacking, including analyzing, modifying, and repairing computer hardware and embedded devices. I also occasionally substitute teach classes on network security and cryptography.

11. My classes cover a diverse array of topics including hardware and software security design, analysis and reverse engineering for vulnerability assessment; hardware and software-based attacks on computer components including RAM and CPUs; applied cryptography; computer architecture; computer

networking; and component-level repair and modification. My classes also utilize hardware and software emulation and virtualization techniques in order to present students with a variety of real-world environments in which to learn and perform hands-on projects.

B. Career

12. While pursuing my undergraduate degree between 2007 and 2011, I worked as a freelance programmer at a hardware repair shop at Brandeis University and at the Digital Research and Curation Center (DRCC) at Johns Hopkins University. I also worked as a Security Intern at a company called Independent Security Evaluators between 2009 and 2011. While I was pursuing my masters degree between 2011 and 2013, I worked as a penetration tester for the University of Michigan's Consortium for Political and Social Research. While pursuing my Ph.D. between 2013 and 2016, I worked as a Research Scientist for Harbor Experts (f/k/a Harbor Labs).

13. In 2018, I was promoted to Director of Firmware Security and Senior Research Scientist at Harbor Experts. In 2024, I was promoted to Vice President of Research and Technology, Senior Research Scientist at Harbor Experts.

14. I began working in the software industry in 2008, when I obtained my first independent consulting client—Brandeis University. I created a program for their computer repair shop's ticketing system to interface with a label printer. This

enabled the computer repair shop to automatically print labels associated with tickets in order to easily identify customers' devices.

15. Shortly after finishing my engagement with Brandeis University, I began working for the Johns Hopkins University Digital Research and Curation Center as a student programmer. There, I created a batch importer for a digital archiving system called DSpace in order to enable hundreds of digitally archived works per week to automatically be added to the repository. I next ported a web interface for statistics tracking and visualization to the same digital archiving system in order to allow library staff to more easily visualize and understand data about how the digital archiving system was being used through a common web-based interface.

16. In 2009, I began working for a security consulting company called Independent Security Evaluators in Baltimore, Maryland. I worked on a wide variety of security and privacy projects including projects to test and break the DRM scheme of a magazine distribution application for IOS and Android (at the behest of the developer of said application), projects to test implementation code for cryptographic secret splitting, projects to assist with fuzz testing for security vulnerabilities of a variety of software applications, software development projects to automate testing of antivirus and antimalware solutions, and a variety of other security-related projects. Of note, I designed automation systems to create and restore virtualized snapshots of Windows machines on which I tested antivirus

software on actual malware samples as part of a project for Consumer Reports. I worked for ISE as an intern throughout the semesters and summers until August 2011. In 2010, I also designed a security architecture for a large-scale digital curation system meant to be a major inter-university initiative to create a successor to existing digital curation systems that could be used for decades.

17. In spring 2011, I completed my bachelor's in computer science at Johns Hopkins University, and I also retained an independent consulting client, the University of Michigan, for whom I performed a penetration test and security assessment of a cloud-based research system that they planned to deploy.

18. In February 2016, I joined Harbor Labs full-time as a research scientist. In January 2019, I was promoted to Director of Firmware Analysis. In December 2023, I was promoted to Vice President of Applied Research and Technology. In October 2024, the division of Harbor Labs that I managed became Harbor Experts. I became Chief Scientist at Harbor Experts during this time. At Harbor Experts, I manage client engagements related to all of the analysis work that Harbor Labs performs. I am also lead Harbor Experts' research projects.

19. A substantial portion of my role at Harbor Experts is in managing source code review engagements. As part of this work, I have reviewed software systems of varying sizes, often totaling in the millions or billions of lines of code. I have reviewed products in the security space, television-based set top boxes,

network appliances, numerous web-based enterprise systems, email management systems, telephony products, embedded system bootloaders, social network platforms, virtualization platforms and countless other products.

20. Prior to joining Harbor Experts, I was also the technical and development lead for a firmware security analysis engine for a product developed at Harbor Labs called Firmware IQ. Firmware IQ is designed to analyze the firmware of embedded devices for security vulnerabilities in an automated fashion. In one configuration of the product, a developer uploads a firmware image to a web-based portal which uses a broker to forward the firmware to an engine for analysis. The engine unpacks the firmware, breaks it into its constituent components, and then performs more than a hundred automated security checks in order to find vulnerabilities in the firmware image. The results are then reported through JSON to a web-based presentation system which displays them in an interactive fashion.

21. A more detailed list of my engagements can be found on my attached CV.

C. Research

22. In spring 2011, I completed my bachelor's in computer science at Johns Hopkins University. In fall 2011, I began working towards my Ph.D. in computer science, also at Johns Hopkins University, and I began researching computer security and privacy systems in order to both design novel technologies for securing

computing systems and to also bridge the interface gap between users and their technology. As such, my research focused not only on developing technical solutions to novel security and privacy systems but to also presenting these solutions in a way that non-technical users could understand. Part of my research also focused on non-deterministically generating virtualized computer networks in order to provide malware education environments for students. I also developed an emulator for analyzing malware code segments as part of a student project with the NSA. This emulator was integrated into an IDA Pro plugin and was able to observe how binary instruction snippets taken from malware samples modified their computing environment.

23. During my tenure as a Ph.D. student at Johns Hopkins, I was fully funded as a research assistant and/or teaching assistant throughout all of my semesters. I was a member of the Upsilon Pi Epsilon computer science honor society, as well as its treasurer for the 2014-2015 school year. I was also a member of the Johns Hopkins chapter of the Association for Computing Machinery. I earned an award for outstanding teaching assistant in the Computer Science department for the year 2014-2015. I co-instructed a short course called, “Introduction to Hardware Hacking,” with a colleague, Dr. Michael Rushanan, which was the highest-rated course in the computer science department (based on student reviews) during the winter session in which it was offered. In this course, we offered lessons on a variety

of topics including modifying game consoles and device firmware; electronics repair; binary analysis and modification, network traffic analysis; and web-based vulnerability assessment and exploitation.

24. I represented Johns Hopkins University as a student speaker at numerous open houses and admitted student days to help introduce prospective students to the computer science department. I was selected to speak to the external advisory board for the computer science department to provide insight into the student experience in the department. I served the department as the faculty liaison czar for several semesters, in which I attended and documented faculty meetings, which I then summarized for the students in the department. I also mentored and supervised a high school student on a year-long student project in software design and I supervised a variety of undergraduate research assistants on some of my research projects.

25. I finished my Ph.D. by successfully defending my dissertation, entitled “Securing Medical Devices and Protecting Patient Privacy in the Technological Age of Healthcare,” on February 12, 2016, at the age of 26. My doctoral thesis tells the story of a secure healthcare practice of the future in which technology is seamless to use for healthcare providers while providing a much higher standard of security than is typical today. It focuses on the juxtaposition of usability and security with an emphasis on simplicity, automation, and error-proofing of security controls.

26. During my doctorate and afterward, I have produced, peer-reviewed, and published research in areas such as fingerprinting, anomaly detection, multifactor authentication, and embedded systems security. In many of these cases, I focused not only on designing novel security systems but also on designing usable web-based security systems for controlling them and understanding their data output. In fact, some of my research has formed the basis for commercial products and services. For example, in my work on integrated audit and access control, which was funded in part by Accenture Labs, I developed a Hadoop-based application to perform large-scale statistical analysis of audit logs from an electronic medical record system. As part of this work, I also designed a web application to automatically produce human-readable reports and graphs for use in consulting. Accenture subsequently patented this technology.

27. Similarly, in my research at Applied Communication Sciences in 2013, I designed and implemented a web-based traffic visualization dashboard and analysis system for field area smart grid networks that could be used to quickly gain an understanding of the current state of a SmartGrid network as well as to detect unexpected anomalies in the network. This dashboard was able to aggregate and visualize multiple data streams in real-time. Applied Communication Sciences subsequently patented this work and continued to build on the project. To my

knowledge, they sold and/or still sell this product as part of their SecureSmart Managed Security Service product offering.

28. I have worked on and published two research projects related to improving the quality and ease-of-use of authentication technologies, especially in healthcare settings. In one project, I and my co-authors designed an indoor location tracking system consisting of unspoofable Bluetooth Low Energy beacons. These beacons were placed around a building and mapped to a backend such that a user could report which beacons he or she was within range of in order to be used as a secondary authentication mechanism for accessing patient medical records. In the way that we designed and envisioned the system, a doctor would log into a mobile device which would connect to a web-based backend. As the doctor walked past patient rooms, he or she would be automatically presented with medical records of nearby patients. In another project, I and my co-authors designed an authentication bracelet that would receive a Kerberos ticket upon login to a modified computer terminal through use of low-energy electrical signals transmitted over the wearer's skin. Upon use of other terminals throughout the hospital, this Kerberos ticket could be sent back over the wearer's skin to the custom contact on the terminal in order to allow the user to login without a password. The bracelet was also designed to immediately lose the cryptographic secret upon being removed from the user.

29. My recent research has focused primarily on embedded systems security with an emphasis on binary analysis, anomaly detection and automated security enforcement. In one case, I designed a security system that can be soldered directly to the CPU of embedded devices in order to perform control-flow integrity for purposes of profile building and enforcement. In another case, I designed a system to automatically discern name and version information from binaries on embedded devices in order to build a software profile of the platform configuration of the device which can then be cross-referenced with a vulnerability database. I have also written a patent on this specific technique.

30. As of 2023 I am a member of the program committee for the IEEE Symposium on Security and Privacy. As a member of this committee, I peer review research that has been submitted for publication at the Symposium. I have recently reviewed research on memory forensics, side channel attacks, password vaults, differential privacy, computational theory, memory module design, and hardware security.

D. Teaching

31. I am a part-time lecturer in the Department of Computer Science at Johns Hopkins University. I currently teach classes on computer and network security and applied cryptography. In the past I have taught classes on the topic of hardware hacking, including analyzing, modifying, and repairing computer

hardware and embedded devices. I also occasionally substitute teach classes on network security and cryptography.

32. My classes cover a diverse array of topics including hardware and software security design; analysis and reverse engineering for vulnerability assessment; hardware and software-based attacks on computer components including RAM and CPUs; applied cryptography; computer architecture; computer networking; and component-level repair and modification. My classes also utilize hardware and software emulation and virtualization techniques in order to present students with a variety of real-world environments in which to learn and perform hands-on projects.

E. Patents

33. I am a named inventor on six patents. A full listing of these patents can be found on my curriculum vitae, attached as EX1015.

F. Publications and Professional Associations

34. Over the course of my career, I have authored and co-authored approximately 5 publications on various aspects of cloud-based networking technology and security/cryptography, as noted in my curriculum vitae.

G. Compensation

35. I have no financial interest in either party or in the outcome of this proceeding. I am being compensated for my work as an expert on an hourly basis.

My compensation is not dependent on the outcome of these proceedings or the content of my opinions.

III. MATERIALS CONSIDERED

36. The materials that I considered and relied upon in preparing my Declaration and forming my opinions include all exhibits cited in this Declaration and the Petition, including each of the Challenged Patents, their file histories, and all of the relevant prior art.

37. I also have relied on my academic and professional experience in reaching the opinions expressed in this Declaration.

IV. RELEVANT LEGAL STANDARDS

38. I am not an attorney. In preparing and expressing my opinions and considering the subject matter of the Challenged Patents, I am relying on certain basic legal principles that counsel have explained to me. These principles are discussed below.

39. I understand that for an invention claimed in a patent to be found patentable, it must be, among other things, new and not obvious in view of what was known before the invention was made.

40. I understand the information that is used to evaluate whether an invention is new and not obvious is generally referred to as “prior art” and generally

includes patents and printed publications (e.g., books, journal publications, articles on websites, product manuals, etc.).

41. I understand that in this proceeding the Petitioner has the burden of proving that the claims of the Challenged Patents are anticipated by or obvious in view of the prior art by a preponderance of the evidence. I understand that “a preponderance of the evidence” is evidence sufficient to show that a fact is more likely true than it is not.

42. I understand that prior art to the Challenged Patents in this proceeding includes patents and printed publications in the relevant art that predate the proper priority date of the alleged inventions.

43. I understand that terms appearing in the patent claims are to be interpreted according to their “ordinary and customary meaning” in an IPR proceeding. In determining the ordinary and customary meaning, the words of a claim are first given their plain meaning as they would have been understood by a POSITA at the time of the alleged invention, in light of the specification and file history. I understand that treatises and dictionaries may be consulted, albeit under limited circumstances, to determine the meaning attributed by a POSITA at the time of the alleged invention. I have followed this approach in my analysis and have applied the ordinary and customary meaning of those terms throughout my analysis in this Declaration.

44. I understand that there are two ways in which prior art may render a patent claim unpatentable. First, the prior art can be shown to “anticipate” the claim. Second, the prior art can be shown to have made the claim “obvious” to a person of ordinary skill in the art. In this declaration I focus on “obviousness.” My understanding of the legal standard for “obviousness” is set forth below.

45. I have been informed that a claimed invention is unpatentable as obvious if the differences between the invention and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which the subject matter pertains. I have also been informed by counsel that the obviousness analysis considers factual inquiries including the level of ordinary skill in the art, the scope and content of the prior art, and the differences between the prior art and the claimed subject matter.

46. Specifically, I understand that the obviousness question requires consideration of four factors (although not necessarily in the following order):

- The scope and content of the prior art;
- The differences between the prior art and the claims at issue;
- The knowledge of a person of ordinary skill in the pertinent art; and
- Whatever objective factors indicating obviousness or non- obviousness may be present in any particular case.

47. I have been informed by counsel that the Supreme Court has recognized several rationales for combining references or modifying a reference to show obviousness of claimed subject matter. Some of these rationales include the following: (a) combining prior art elements according to known methods to yield predictable results; (b) simple substitution of one known element for another to obtain predictable results; (c) use of a known technique to improve a similar device (method, or product) in the same way; (d) applying a known technique to a known device (method, or product) ready for improvement to yield predictable results; (e) choosing from a finite number of identified, predictable solutions, with a reasonable expectation of success; and (f) some teaching, suggestion, or motivation in the prior art that would have led one of ordinary skill to modify the prior art reference or to combine prior art reference teachings to arrive at the claimed invention.

48. I further understand that certain factors may support or rebut the obviousness of a claim. I understand that such secondary considerations include, among other things, commercial success of the patented invention, skepticism of those having ordinary skill in the art at the time of invention, unexpected results of the invention, any long-felt but unsolved need in the art that was satisfied by the alleged invention, the failure of others to make the alleged invention, praise of the alleged invention by those having ordinary skill in the art, and copying of the alleged invention by others in the field. I understand that there must be a nexus—a

connection—between any such secondary considerations and the alleged invention. I also understand that contemporaneous and independent invention by others is a secondary consideration tending to show obviousness.

49. I am not aware of any allegations by the named inventor or any assignee of the Challenged Patents that any secondary considerations are relevant to the obviousness analysis of any Challenged Claims.

50. I understand that a patent may define terms, such that the patentee acts as their own lexicographer. I also understand that the definitions in the patent must be clearly set forth and not be something other than the ordinary and customary meaning of the term.

51. I understand that a petition must identify, for any terms with means-plus-function limitations, the specific portions of the specification that describe the structure, material, or acts corresponding to each claimed function. I also understand a rebuttable presumption exists that without use of the term, “means,” a claim term is not interpreted as means-plus-function, but that other nonce words like “component” have been found to satisfy the rebuttable presumption and invoke a means-plus-function claim construction of such a claim term.

52. I also understand that a petitioner seeking to demonstrate that the prior art discloses a means-plus-function limitation must prove that the corresponding

structure—or an equivalent—was present in the prior art and performing the same claimed function.

53. I understand that the standard for determining whether the prior art discloses the corresponding structure, or an equivalent, is whether there are insubstantial differences, which means that if the assertedly equivalent structure performs the claimed function in substantially the same way to achieve substantially the same result as the corresponding structure described in the specification, it is equivalent.

54. I understand that proving structural equivalence does not require a component-by-component analysis, but rather that it is sufficient to show structure, materials, or acts that perform the claimed function in substantially the same way to achieve substantially the same result because the claim limitation is the overall structure corresponding to the claimed function and not individual components.

55. I understand that for a claim to be valid, its scope must be definite, and that improperly introducing a term can lead to an unascertainable scope of a term, rendering it indefinite. I understand that any claim element first addressed in the claims by the antecedent basis, “a” may then be identified later in the claims by the antecedent basis, “the.” I understand that failing to follow this convention may lead to ambiguity in the identification of the particular claim element in question, leading to an indefinite scope of the claim’s term. For purposes of this proceeding, I

identified and provided my best understanding of any such terms despite such ascertainable scope.

V. THE '658 AND '068 PATENTS

56. The '658 patent is titled “System for Managing Web-Based Content Data and Applications.” '658 patent, Cover. The '068 patent is titled “System for managing web-based swipe module tool and software for scrolling and paging through content data and applications.” '068 patent, Cover. The '068 patent is a continuation of the '658 patent. *Id.* The '068 patent shares its specification with the '658 patent, thus my analysis of the specifications is the same. For clarity, I will cite to both patents in this section.

57. The system of the Challenged Patents purportedly enables users to interact with multiple data sources, accessing, managing, and navigating content from a centralized “master web-page” on remote devices such as computers, tablets, and smartphones. '658 patent, 2:52-59; '068 patent, 2:60-67.

A. Background

58. The Challenged Patents aim to solve challenges associated with managing and accessing cloud-based content from multiple sources. '658 patent, 1:39-62; '068 patent, 1:47-2:3. In traditional cloud-based systems, users often need to log into each content source separately. '658 patent, 1:53-56; '068 patent, 1:61-64. Furthermore, the '658 patent states that managing the multiple content sources

from a single user interface was not provided. '658 patent, 1:56-58; '068 patent, 1:64-66.

59. As I explain below, Cloud-based computing was well-known at the time of the Challenged Patents. In these cloud-based computing systems, a user is not required to know what types of resources are used and how tasks are accomplished because the computing assets are remote and in the cloud. '658 patent, 1:39-43; '068 patent, 1:47-51. The Challenged Patents also admit “[m]ost cloud-based computing systems and architectures include a master sever that networks to other service provider servers and streams content data to the master server that is then accessed through the remote computing device.” '658 patent, 1:47-50; '068 patent, 1:55-58. This can be “any kind of content data and/or services [delivered] in real-time...typically over the internet.” '658 patent, 1:43-46; '068 patent, 1:47-54.

60. The Challenged Patents explain that multiple logins are required to access cloud-based content from multiple sources. '658 patent, 1:53-58; '068 patent, 1:61-66. The inventors allegedly solve this problem by providing a “single user interface” that allows a user to access, manage, and migrate though content from various sources. '658 patent, 1:58-62, 2:30-34; '068 patent, 1:66-2:3, 2:38-42.

61. But the Challenged Patents acknowledge known methods to “migrate through content data using a scrolling-type feature or tool.” '658 patent, 1:66-2:2; '068 patent, 2:7-10. They describe two types of scrolling mechanisms. The first

allows a user to “swipe[] through content data contained in different files” on a local device. ’658 patent, 2:2-15; ’068 patent, 2:10-23. The second is a “plug-in and play” software that allows a user to page through content in a “slide-like” presentation. ’658 patent, 2:16-29; ’068 patent, 2:24-37.

62. As I explain below, content aggregation services, such as RSS aggregators were well-known in the art by the time of the Challenged Patents. These systems were able to present content from multiple sources in a single user interface.

B. Specification

63. To solve the identified problems, the Challenged Patents propose a “cloud-based computer system and architecture” designed to allow users to manage and migrate through content data from multiple sources using a “single user interface” accessible from a remote device. ’658 patent, Abstract, 2:30-37; ’068 patent, Abstract, 2:38-45. The solution uses a “master server” acting as a “single point access” to various “content data sources or service providers.” ’658 patent, Abstract, 2:30-37; ’068 patent, Abstract, 2:38-45. This architecture includes a “master web-page” that users can view from a remote computing device, enabling them to “scroll or page through the content data and/or content applications from the multiple content data sources or service providers in real time.” ’658 patent, Abstract; ’068 patent, Abstract.

64. The system's master web-page features an “icon-based” design that allows users to interact with multiple content sources simultaneously. ’658 patent, 2:38-42; ’068 patent, 2:46-50. Additionally, a “slide tool” incorporated into this interface “allows a user to Scroll or page through the content data and/or content applications in real time from master server on a master web-page from the remote computer device.” ’658 patent, 2:60-65; ’068 patent, 3:1-6.

65. The alleged invention “provides a unique web-based database, user interface, remote slide tool and the user history analytics which is used as a comprehensive solution to manage content data and/or applications from multiple content data sources.” ’658 patent, 3:33-37; ’068 patent, 3:44-48. Figure 1 of the Challenged Patents shows the architecture of the disclosed cloud-based computer system:

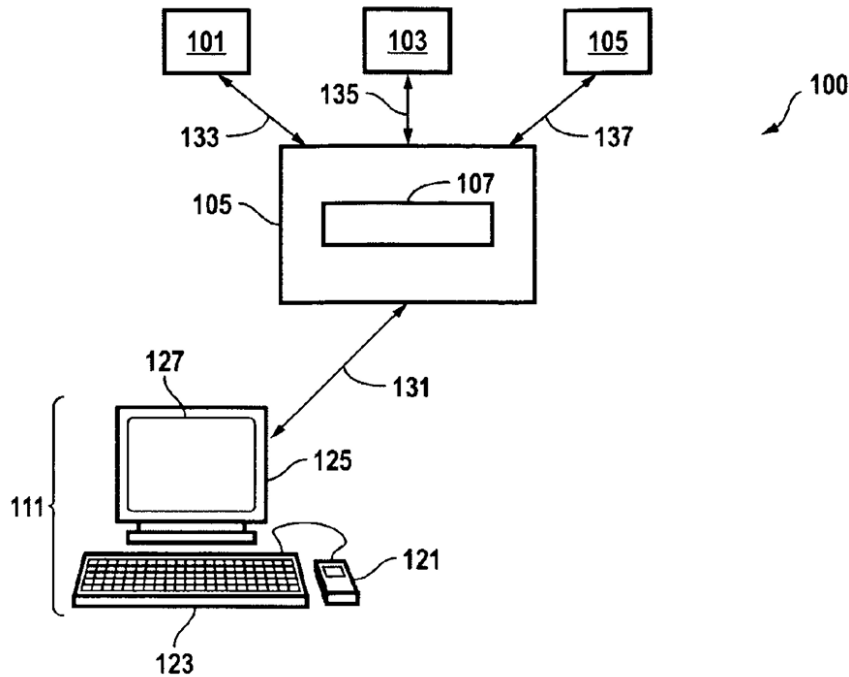


FIG. 1

'658 patent, Fig. 1; '068 patent, Fig. 1.

66. “[C]ontent data sources or service provider servers 101, 103, and 105 are in communication or linked to...master server 105.” ’658 patent, 4:41-46; ’068 patent, 4:52-57. Content is aggregated and organized by master application 107 and accessed by a user using a remote computing device, such as desktop computer with a monitor or display. ’658 patent, 4:50-67; ’068 patent, 4:61-5:11. The preferred embodiment is “an icon-based master web-page viewable over the master server” with a slide tool that “allows a user to scroll or page through the content data and/or content applications in real time from master server on a master web-page from the remote computer device.” ’658 patent, 2:40-42, 2:62-65; ’068 patent, 2:48-50, 3:3-6. Figures 2A-2C illustrate content that is displayed and scrollable on the remote

computing device. Figure 2B, for example, shows “a slide tool or scrolling tool indicated by the arrow 226 operable from the computing device 111.” ’658 patent, 5:8-11; ’068 patent, 5:19-22.

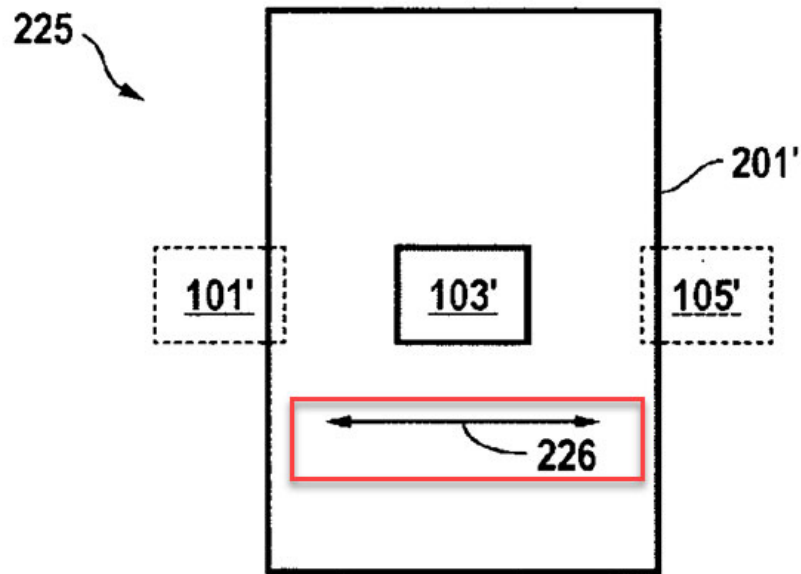


FIG. 2B

’658, Fig. 2B; ’068 patent, Fig. 2B.²

67. The specifications describe a “remote tool” that “is configured [to] deliver a subset of content data or applications to a third party enabled remote computing device to allow access only to the content data and/or applications that match an authorized user and/or target audience.” ’658 patent, 3:28-33; ’068 patent, 3:39-43.

² Annotations added throughout unless otherwise noted.

68. The specifications also disclose a “tool box” that “provides for the capability to include lock-out controls, collect history analytics and have subsets of accessible content data or content applications.” ’658 patent, 5:48-52; ’068 patent, 5:59-63. The Challenged Patents further explain the “user history analytics” can be recorded, stored, and used to tailor content—such as “promotions, advertisement, news and other information”—based on the user’s activities. ’658 patent, 3:13-23; ’068 patent, 3:23-33. The system is also designed to be customizable, enabling users to “add or subtract any number of content data sources or service providers” based on their specific needs and preferences. ’658 patent, 3:5-9; ’068 patent, 3:14-19.

C. The Patent Claims

1. The ’658 Patent

69. The ’658 patent claims the alleged invention just described in eighteen claims: three independent claims and fifteen dependent claims.

70. I will address all of the claims in my analysis below, but I would first like to explain the substantial overlap between the claims in the following tables to give my cross-references in Sections XI-XI.D below context.

71. Below, I have prepared a table comparing independent claims 1 and 7 on a limitation-by-limitation basis. I emphasize the differences between the two

independent claims by underlining and bolding the portions of independent claim 7 that differ from independent claim 1.

<u>Claim 1</u>	<u>Claim 7</u>
1. A cloud-based computer system comprising:	7. A computer system with
[1a] master server with a processor and memory, wherein the master server accessible over the internet/Intranet from remote computer devices by a user;	[7a] a master server with a memory and a processor <u>for storing and executing a master application</u>
[1b1] a master application stored in the memory of the master server and that executes on the processor of the master server and that includes software coded for organizing content data and content applications from multiple content sources or service providers pre-selected by the user	[7b] – <i>see [7b] below mapped to elements [1d] and [1e].</i>
[1b2] and streamed to the master server simultaneously in real time from their respective servers on a master web-page and are viewable from the remote computer devices through the master server on the master web-page,	[7c] wherein the content data or content applications are simultaneously streamed to the master server from servers of multiple content data sources or service providers over the internet in real time <u>to an account belonging to the user,</u>
[1b3] wherein the master application programs a processor on the remote computer devices to operate and execute a slide tool program for scrolling or paging through the content data or the content applications from the multiple content sources or service providers simultaneously and through the master server in real time from the master web-page and from the remote	[7b] <u>with a remote tool box feature</u> that programs a processor on a remote computer with a slide tool program for scrolling or paging through content data or content applications from multiple content data sources or service providers <u>enabled by a user</u> simultaneously on a web-page created by the master application and hosted

<u>Claim 1</u>	<u>Claim 7</u>
computer by operating the slide tool program from the remote computer; and	on the master server using hardware on remote computer,
[1c1] a remote tool box feature application that allows the user to write code, select code or delete code to enable and disable tool features that control how sub-sets of the content data or the content applications from the multiple content sources or service providers are received, displayed or organized,	[7d] wherein the remote tool box feature allows the user to write code, select code or delete code to enable and disable tool features that control how sub-sets of the content data or the content applications from the multiple content sources or service providers are received, displayed or organized
[1c2] wherein the master server collects analytics based which tool features are enabled to target specific products, services or information to the user.	[7e] and wherein the master server collects history analytics based which tool features are enabled to target specific products, services or information to the user.

72. As one can see, the limitations of claims 1 and 7 almost entirely overlap with each other. I will rely on this table to present consolidated arguments below.

73. Independent claim 12 also substantially overlaps with independent claim 1, but because of how the claim language is phrased and presented, I address that independent claim separately in Ground 1B below.

74. Some of the dependent claims also overlap substantially, such as dependent claims 5 and 8 and dependent claims 3 and 11.

75. Below, I have prepared two tables comparing each of the two pairs of those respective four dependent claims:

<u>Claim 5</u>	<u>Claim 8</u>
<u>The cloud-based computer system of claim 1</u> , wherein the master server collects and stores history analytics based <u>content data and content applications viewed by the user.</u>	<u>The computer system of claim 7</u> , wherein the master server collects and stores history analytics <u>based on usage of the remote computer.</u>

<u>Claim 3</u>	<u>Claim 11</u>
<u>The cloud-based computer system of claim 2</u> , wherein the personal content account is <u>an e-mail account.</u>	<u>The computer system of claim 7</u> , wherein one of the content sources or service providers includes <u>an e-mail web-site belonging to the user.</u>

76. As one can see, claims 5 and 8, and 3 and 11 are, respectively, nearly identical except for the underlying independent claim from which they depend. I grouped my analysis of those claims together below accordingly as well.

77. The remainder of the dependent claims of the '658 patent are addressed separately in the respective grounds in which they are challenged.

2. The '068 Patent

78. The '068 patent has a single claim. I have reviewed this claim and find that it is nearly identical to claim 1 of the '658 patent. The only differences are that “by the user” is deleted from claim limitation [1b1] and a user is not required for “code to be written, selected or deleted” in [1c1]. I provide a table below comparing the claims.

<u>'658 Patent Claim 1</u>	<u>'068 Patent Claim 1</u>
1. A cloud-based computer system comprising:	1. A cloud-based computer system comprising:
[1a] master server with a processor and memory, wherein the master server accessible over the internet/Intranet from remote computer devices by a user;	[1a] master server with a processor and memory, wherein the master server accessible over the internet/Intranet from remote computer devices by a user;
[1b1] a master application stored in the memory of the master server and that executes on the processor of the master server and that includes software coded for organizing content data and content applications from multiple content sources or service providers pre-selected <u>by the user</u>	[1b1] a master application stored in the memory of the master server and that executes on the processor of the master server and that includes software coded for organizing content data and content applications from multiple content sources or service providers pre-selected
[1b2] and streamed to the master server simultaneously in real time from their respective servers on a master web-page and are viewable from the remote computer devices through the master server on the master web-page,	[1b2] and streamed to the master server simultaneously in real time from their respective servers on a master web-page and are viewable from the remote computer devices through the master server on the master web-page,
[1b3] wherein the master application programs a processor on the remote computer devices to operate and execute a slide tool program for scrolling or paging through the content data or the content applications from the multiple content sources or service providers simultaneously and through the master server in real time from the master web-page and from the remote computer by operating the slide tool program from the remote computer; and	[1b3] wherein the master application programs a processor on the remote computer devices to operate and execute a slide tool program for scrolling or paging through the content data or the content applications from the multiple content sources or service providers simultaneously and through the master server in real time from the master web-page and from the remote computer by operating the slide tool program from the remote computer; and
[1c1] a remote tool box feature application that allows <u>the user to</u>	[1c1] a remote tool box feature application that allows <u>code to be</u>

<u>'658 Patent Claim 1</u>	<u>'068 Patent Claim 1</u>
<u>write code, select code or delete code</u> to enable and disable tool features that control how sub-sets of the content data or the content applications from the multiple content sources or service providers are received, displayed or organized,	<u>written, selected or deleted</u> to enable and disable tool features that control how sub-sets of the content data or the content applications from the multiple content sources or service providers are received, displayed or organized,
[1c2] wherein the master server collects analytics based which tool features are enabled to target specific products, services or information to the user.	[1c2] wherein the master server collects analytics based which tool features are enabled to target specific products, services or information to the user.

79. As shown above, claim 1 of the '068 patent omits “by the user,” which modifies “pre-selected” in claim [1b1]. Further, in claim [1c1] the “user” is removed. Thus, in my opinion, claim 1 of the '068 patent is broader than claim 1 of the '658 patent. Accordingly, any prior art or prior art combination that renders claim 1 of the '658 patent obvious would also render claim 1 of the '068 patent obvious.

D. Prosecution History of the '658 Patent

80. The application leading to the '658 patent was filed on October 11, 2011, and claims priority to provisional applications filed on February 22, 2011, and October 12, 2010. '658 patent, Cover. After extensive prosecution history, the '658 patent issued on August 8, 2017.

81. Over six years, the Examiner repeatedly rejected all pending claims because various prior art references independently taught the claimed “master server” and “master application.” *See generally, '658 Prosecution History, 29-47,*

68-87, 104-20, 141-61, 183-204, 225-49, 277-301, 359-86. To overcome the repeated rejections, PO amended the independent claims to recite:

a remote tool box feature application that allows the user to write code, select code or delete code to enable and disable tool features that control how sub-sets of the content data or the content applications from the multiple content sources or service providers are received, displayed or organized, wherein the master server collects analytics based which tool features are enabled to target specific products, services or information to the user.

Id., 415-19. Applicant argued that the amendment was supported by the provisional applications. *Id.*, 421-22. The amendment overcame the Examiner's rejections and led to allowance. *Id.*, 430-431.

E. Prosecution History of the '068 Patent

82. The application leading to the '068 Patent was filed on June 26, 2017, and claims priority to provisional applications filed on February 22, 2011, and October 12, 2010. '068 Patent, Cover. After two rejections, the '068 Patent issued on June 25, 2019.

83. All pending claims were rejected for double patenting and because various prior art references taught the claimed "master server" and "master application." *See generally*, '068 *Prosecution History*. To overcome the rejections,

Applicant cancelled all claims and added pending claim 34, which was allowed. *Id.*, 169, 186-193. The Examiner stated:

the main reason for allowance in conjunction with all the other limitation are the claimed feature of a remote tool box feature application that allows code to be written, selected or deleted to enable and disable tool features that control how sub-sets of the content data or the content applications from the multiple content sources or service providers are received, displayed or organized, wherein the master server collects analytics based which tool features are enabled to target specific products, services or information to the user to achieve the claimed invention in combination with the independent claims as a whole. None of the prior art of record disclose, teach, suggest and/or render obvious the above features to achieve the claimed invention in combination with the independent claims as a whole.

Id., 191-92.

84. Issued claim 1 (then pending claim 34) of the '068 Patent is nearly identical to allowed claim 1 of the '658 Patent. § V.C.2. Claim 1 is subject to a terminal disclaimer to the '658 patent. '068 *Prosecution History*, 181-184.

F. Priority Date

85. I have been informed that a U.S. patent application may claim the benefit of the filing date of an earlier application if the earlier patent application

disclosed each limitation of the invention claimed in the later-filed U.S. patent application.

86. As explained above, the Challenged Patents claim priority to two provisional applications filed on October 12, 2010, and February 22, 2011. I have been asked to assume that the Challenged Patents can claim priority to October 12, 2010.

87. If Patent Owner introduces evidence in this proceeding of an earlier invention date, I reserve the right to offer additional opinions and testimony to respond to such evidence.

VI. OVERVIEW OF THE TECHNOLOGY

A. Cloud Computing

88. Cloud applications, often referred to as cloud apps, are software programs that run on remote servers rather than on users' local devices. These applications are typically accessed over a network, meaning that users can interact with them via a webpage or other dedicated client interfaces (e.g., a mobile app) without needing to install all necessary infrastructure—such as networks, servers, and operating systems utilized to fulfill the application's functionalities—on their own devices (“on-premise”). Cloud applications are considered cloud-based “Software as a Service” (SaaS) that are directly provided to general application users.

89. Cloud applications rely on cloud computing infrastructures to perform tasks, store data, and manage services. Cloud computing infrastructures provide access to computing resources over a network, e.g., the internet. In 2006, Amazon Web Services (AWS) began offering IT infrastructure services to businesses in a cloud computing environment. As an example, Amazon announced Elastic Compute Cloud (EC2) in 2006, that allowed users to rent virtual servers in the cloud. EX1016. AWS utilized virtualization technologies that assigns virtual CPUs and storage to provide scalable and flexible computing resources to these EC2 instances. *Id.* Similarly, Microsoft announced Windows Azure in 2008 (EX1017); Google announced Google App Engine (GAE) in 2008, which later expanded to the Google Cloud Platform (GCP) that we know today. EX1018. AWS, Azure, and GAE are all cloud-based Platform as a Service (PaaS) and cloud-based Infrastructure as a Service (IaaS) systems that enables application developers to build, run, and operate cloud applications (cloud-based SaaS'es).

90. In July 2010, National Institute of Standards and Technology (NIST) had already established definitions and recommendations for cloud computing technologies and cloud applications. NIST's Director of Information Technology Laboratory, provided a testimony to the U.S. House of Representatives Committee on Oversight and Government Reform on the topic of Cloud Computing: Benefits and Risks of Moving Federal IT into the Cloud, on July 9, 2010. This testimony

recognizes “the development and deployment of cloud computing technology” in and before 2010, and summarizes essential characteristic, service models, and deployment models of cloud computing. According to NIST, cloud computing enabled shared cloud applications in “a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.” EX1019. This testimony further discusses Cloud SaaS, Cloud PaaS, and Cloud IaaS technologies available in 2010. *Id.*

1. Moving from Local Apps to Cloud Apps

91. As I discussed above, a cloud application offloads much of the computing, storage, and processing work to the cloud, i.e., remote servers, rather than running these on users’ local devices. Cloud applications improve upon local applications, which usually includes all application components—such as all networks, servers, and operating systems—installed on-premise on users’ devices. Compared to local applications, cloud applications offer a range of benefits, including enhanced accessibility, automatic updates, reduced infrastructure costs, scalability, and seamless collaboration across devices and locations.

92. Cloud applications can be accessed from any device with an internet connection, allowing users to work remotely or switch between devices easily. For

example, a user can log into a cloud application from their desktop, smartphone, or a laptop while they are traveling, without having to install a complex infrastructure on each device to execute certain application functionalities. This provides enhanced flexibility for accessing application functionalities.

93. Cloud applications automatically update “on the backend” in the cloud computing infrastructure, so users always have access to the latest features and security patches regardless of which user device is being used. This eliminates the need for manual installations of individual users on-premise, which is an effective way to ensure that the application is always up-to-date and secure.

94. Cloud applications also reduce infrastructure cost and maintenance for users of cloud services. With cloud applications, computing infrastructures are shared on the network and are sometimes managed by PaaS and IaaS entities in a subscription. Therefore, there’s no need for individual users (or entities utilizing cloud applications) to invest in expensive hardware, or to maintain servers and storage devices on-premise.

95. Cloud applications can also quickly scale up or down based on demand, allowing application developers to adjust resources without a complicated process to purchase physical hardware, add new hardware on-premise, then perform manual installation and maintenance. This flexibility also ensures that application

developers are quick to adapt to user needs without having to over-invest in physical infrastructure that may become underutilized.

96. These benefits of cloud applications are generally recognized around 2010. In July 2009, IBM published white paper “The Benefits of Cloud Computing” that discussed how IBM benefited from implementing cloud computing across the organization and within its products. According to IBM, “Thanks to cloud computing, IBM is able to respond rapidly to customer needs in a far more cost-effective manner. For example, cloud computing has enabled IBM to offer customers an extremely flexible, scalable solution for those peak periods when processing demands far exceed what customers’ internal IT resources can accommodate ... Similarly, ... [w]ith cloud computing, the provisioning of the Benchmark Center’s resources for customers has been condensed into a single day instead of a week or more. This has dramatically increased the Center’s capacity and greatly shortened the sales cycle, directly contributing to IBM’s bottom line. IBM’s Watson and Zurich Research Centers embraced cloud computing to schedule and complete far more projects than they otherwise would have been able to undertake. Instead of the usual two weeks required to request, schedule and provision a particular software stack for a research project, cloud computing allowed them to reserve IBM resources in numerous remote locations in a fraction of the time. This

has driven better resource utilization, consolidation of capital budgets, and greater productivity.” EX1020.

97. In 2010, Microsoft also published research in collaboration with Accenture (technology consultancy) and WSP Environment & Energy (environmental and sustainability consultancy) that further recognized the benefits of cloud applications. According to their research “Cloud Computing and Sustainability: The Environmental Benefits of Moving to the Cloud,” cloud computing enables increased efficiency and flexibility in resource provisioning, accessibility, and better resource utilization in servers and data centers. As a result, energy use and carbon emissions of cloud applications are greatly reduced compared to corresponding applications installed on-premise. EX1021.

B. RSS Feed Readers

98. RSS is a protocol that provides a standardized method for providing content (EX1022), usually for a website. Content can be in the form of metadata, which is data that is used to describe other data. RSS is usually used as a means for distributing news, events, blog updates, etc., to a broad audience across the web, including web users and applications. EX1023. Metadata in RSS includes information such as tags and related document structure that allows for consistent and straightforward parsing of the RSS data.

99. RSS is generally used as an acronym for “really simple syndication” but is sometimes also understood to mean “RDF site summary” or “rich site summary.” (EX1048, excerpts from Practical RDF).

When Netscape originally created the first implementation of an RSS specification, RSS stood for RDF Site Summary, and the plan was to use RDF/XML. When the company released, instead, a non-RDF XML version of the specification, RSS stood for Rich Site Summary. Recently, there has been increased activity with RSS, and two paths are emerging: one considers RSS to stand for Really Simple Syndication, a simple XML solution (promoted as RSS 2.0 by Dave Winer at Userland), and one returns RSS to its original roots of RDF Site Summary (RSS 1.0 by the RSS 1.0 Development group).

Id. at 10.

100. An “RSS feed” refers to a file in a standardized, computer-readable format that is designed to be utilized and interpreted by an application. This file contains up-to-date web content that is designed to be distributed.

101. For a website that offers RSS services, it is usually possible for a web user to navigate to the website and manually acquire an RSS feed by downloading the RSS feed file. However, a more common way is to use an “RSS aggregator,” also called an “RSS reader.” The RSS aggregator is usually responsible for acquiring RSS feeds that its user subscribes to, then centrally monitoring and displaying

information included in new RSS feeds in an easily readable format. The user can then use the RSS aggregator to review information and updates from multiple web resources that he or she subscribes to. EX1024.

102. RSS aggregators have a long and storied history. The first RSS aggregators were desktop applications such as NetNewsWire (EX1025), FeedDemon (EX1026), Newsgator (EX1027), and others. Soon after, web-based aggregators became common. The largest and most well-known of these aggregators was Google reader (EX1028), released in 2005 and discontinued in 2013.

103. The RSS Specification was initially released in March 1999; this RSS version was later referred to as “RSS 0.9.” In RSS 0.9, the expanded name of the term RSS was “RDF Site Summary.” RDF refers to “Resource Description Framework,” a World Wide Web Consortium (W3C) standard for offering descriptions in a directed graph. The RSS 0.9 Specification already details that a website can generate an RSS file of XML file format that includes content that it wants to distribute. According to the RSS 0.9 Specification, an RSS feed contains the following elements for displaying metadata, including components for image links, titles, description texts, etc. (EX1029).

Table 1. RSS 0.9 tags and syntax

Tag Description	Required?	Max. Length	Example Usage
XML version identifier	Always, exact	--	<?xml version="1.0"?>
RDF version identifier	Always, exact	--	<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns="http://channel.netscape.com/rdf/simple/0.9/">
Start channel data	Always	--	<channel>
Channel title	Always	40 char.	<title>Mozilla Dot Org</title>
Channel description	Always	500 char.	<description>the Mozilla Organization web site</description>
Channel title links to (URL)	Always	500 char.	<link>http://www.mozilla.org</link>
End channel data	Always	--	</channel>
Start image data	Optional	--	<image>
Image ALT text	If <image>	40	<title>Mozilla Dot Org</title>
Image location (URL)	If <image>	500	<url>http://www.mozilla.org/images/moz.gif</url>
Image links to (URL)	If <image>	500	<link>http://www.mozilla.org</link>
End image data	If <image>	--	</image>
Start item data	Always (1-15 items per channel)	--	<item>
Item title	Always	100	<title>Mozilla Party, 2.0!</title>
Item links to (URL)	Always	500	<link>http://www.mozilla.org/party/1999/</link>
End item data	Always	--	</item>
Start text input data *	Optional	--	<textinput>
Text input title	If <textinput>	40	<title>Send Comments</title>
Text input description	If <textinput>	100	<description>Use the text box below to send your comments to mozilla.org</description>
Text input name	If <textinput>	500	<name>ResponseText</name>
Text input links to (URL)	If <textinput>	500	<link>http://www.mozilla.org/cgi-bin/feedback.cgi</link>
End text input data	If <textinput>	--	</textinput>
End RDF	Always, exact	--	</rdf:RDF>

EX1029.

104. The RSS 1.0 Specification was developed by the RSS-DEV Working Group and released in December 2000. RSS 1.0 kept the core design of RSS 0.9 (including later minor modifications RSS 0.91 and 0.92), while adding support for XML Namespaces for modularization. RSS 1.0 Specification “compartmentaliz[es]” specific functionality into pluggable RSS modules, which means that RSS users can select different RSS modules, and use different RSS functionalities in selected

modules to add different types of metadata in an RSS feed. EX1022. RSS users are no longer restricted to the limited set of metadata components defined in RSS 0.9.

105. The RSS 1.0 Specification provides three standard RSS 1.0 modules as described at <https://web.resource.org/rss/1.0/modules/>. EX1030. These three modules are the Dublin Core module, the Syndication module, and the Content module. The Dublin Core module intends to include metadata elements from the Dublin Core Metadata Element Set. EX1031,. The Syndication module “[p]rovides syndication hints to aggregators and others picking up this RDF Site Summary (RSS) feed regarding how often it is updated.” EX1032. The Content module is the module that contains the actual contents of websites. EX1033.

106. The RSS 2.0 Specification was first released in September 2002; the current version of RSS used is RSS 2.0.11, published by the RSS Advisory Board in March 2009. EX1034. The RSS 2.0 Specification changed the “RSS” acronym to be “Really Simple Syndication.” RSS 2.0 includes a different list of both required elements and optional elements in an RSS feed, as specified below:

Required channel elements ↴

Here's a list of the required channel elements, each with a brief description, an example, and where available, a pointer to a more complete description.

Element	Description	Example
title	The name of the channel. It's how people refer to your service. If you have an HTML website that contains the same information as your RSS file, the title of your channel should be the same as the title of your website.	GoUpstate.com News Headlines
link	The URL to the HTML website corresponding to the channel.	http://www.goupstate.com/
description	Phrase or sentence describing the channel.	The latest news from GoUpstate.com, a Spartanburg Herald-Journal Web site.

EX1034.

Optional channel elements 📌

Here's a list of optional channel elements.

Element	Description	Example
language	The language the channel is written in. This allows aggregators to group all Italian language sites, for example, on a single page. A list of allowable values for this element, as provided by Netscape, is here . You may also use values defined by the W3C.	en-us
copyright	Copyright notice for content in the channel.	Copyright 2002, Spartanburg Herald-Journal
managingEditor	Email address for person responsible for editorial content.	geo@herald.com (George Matesky)
webMaster	Email address for person responsible for technical issues relating to channel.	betty@herald.com (Betty Guernsey)
pubDate	The publication date for the content in the channel. For example, the <i>New York Times</i> publishes on a daily basis, the publication date flips once every 24 hours. That's when the pubDate of the channel changes. All date-times in RSS conform to the Date and Time Specification of RFC 822 , with the exception that the year may be expressed with two characters or four characters (four preferred).	Sat, 07 Sep 2002 00:00:01 GMT
lastBuildDate	The last time the content of the channel changed.	Sat, 07 Sep 2002 09:42:31 GMT
category	Specify one or more categories that the channel belongs to. Follows the same rules as the <item>-level category element. More info .	<category>Newspapers</category>
generator	A string indicating the program used to generate the channel.	MightyInHouse Content System v2.3
docs	A URL that points to the documentation for the format used in the RSS file. It's probably a pointer to this page. It's for people who might stumble across an RSS file on a Web server 25 years from now and wonder what it is.	https://www.rssboard.org/rss-specification
cloud	Allows processes to register with a cloud to be notified of updates to the channel, implementing a lightweight publish-subscribe protocol for RSS feeds. More info here .	<cloud domain="rpc.sys.com" port="80" path="/RPC2" registerProcedure="pingMe" protocol="soap"/>
ttl	ttl stands for time to live. It's a number of minutes that indicates how long a channel can be cached before refreshing from the source. More info here .	<ttl>60</ttl>
image	Specifies a GIF, JPEG or PNG image that can be displayed with the channel. More info here .	
rating	The PICS rating for the channel.	
textInput	Specifies a text input box that can be displayed with the channel. More info here .	
skipHours	A hint for aggregators telling them which hours they can skip. This element contains up to 24 <hour> sub-elements whose value is a number between 0 and 23, representing a time in GMT, when aggregators, if they support the feature, may not read the channel on hours listed in the <skipHours> element. The hour beginning at midnight is hour zero.	
skipDays	A hint for aggregators telling them which days they can skip. This element contains up to seven <day> sub-elements whose value is Monday, Tuesday, Wednesday, Thursday, Friday, Saturday or Sunday. Aggregators may not read the channel during days listed in the <skipDays> element.	

EX1034.

107. A newer web syndication format called Atom was developed as an alternative to RSS in October of 2005. EX1035; EX1036. Atom is also an XML-

based web syndication format. Atom was initially designed to address various deficiencies in RSS. For example, Daniel Miessler describes the following (EX1037):

- “RSS 2.0 requires feed-level title, link, and description. RSS 2.0 does not require that any of the fields of individual items in a feed be present. Atom 1.0 requires that both feeds and entries include a title (which may be empty), a unique identifier, and a last-updated timestamp.”
- “RSS 2.0 may contain either plain text or escaped HTML, with no way to indicate which of the two is provided, and the RSS 2.0 content model does not permit actual well-formed XML markup, which reduces the re-usability of content. Atom has a carefully-designed payload container, where content must be explicitly labeled.”
- “RSS autodiscovery has been implemented several times in different ways, often relies on unregistered (thus invalid) application/rss+xml MIME type, and has never been standardized. Atom [WWW]standardizes autodiscovery. The application/atom+xml MIME Type is registered with IANA. Additionally, Atom feeds contain a ‘self’ pointer, so a newsreader can auto-subscribe given only the contents of the feed, based on Web-standard dispatching techniques.”

- “RSS 2.0 does not specify the handling of relative URI references. Different feed readers implement differing heuristics for their interpretation. There is no interoperability. In practice, relative URI references cannot be used in RSS feeds. Atom 1.0 specifies use of the XML’s built-in `xml:base` attribute for allowing the use of relative references.”
- “RSS 2.0 can be encrypted or signed like any other web content, if treated as arbitrary content. Rules for applying standard XML Encryption and XML Digital Signature on entries are included in Atom 1.0.”
- “The RSS 2.0 specification includes no schema. Atom 1.0 includes a (non-normative) ISO-Standard RelaxNG schema, to support those who want to check the validity of data advertised as Atom 1.0.”

108. As demonstrated by the W3C, Atom feeds contain similar elements to RSS feeds. (EX1038):

Elements of <feed>

A Feed consists of some metadata, followed by any number of entries.

Required feed elements

Here's a list of the required feed elements, each with a brief description, and an example.

Element	Description
id	Identifies the feed using a universally unique and permanent URI . If you have a long-term, renewable lease on your Internet domain name, then you can feel free to use your website's address. <id>http://example.com/</id>
title	Contains a human readable title for the feed. Often the same as the title of the associated website. This value should not be blank. <title>Example, Inc.</title>
updated	Indicates the last time the feed was modified in a significant way. <updated>2003-12-13T18:30:02Z</updated>

Recommended feed elements

Atom makes a number of additional requirements and recommendations for feed elements that you should be aware of. They are as follows:

Element	Description
author	Names one author of the feed. A feed may have multiple <code>author</code> elements. A feed must contain at least one <code>author</code> element unless all of the <code>entry</code> elements contain at least one <code>author</code> element. More info here . <author> <name>John Doe</name> <email>JohnDoe@example.com</email> <uri>http://example.com/~johndoe</uri> </author>
link	Identifies a related Web page. The type of relation is defined by the <code>rel</code> attribute. A feed is limited to one <code>alternate</code> per <code>type</code> and <code>hrefLang</code> . A feed should contain a link back to the feed itself. More info here . <link rel="self" href="/feed" />

EX1038.

Optional feed elements

Here's a list of optional feed elements.

Element	Description
category	Specifies a category that the feed belongs to. A <code>feed</code> may have multiple <code>category</code> elements. More info here . <code><category term="sports"/></code>
contributor	Names one contributor to the feed. An feed may have multiple contributor elements. More info here . <code><contributor></code> <code> <name>Jane Doe</name></code> <code></contributor></code>
generator	Identifies the software used to generate the feed, for debugging and other purposes. Both the <code>uri</code> and <code>version</code> attributes are optional. <code><generator uri="/myblog.php" version="1.0"></code> <code> Example Toolkit</code> <code></generator></code>
icon	Identifies a small image which provides iconic visual identification for the feed. Icons should be square. <code><icon>/icon.jpg</icon></code>
logo	Identifies a larger image which provides visual identification for the feed. Images should be twice as wide as they are tall. <code><logo>/logo.jpg</logo></code>
rights	Conveys information about rights, e.g. copyrights, held in and over the feed. More info here . <code><rights> © 2005 John Doe </rights></code>
subtitle	Contains a human-readable description or subtitle for the feed. More info here . <code><subtitle>all your examples are belong to us</subtitle></code>

EX1038.

Required Elements of <entry>

Here's a list of the required feed elements, each with a brief description, and an example.

Element	Description
id	Identifies the entry using a universally unique and permanent [URI]. Suggestions on how to make a good id can be found here . Two entries in a feed can have the same value for id if they represent the same entry at different points in time. <id> http://example.com/blog/1234 </id>
title	Contains a human readable title for the entry. This value should not be blank. <title>Atom-Powered Robots Run Amok</title>
updated	Indicates the last time the entry was modified in a significant way. This value need not change after a typo is fixed, only after a substantial modification. Generally, different entries in a feed will have different updated timestamps. <updated>2003-12-13T18:30:02-05:00</updated>

Recommended elements of <entry>

Atom makes a number of additional requirements and recommendations for entry elements that you should be aware of. They are as follows:

Element	Description
author	Names one author of the entry. An entry may have multiple authors. An entry must contain at least one <code>author</code> element unless there is an <code>author</code> element in the enclosing <code>feed</code> , or there is an <code>author</code> element in the enclosed <code>source</code> element. More info here . <author> <name>John Doe</name> </author>
content	Contains or links to the complete content of the entry. Content must be provided if there is no <code>alternate</code> link, and should be provided if there is no <code>summary</code> . More info here . <content>complete story here</content>
link	Identifies a related Web page. The type of relation is defined by the <code>rel</code> attribute. An entry is limited to one <code>alternate</code> per <code>type</code> and <code>hreflang</code> . An entry must contain an <code>alternate</code> link if there is no <code>content</code> element. More info here . <link rel="alternate" href="/blog/1234"/>
summary	Conveys a short summary, abstract, or excerpt of the entry. Summary should be provided if there either is no content provided for the entry, or that content is not inline (i.e., contains a <code>src</code> attribute), or if the content is encoded in base64. More info here . <summary>Some text.</summary>

Optional elements of <entry>

Here's a list of optional feed elements.

Element	Description
category	Specifies a category that the entry belongs to. A <code>entry</code> may have multiple <code>category</code> elements. More info here . <category term="technology"/>
contributor	Names one contributor to the entry. An entry may have multiple <code>contributor</code> elements. More info here . <contributor> <name>Jane Doe</name> </contributor>
published	Contains the time of the initial creation or first availability of the entry. <published>2003-12-13T09:17:51-08:00</published>
rights	Conveys information about rights, e.g. copyrights, held in and over the entry. More info here . <rights type="html"> &copy; 2005 John Doe </rights>
source	Contains metadata from the source feed if this entry is a copy. <source> <id> http://example.org/ </id> <title>Example, Inc.</title> <updated>2003-12-13T18:30:02</updated> </source>

EX1038.

109. Atom, like RSS, has reached widespread adoption. Google recommends that site operators offer RSS/Atom feeds in order to optimize site crawling. EX1039. Indeed, numerous applications and services exist to generate, ingest, or parse RSS and Atom feeds. EX1040. These include desktop, mobile, self-hosted and remotely-hosted feed readers; feed parsing libraries and feed generators. Common web browsers, such as Mozilla Firefox, Google Chrome and Apple Safari, support subscribing to RSS and Atom feeds. EX1041; EX1042.

1. RSS Readers Moved To The Cloud

110. By 2010, several RSS readers were released as cloud applications. These cloud-based RSS readers allowed users to access their RSS feeds from any device with internet access, as I discussed above in the benefits of cloud applications.

111. Google Reader was a cloud-based RSS Reader launched in October 2005. EX1043. According to a user tutorial shared by Google, Google Reader allowed users to subscribe to RSS feeds, view content updates, and share articles with others in a web interface (formerly accessed through <http://reader.google.com/>). EX1044. According to Google (EX1043), Google Reader utilizes Google's BigTable infrastructure to retrieve, store, and present large amounts of data in a semi-structured manner to Google Reader users. EX1045 (now accessible at: <https://andrewhitchcock.org/2005/bigtable.html>). With Google Reader being a cloud application, an individual Google Reader user does not need to invest in powerful hardware on-premise, or run their own server and database for RSS content retrieval and storage, while still accessing RSS feeds that they are subscribed to by simply navigating to Google Reader in a browser.

112. Feedly was another cloud-based RSS reader around 2010. It enabled users to subscribe and view to RSS feeds, share articles, as well as other functionalities like posting articles to social media. Similar to Google Reader, Feedly also provides a web interface for users, accessed through <http://www.feedly.com/>.

(see tutorial <https://www.youtube.com/watch?v=GqTr-Qiiwwk>). Therefore, a user does not need to configure or run in-house sever or database, while still accessing RSS feeds in a browser.

113. The benefits of cloud applications, as I discussed above, such as enhanced accessibility, automatic application updates, and scalability, apply to cloud-based RSS readers.

114. This shift to the cloud made RSS readers more accessible, particularly to users who wanted to keep up with content across different devices without manually syncing or maintaining an offline version of the software. Users simply navigate to a web interface on any of their devices to access their RSS feeds.

115. Users could seamlessly sync their feed subscriptions, read/unread status, and preferences across multiple devices, ensuring a consistent experience no matter where they accessed their reader.

116. Since the RSS reader is hosted on the cloud, application updates are pushed directly to users without requiring manual downloads or installations. Users automatically access the most up-to-date RSS reader features whenever they navigate to the Google Reader or Feedly website, ensuring that the latest features, bug fixes, and security patches are always in place.

117. In addition, cloud-based RSS readers are highly scalable for their developers. The cloud infrastructure—such as Google’s BigTable technology—can be adjusted to accommodate the growth in user numbers and user needs.

118. Cloud-based RSS readers also enable collaboration and sharing by allowing users to easily share articles, feeds, or curated content with others. Whether through social media integration, shared folders, or collaborative reading lists, these features make it simple for teams or communities to stay informed and exchange insights in real time.

VII. LEVEL OF ORDINARY SKILL IN THE ART

119. I am not an attorney and offer no legal opinions. I have been informed about certain aspects of the law for purposes of my analyses and opinions.³ Counsel has informed me that I should consider these materials through the lens of one of ordinary skill in the art related to the ’658 patent at the time of the invention, and I have done so during my review of these materials.

³ I understand that the patent laws were amended by the America Invents Act (AIA), but that the earlier statutory requirements still apply to pre-AIA patents. I have been further informed that ’658 patent is a pre-AIA patent, so the pre-AIA requirements control.

120. I understand that the claims and written description of a patent must be understood from the perspective of a person of ordinary skill in the art (“POSITA”). I have been informed that the following factors may affect the level of skill of one of ordinary skill in the art: (1) the educational level of the inventor; (2) the type of problems encountered in the art; (3) the prior-art solutions to those problems; (4) the rapidity with which innovations are made; (5) the sophistication of the technology; and (6) the educational level of active workers in the field. A person of ordinary skill in the art is also a person of ordinary creativity in the art.

121. As explained above, I have been asked to assume the Challenged Patents can claim a priority of October 12, 2010. My analysis reflects the views of a person of ordinary skill in the art in the field of the Challenged Patents as of October 12, 2010. This would have been someone knowledgeable and familiar with the field of cloud-computing, and in particular, content aggregation by a cloud-base system and would further understand presenting aggregated content to a remote computer using web pages that include the capability to scroll through the content. That person would have had a bachelor’s degree in computer science, or a related field, and at least two years of experience in the research, design, or development of cloud computing and networking systems, or the equivalent or a master’s degree in computer science or similar field, with additional education substituting for experience and vice versa.

122. For purposes of this Declaration, in general, and unless otherwise noted, my statements and opinions, such as those regarding my experience and the understanding of a POSITA generally (and specifically related to the references I consulted herein), reflect the knowledge that existed in the field as of the earliest assumed priority date of the Challenged Patents (i.e., October 12, 2010). Unless otherwise stated, when I provide my understanding and analysis below, it is consistent with the level of a POSITA prior to the earliest alleged priority date of the Challenged Patents.

VIII. CLAIM CONSTRUCTION

123. It is my understanding that in order to properly evaluate the Challenged Patents, the terms of the claims must first be interpreted. It is my understanding that for the purposes of this inter partes review, the claims are to be construed under the so-called *Phillips* standard, under which claim terms are given their ordinary and customary meaning as would have been understood by one of ordinary skill in the art in light of the specification and prosecution history, unless the inventor has set forth a special meaning for a term. For the purposes of this declaration, I have given all the claim terms of the Challenged Claims their plain and ordinary meaning, as would have been understood by a person of ordinary skill in the art, at the time of the alleged invention, which I have been asked to assume is October 12, 2010, for all claims.

124. The Challenged Patents state that “[t]o facilitate the clarity of the ensuing description, words and phrases listed below have been ascribed the following meanings.” ’658 patent, 3:65-67; ’068 patent, 4:10-12. I have reproduced those “words and phrases” that are definitively stated, below.

Claim	Term	Construction
1, 7, 9, 12, 17	“Content data”	“[A]ny data or information that is stored remotely on servers that are generally accessible over the internet/intranet via a service provider servers.” ’658 patent, 4:1-3; ’068 patent, 4:13-15.
1, 5, 7, 9, 12, 17	“Content applications”	“[S]oftware applications that accesses display and organized content data, generally from a web-sites.” ’658 patent, 4:4-6; ’068 patent, 4:16-18.
1, 5, 7, 8, 9, 12, 16, 18	“master server”	“[A] dedicated server for supporting a master program and user interface to access and view content data and applications from multiple content data sources or providers through a single user interface.” ’658 patent, 4:7-10; ’068 patent, 4:19-22.
1, 7, 12	“tool-box”	“[A] set of sub-features, generally accessible through an administrator account or password, that allow a web-page layout and/or operation to be customized by linking tools with content applications.” ’658 patent, 4:16-19; ’068 patent, 4:28-31.
1, 6, 10, 12	“master web-page”	“[A] web-page viewable remotely from a remote computing device via log-in server or master server and which displays a slide tool that scrolls or pages through content data and/or content application from multiple content data sources and/or service providers in real time.” ’658 patent, 4:20-25; ’068 patent, 4:32-37.
1	“remote comput[er] device”	“[A]n electronic device that is capable of accessing the log-in server or master server over the internet/intranet and displaying or viewing a master web-page and content data therefrom.” ’658 patent, 4:30-33; ’068 patent, 4:42-45.

IX. PRIOR ART

A. Summary of Darnell (Ex. 1004)

125. Darnell is in the same field as the Challenged Patents and directed to solving similar problems, such as presenting multiple content items to a user more efficiently by aggregating content items in a single user interface. '658 patent, 1:51-62; '068 patent, 1:59-2:3; Darnell, ¶¶9-11. Darnell is a prior art reference with direct relevance to the Challenged Patents, offering similar cloud-based content aggregation functionalities.

126. Darnell discloses a “server system for serving extended content feeds to clients.” Darnell, Title. Darnell’s architecture, illustrated in Figure 1 below, includes a “server system 104” connected to “one or more client systems 102 and to one or more hosts 120 (or ‘feed sources’) by network 106.” Darnell, ¶32. This server is responsible for “retriev[ing]...a currently published set of content items for a respective content feed, and merg[ing] the retrieved content items into a previously stored set of previously published content items for the respective content feed.” *Id.*, Abstract, ¶¶11-14. This merging process enables Darnell’s system to aggregate content feeds and stream them into a unified user interface on the client device, ensuring users receive both current and historical content within one application interface. *Id.*, ¶¶11-14, 41.

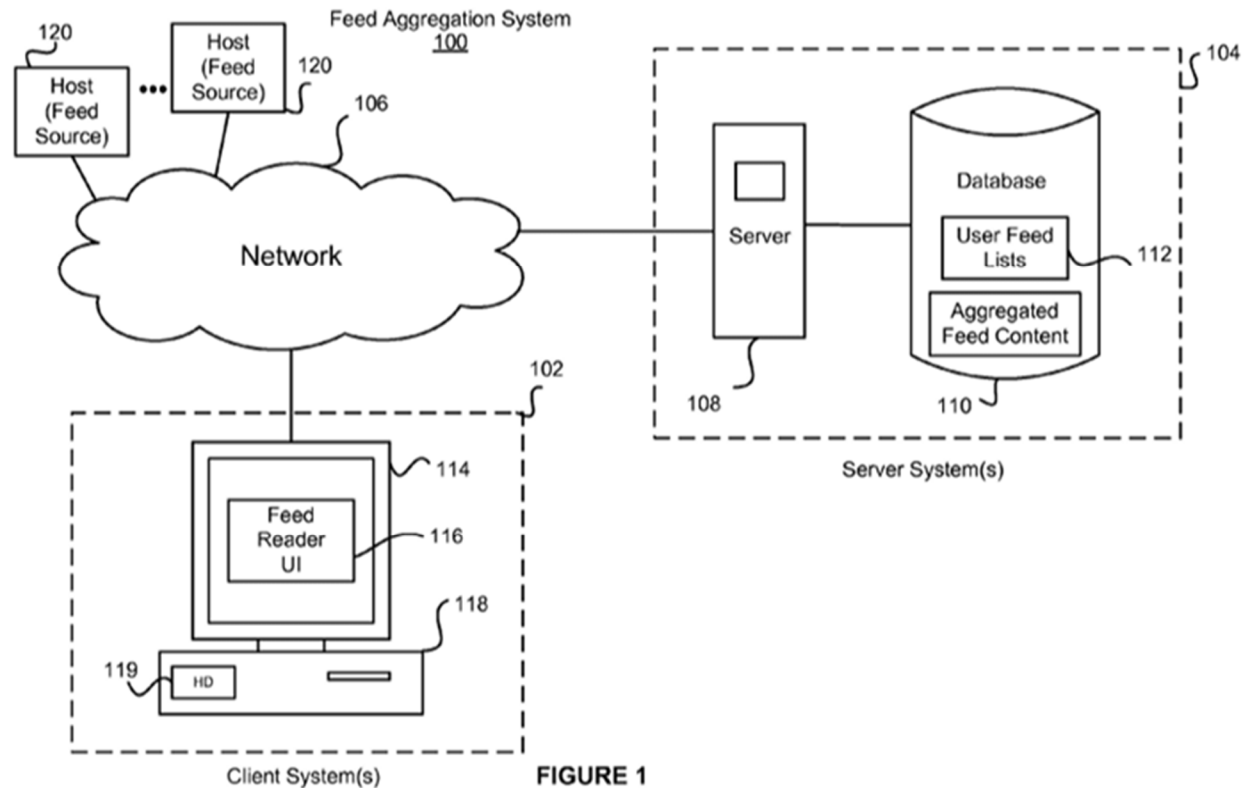


FIGURE 1

Id., Fig. 1.

127. Darnell's system supports various content syndication formats, including RSS, which uses XML for content item specification, and extends compatibility to "the Atom syndication format or the VCALENDAR calendar format." *Id.*, ¶¶37-38. This flexibility enables Darnell to accommodate diverse content sources, allowing users to access a variety of content feeds.

128. The content feed aggregation system in Darnell allows users to "subscribe to one or more content feeds and be presented with the content items included in those content feeds." *Id.*, ¶¶30-32. Through the feed reader user interface 116, users can build a subscription list, define filters, and apply labels to their content

feeds, all within a unified interface. *Id.*, ¶40, Fig. 1. Additionally, Darnell’s system stores “user feed information”, including “content feed subscriptions, as well as sources, filters, and read states for that particular user.” *Id.*, ¶34. Displayed content items are organized chronologically by timestamp and sorted into topical folders, such as “News” and “Sports,” enhancing content accessibility and user experience, as depicted in Figure 3A, below. *Id.*, ¶¶43-44, 50-54.

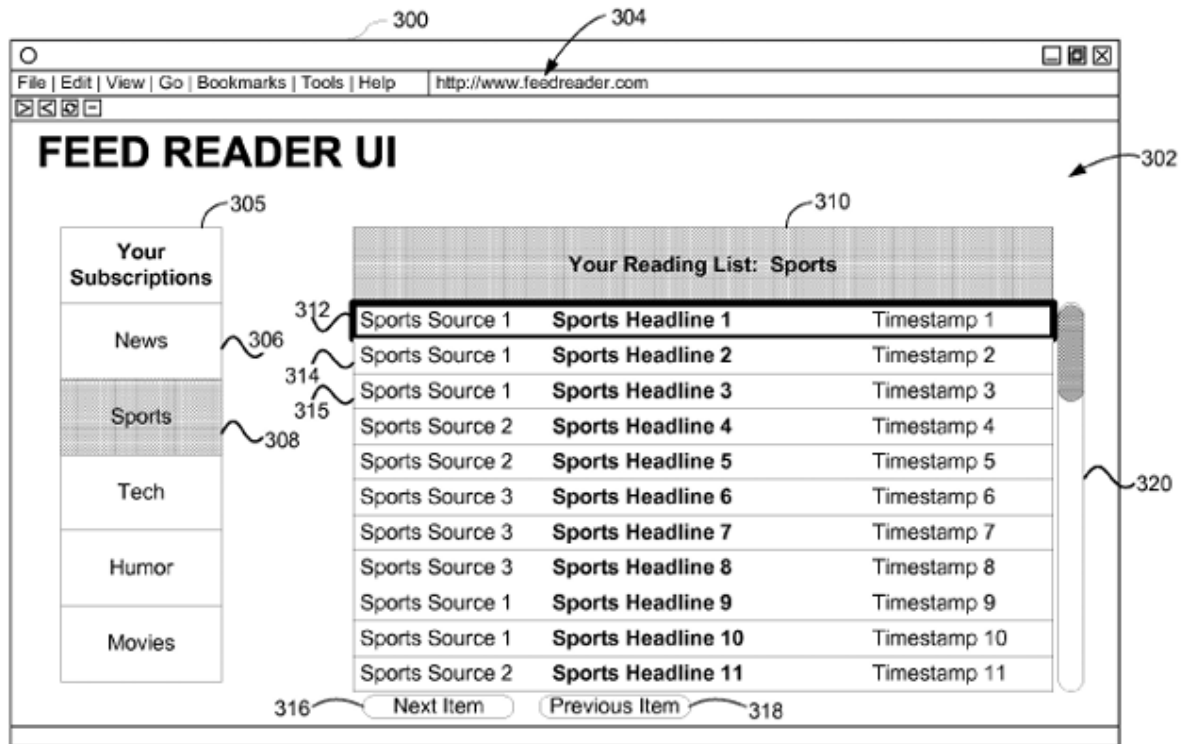


Figure 3A

Id., Fig. 3A.

129. Furthermore, Darnell’s system allows users to customize their viewing experience by setting a sort order flag to control the display sequence of content

items and adjusting UI flags that “specify the formatting and color of a user interface.” *Id.*, ¶85. In terms of security, Darnell employs a login mechanism that leverages a user ID and token, which the server uses “to authenticate requests from the user” and prevent unauthorized access or manipulation of the user’s bookmarks within the feed reader system. *Id.*, ¶84.

B. Summary of Kim (Ex. 1005)

130. Kim is in the same field as the Challenged Patents, disclosing a content feed aggregation system designed to present content in a unified interface, classifying and organizing content items based on identified traits. Kim, Abstract. I also understand that Kim was cited by the examiner during the prosecution of Darnell. EX1009, 20.

131. Kim's system classifies items by analyzing traits such as content type, source, and structural elements, and subsequently organizes them into distinct groups for improved presentation. Kim, Abstract, ¶2, Claim 1. This classification system accommodates a range of content types, including feeds in XML format—a markup language—and extends to RSS and email feeds. *Id.*, ¶24, ¶37. The system enables the integration of various content sources, such as news, blogs, and email organized into groups based on classifications and presented in a single display interface, allowing accessibility and usability of multiple content streams for the user. *Id.*, ¶¶24-27, 36-42.

132. A primary feature of Kim's system is its ability to organize content items into groups based on their traits. *Id.*, ¶18. For example, content items sharing a common trait, such as originating from the same feed source or containing media enclosures, are classified into the same group. *Id.*, ¶¶43-47. This grouped organization enables users to navigate content more intuitively within the user interface (UI), which provides tailored displays based on classifications like links, text, formatted text, enclosed media, and conversations. *Id.*, ¶¶18, 26-29, Figs. 3-5, 7-9, 11-13. Through this classification, the UI presents users with content in a format that best suits each type. For example, link-based content may appear as concise, headline-focused summaries, while blog entries with comments may display both the original content and associated comments side by side, as illustrated in Figure 4, below. *Id.*, ¶¶33, 38-39, Fig. 4. By enabling users to access associated content, such as blog comments, in a visually connected manner, Kim's interface enhances the comprehensibility and coherence of the presented information.

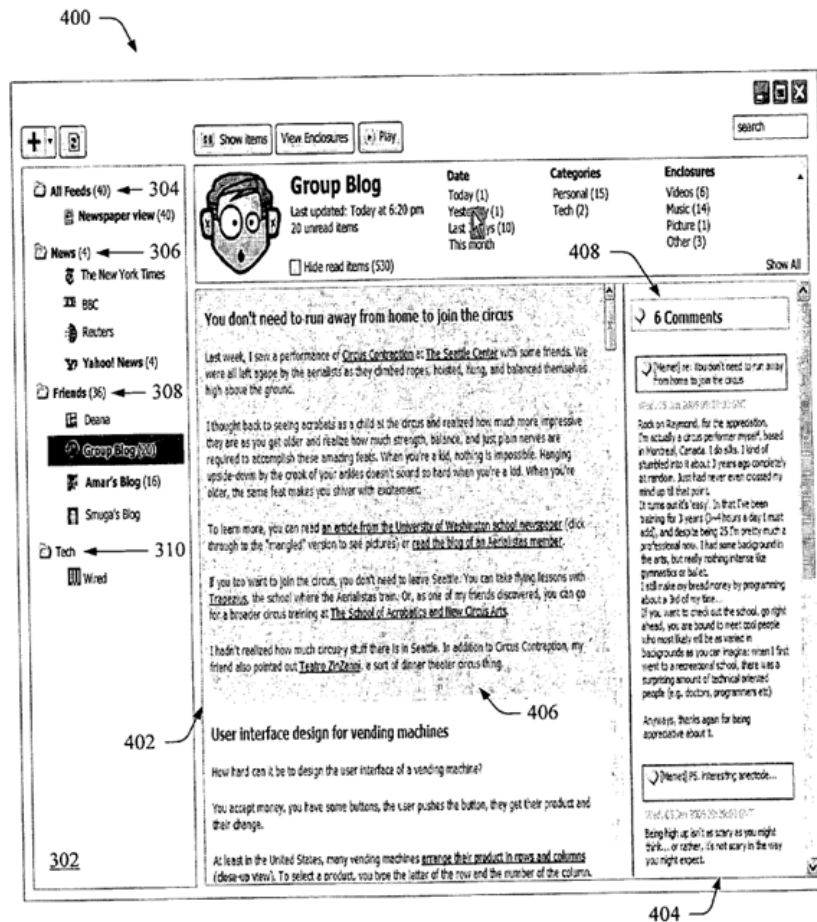


FIG. 4

Id., Fig. 4.

133. Moreover, Kim's system supports trait-based navigation and selection of content. *Id.*, ¶¶43-47, 58, Fig. 7. It allows users to view and select groups of content items sharing traits like unread status, file type, or other classifications. For instance, the system may group emails with attachments or items containing specific enclosures, such as audio or video files, based on these characteristics. *Id.*, ¶¶39-41, 45-50. Users can mark items as "read" or select to play enclosures within the

interface, making it possible to interact with grouped content efficiently. This design allows users to make the most of their content by emphasizing frequently accessed or categorized information and enhances the interaction with specific types of media directly within the feed application. *Id.*, ¶¶63-66.

134. Kim’s system is also adaptive, responding to user preferences and usage patterns by customizing the order and priority of displayed content. Within its interface, Kim includes a summary region that highlights significant content, such as headlines and brief descriptions organized by chronology and feed source. *Id.*, ¶¶68. Furthermore, the system utilizes usage history data to determine which feeds and content are “most popular” with the user, effectively curating a prioritized view of content based on previous interactions. *Id.*, ¶¶69-70. The prioritization mechanism ensures that frequently accessed feeds are prominently displayed, while less relevant feeds are deemphasized, thereby optimizing the user experience. This adaptable user interface aligns displayed content with the user’s historical preferences, allowing the interface to dynamically present content that is likely to be of greater interest, as depicted in the organization of the feeds within Figure 13. *Id.*, ¶¶68-70.

135. In addition to these adaptive features, Kim’s system displays items in a hierarchical structure within folders or groups, presenting users with an organized and logical layout of content. For example, the system may display news items

within a “News” folder or user-specific feeds in a “Friends” folder, thereby maintaining a topical structure that simplifies navigation across a broad array of content types. *Id.*, ¶¶30-32, Fig. 3. By enabling users to filter content by these topical groups or traits, Kim’s system allows access to relevant content within a diverse set of feeds.

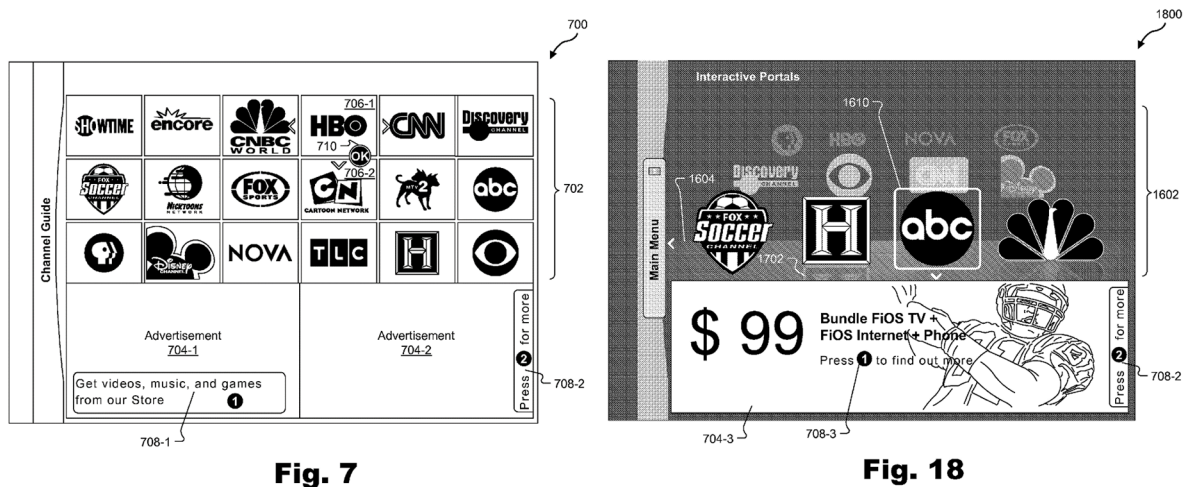
C. Summary of Dasgupta (Ex. 1006)

136. Dasgupta is in the same field as and relevant to the Challenged Patents, solving similar problems by implementing “GUIs that provide an intuitive and appropriate balance of information, advertisements, usability, aesthetics, and functionality.” Dasgupta, 1:15-30.

137. Dasgupta discloses a graphical user interface (GUI) designed that aggregates media content through a structured channel guide, providing both content accessibility and targeted advertising. Specifically, Dasgupta’s system displays a channel guide that organizes media content channels within a “grid matrix of channel icons representing a plurality of media content channels,” allowing users to intuitively browse and select channels for media content access. *Id.*, Abstract. This interface is further enhanced by a responsive advertisement display that adapts based on user navigation within the GUI. *Id.*, Abstract; 1:7-14, 1:54-56.

138. The GUI features a graphical representation of the channel guide displaying content in a “two-dimensional grid matrix view” or in a “a three-

dimensional grid matrix view” alongside advertisements, as illustrated in Figures 7 and 18, reproduced below. *Id.*,13:9-23, 17:43-63, 19:4-15. This layout allows for a unified display where channel icons and advertisements coexist, offering users an organized and informative overview of available content and relevant promotions. *Id.*, Abstract,1:64-2:16. Notably, the advertisement content within the GUI is dynamic; it updates based on user actions, such as when a user navigates through the guide or selects a specific channel. *Id.*,4:9-23. This adaptive functionality is seen as particularly useful for targeting content and advertisements to the user’s current focus, ensuring that advertisements are contextually relevant. *Id.*,14:45-15:44, 17:43-63, 19:4-15.



Id., Figs. 7 and 18.

139. Dasgupta's GUI also integrates a “selector” tool (shown as selector 710 in Figure 7), which enables users to scroll through different content items within the feed, such as selecting the HBO channel icon (706-1). *Id.*,14:45-15:44. The selector

allows users to highlight and navigate through channels in the channel guide's two- or three-dimensional, grid-based layouts. *Id.*, 14:45-67, 18:49-19:3. This grid matrix interface, paired with selector 710, allows a user to scroll through organized content to select the any content displayed. *Id.*, 14:45-15:49, 18:39-19:3.

140. The GUI's design also includes a mechanism for organizing advertisements. In "launch mode," which immediately follows launching of the channel guide GUI, "a first category of advertisement content may be displayed together with channel guide." *Id.*, 19:25-36. However, in "navigation mode," when the user is scrolling through the content, the advertisement content can be tailored to match the selected channel. *Id.*, 15:10-29. For instance, selecting HBO might replace a general ad with content specific to HBO's subscription services, creating a personalized experience and promoting user engagement. *Id.*, 15:30-44, 16:6-16, Fig. 9.

141. Dasgupta's media content access system includes "a media content access facility 102, a channel guide facility 104, an advertisement facility 106, a user interface facility 108, an interactive portal access facility 110, and a storage facility 112 selectively and communicatively coupled to one another." *Id.*, 3:4-13. The "[m]edia content access facility 102 may be configured to access, process, and present media content for experiencing by a user" and "may employ any

technologies suitable for accessing media content, including accessing media content distributed by a remote source.” *Id.*, 3:14-28.

X. SUMMARY OF OPINIONS

142. My findings, as explained below, are based on my study, experience, and background in the fields discussed above, informed by my education in applied mathematics and electrical engineering, and my experience in the design and analysis of cloud-based content aggregation systems. In my opinion, all Challenged Claims are invalid as either anticipated or obvious.

A. ’658 Patent

143. As explained below, my opinion is that claims 1-5 and 7-10 of the ’658 patent are obvious in view of the combination of Darnell and Kim. I also believe that claims 6 and 12-18 are obvious in view of the combination of Darnell, Kim, and Dasgupta. In my opinion, claims 1 and 4-10 are also obvious in view of the combination of Dasgupta and Darnell. I also opine that claims 2-3 and 11 are obvious in view of the combination of Dasgupta, Darnell, and Kim. Below is a summary of the Grounds for the ’658 Patent:

Ground (all under 35 U.S.C. §103)	Claims	References
1A	1-5, 7-10	Darnell, Kim
1B	6, 12-18	Darnell, Kim, Dasgupta
2A	1, 4-10	Dasgupta, Darnell
2B	2-3, 11	Dasgupta, Darnell, Kim

B. '068 Patent

144. Also explained below, my opinion is that claim 1 of the '068 patent is obvious in view of the combination of Darnell and Kim. In my opinion, claim 1 is also obvious in view of the combination of Dasgupta and Darnell. Below is a summary of the Grounds for the '068 Patent:

Ground (all under 35 U.S.C. §103)	Claim	References
1	1	Darnell, Kim
2	1	Dasgupta, Darnell

XI. '658 PATENT: DETAILED EXPLANATION OF THE GROUNDS

A. GROUND 1A: DARNELL AND KIM RENDER CLAIMS 1-5 AND 7-11 OBVIOUS.

145. In my opinion, as discussed below, the combination of Darnell and Kim renders obvious to a POSITA at the time of the alleged invention on October 12, 2010, at least claims 1-5 and 7-11.

1. Motivation to Combine Darnell with Kim

146. In my opinion, Kim's teachings of aggregating diverse content types, such as news, blogs, and emails, are suitable for use in Darnell's feed reader system, in which content sources are aggregated on a remote server rather than at a user device. Kim, Abstract, ¶¶2, 18; Darnell, Abstract, ¶¶30-32. Darnell and Kim address similar problems of aggregating and presenting multiple feed sources to a user. For example, Darnell addresses "[l]ooking through a dense list of content items to locate

items of interest,” which is cumbersome for a user. Darnell, ¶8. And Kim similarly addresses the problem of viewing multiple pieces of fragmented content from news sources, blog entries, email, and the like in a “consuming experience having significant drawbacks.” Kim, ¶1. Both Darnell and Kim utilize RSS or XML-based formats for feed content, which enables integration of feed items from different sources. Darnell, ¶¶36-37; Kim, ¶¶24, 37. For example, both Darnell and Kim may receive feed content using RSS or similar feeds, motivating a POSITA to combine complementary features in receiving feed content using RSS with a reasonable expectation of success. In my opinion, Kim’s feed application, email application, media application, and photo application could be implemented using Darnell’s server system 104 rather than on a user device, which would allow the user to access personalized content feeds on any user device (e.g., phone, computer, tablet, etc.) instead of only on a single user device. Moreover, Kim was cited by the Examiner during the prosecution of Darnell suggesting an underlying relevance between the two references, indicating that Kim’s disclosure was viewed as pertinent to Darnell’s system. EX1009, 20. Citing Kim in Darnell’s prosecution underscores that a POSITA would have reasonably been motivated to combine Kim’s teachings with Darnell.

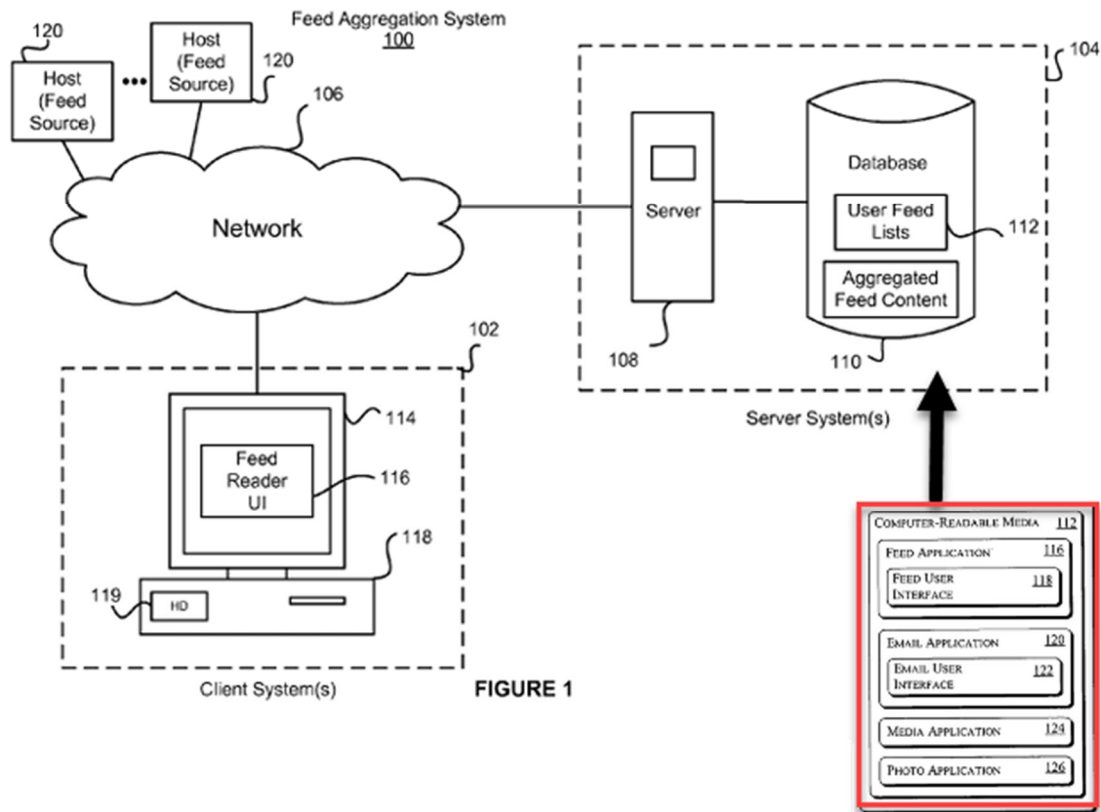
147. The inclusion of Kim’s classification and trait-based organization with Darnell’s server-oriented architecture would represent a straightforward extension

of Darnell's server system, where Kim's content analysis and tailored presentation functionalities would have been incorporated directly into Darnell's server system. This combination would bring the benefits of Kim's trait-organized interface to any user device accessing Darnell's server-based architecture, offering users a cohesive experience across multiple devices without dependence on a single local application. Darnell, ¶32, Fig. 1.

148. A POSITA would have been motivated to combine these teachings for several reasons. For example, both references are directed toward content aggregation systems that centralize content from multiple sources and present it to a user through a unified interface on a user device. Darnell, ¶¶8, 11-14, 30-32, Fig. 1; Kim, Abstract, ¶¶18-23, Fig. 1. Both references share the primary goal of simplifying content presentation from diverse feeds and specifically address the inefficiencies users face when attempting to locate specific items of interest within a large volume of content. For example, Darnell explicitly identifies the problem of users "looking through a dense list of content items to locate items of interest" as cumbersome (Darnell, ¶9), while Kim describes a "consuming experience having significant drawbacks" due to the fragmented presentation of content from news, blogs, and email (Kim, ¶1). By combining Kim's trait-based classification system with Darnell's server-based structure, a user would achieve a more efficient, organized, and user-friendly interface for presenting content.

149. Furthermore, a POSITA would have been motivated to incorporate Kim's aggregation of news, blogs, and email in a network computer and sending the feeds to a remote computer for display in a user interface, within Darnell's architecture to allow for centralized analysis and grouping based on content traits on the server before content is sent to a client. Kim, ¶¶37, 39. A POSITA would have seen value in implementing Kim's trait identification tools within Darnell's server system 104, because it would have allowed users to access personalized, trait-organized feeds from any device, removing the dependency on a single user device and enhancing the flexibility and portability of the system. Kim, ¶¶20-24, 33, 38-39. For example, Kim discloses that "[a] feed from Yahoo!™ News for instance, may predominantly comprise links to news articles." Kim, ¶24. Yahoo! News stores content on a remote server that is then served to a user device. As such, a POSITA would have recognized that Kim's content aggregation system was comparable with Darnell's server-based feed aggregation system. Such a combination would have been motivated by the increasing demand for cloud-based, device-agnostic services available at the time.

150. To illustrate the simplicity of combining Kim and Darnell, the following figure combines the architectures of Kim and Darnell, specifically moving Kim's computer readable media to Darnell's server system.



Darnell, Fig. 1.

151. Both Darnell and Kim disclose systems that aggregate content feeds and present them in user-friendly formats, each aiming to provide users with an organized and accessible interface. Combining Darnell and Kim would have been nothing more than using Kim’s known technique of detailed, trait-oriented organization in the same way as in Darnell’s remote server. Darnell, Abstract, ¶¶30-32; Kim, ¶¶20-24, Figs. 3-5. Specifically, a POSITA would have recognized the logical extension of Darnell’s aggregation capabilities by including Kim’s classification of items into “link,” “text,” “enclosed media,” and “conversation”

formats based on their content traits. Kim, ¶¶26. This combination would have provided users with a streamlined and tailored experience across content types, enhancing content accessibility and reducing time spent sifting through disorganized items.

152. Applying Kim's trait-based classification to would have been a predictable improvement to Darnell's server feed aggregation system. Both references disclose user-oriented content presentation, and in my opinion, integrating Kim's method of grouping by traits would achieve a more refined and customizable user experience without requiring significant modifications to Darnell's architecture. Using Kim's organization of feeds based on user history and relevance would have improved Darnell's similar system by tailoring content presentation to individual user preferences, making frequently accessed or popular feeds readily available. Kim, ¶¶68-70. The user-centric emphasis of both systems would have been retained and enhanced through such a combination, allowing the resulting interface to display feeds that are both aggregated remotely and organized for efficient user consumption.

153. Thus, in my opinion, a POSITA would have been motivated to combine Darnell and Kim not only due to their complementary goals and solutions but also because this combination would have involved merely applying known techniques (Kim's trait-based sorting) to an existing feed aggregation system using a remote

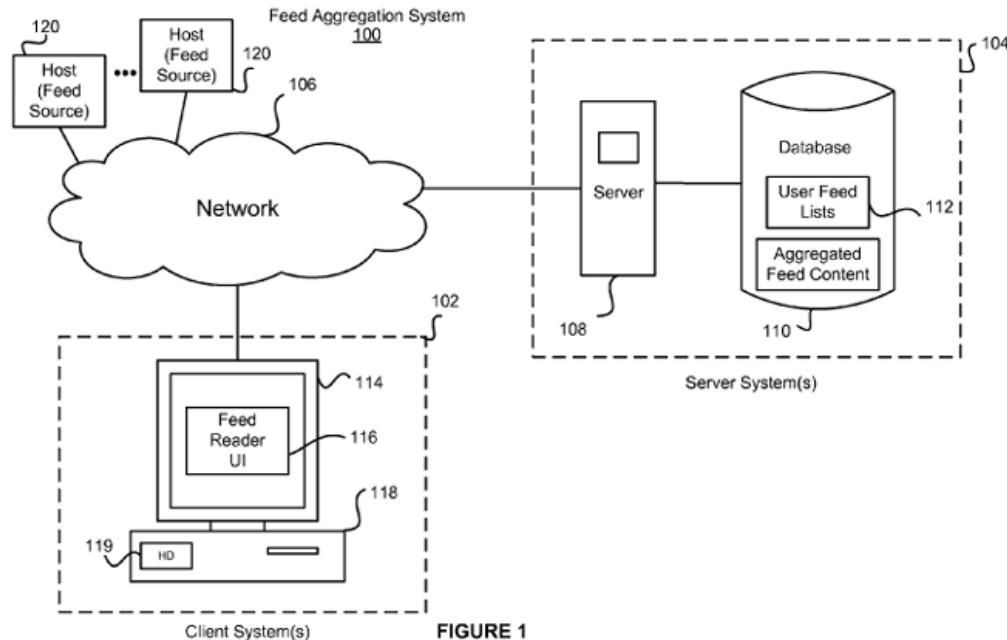
server architecture (Darnell). And using Kim’s organization of feeds based on user history and relevance would have improved Darnell’s similar system by tailoring content presentation to individual user preferences, making frequently accessed or popular feeds readily available. Kim, ¶¶68-70. This approach reflects routine skill in the art by implementing Kim’s trait identification and trait-based grouping as a means to enhance user accessibility within Darnell’s already centralized feed reader environment. By doing so, a POSITA would have successfully created a highly adaptive and cross-device content aggregation system, addressing both the user’s need for organized content and the industry’s shift toward centralized, server-based solutions.

2. Independent Claim 1

a. Preamble of claim 1, “A cloud-based computer system comprising:”

154. To the extent the preamble of Claim 1 is limiting, Darnell’s disclosure meets the definition of a “cloud-based computer system” as claimed in the ’658 patent. Darnell’s Figure 1 (below) illustrates a “content feed aggregation system” that includes “a server system 104” communicating with “one or more client systems 102” and “one or more hosts 120 (or ‘feed sources’)” by network 106. Darnell, ¶32. Both client and server systems are computers. *Id.*, ¶¶61-80. And Darnell further explains “network 106 may be any suitable wired and/or wireless network and may

include a local area network (LAN), wide area network (WAN), the Internet, metropolitan area network (MAN), or any combination of such networks.” *Id.*, ¶32



Id., Fig. 1.

155. This structure is illustrated in Darnell’s Figure 1, which depicts a server-client setup designed to aggregate and distribute content across a network, which a POSITA would have recognized as a cloud-based computer system. ’658 patent, 1:43-46, 4:27-30 (“Cloud-based computer system or architecture includes multiple content data sources or service provider servers that are linked to a log-in server or master server via the internet/intranet.”).

156. Darnell provides a comprehensive description of the server and client systems, making it clear that each is a computer with specific roles in the aggregation

and distribution of content to remote clients. For example, Darnell describes that client systems may be configured to receive, store, and organize feed content aggregated by the server. Darnell, ¶¶61-80. Darnell also explains that network 106 “may be any suitable wired and/or wireless network and may include a local area network (LAN), wide area network (WAN), the Internet, metropolitan area network (MAN), or any combination of such networks.” *Id.*, ¶32. This setup allows the server to aggregate content from various sources and deliver it to client devices across a geographically dispersed network, a configuration aligning with cloud-based architecture.

157. Darnell’s disclosure emphasizes that the server’s operations extend to managing and processing content for diverse feeds stored in a shared repository. Specifically, Darnell explains that the server system may include or have access to one or more databases configured to store the feeds, allowing dynamic retrieval and management of content stored centrally. *Id.*, ¶¶34-35. Darnell’s further description of using databases to manage content aggregation and access indicates a server-centric model that distributes content from a centralized location to various clients.

158. Darnell’s network structure also supports flexible and scalable access to the aggregated feed content. The description of the server system 104 accessing hosts 120 to collect feed content and aggregating it for user access via client systems would have aligned with a POSITA’s understanding of a cloud-based system. For

example, the '658 patent describes a “[c]loud-based computer system or architecture [that] includes multiple content data sources or service provider servers that are linked to a log-in server or master server via the internet/intranet.” ’658 Pat., 1:43-46, 4:27-30. This disclosure is like Darnell’s use of a central server to distribute aggregated content over a network to client devices and indicates a cloud-based approach to handling, storing, and distributing feed content for user access.

159. Finally, Darnell’s specification of a wide range of content types—digital audio, video, and text—that may be aggregated on the server for user access further solidifies its applicability as a cloud-based system. Darnell, ¶36. The ability of the server to manage multiple data types and serve them to various client devices over a broad network structure fulfills the general characteristics of a cloud-based computer system, providing centralized storage, dynamic content management, and networked access.

b. Claim 1, limitation [a], “master server with a processor and memory, wherein the master server accessible over the internet/Intranet from remote computer devices by a user”

160. In my opinion, Darnell discloses or at least suggests the limitation of a “master server with a processor and memory, wherein the master server [is] accessible over the internet/Intranet from remote computer devices by a user.” Darnell’s server system 104 is a “master server” as claimed, as it provides the components of processor and memory required to support the system’s aggregation

and distribution of content across multiple remote client systems. In Figure 1 of Darnell, server system 104 (master server) “includes a server 108 and a database 110.” Darnell, ¶33. Darnell further illustrates the server in Figure 5, which includes a CPU 502 and memory 504. *Id.*, ¶71. Memory 504 (memory) “includes high-speed random access memory, such as DRAM, SRAM, DDR, RAM or other random access solid state memory devices,” and CPU 502 (a processor) executes instructions. *Id.* Server system 104 may be implemented on a single computer or a distributed system of multiple computers. *Id.*, ¶35. A POSITA would have understood that Darnell’s server system 104, for example as implemented in a single computer, is a master server because, as explained below, it aggregates feeds from multiple sources and distributes them to clients. ’658 patent, 4:7-10.

161. I have annotated Darnell’s Figure 5 below, which shows memory 504 outlined in red and the processing unit outlined in blue.

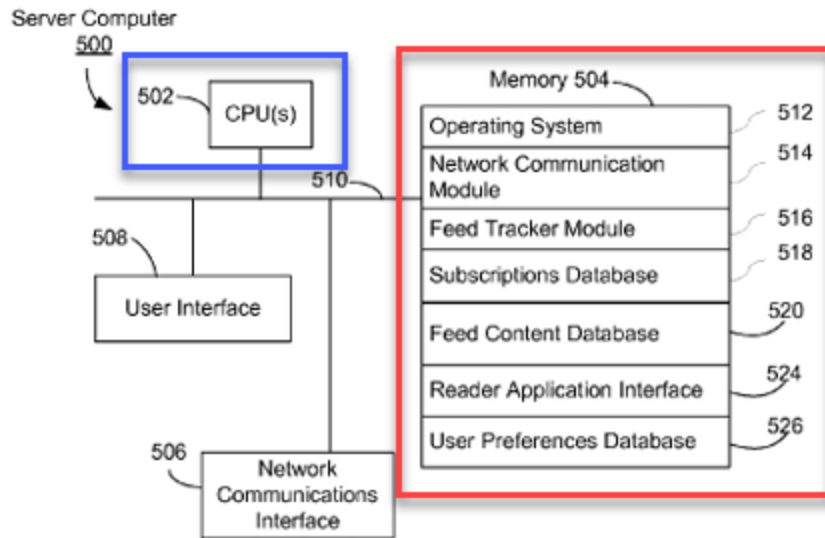


Figure 5

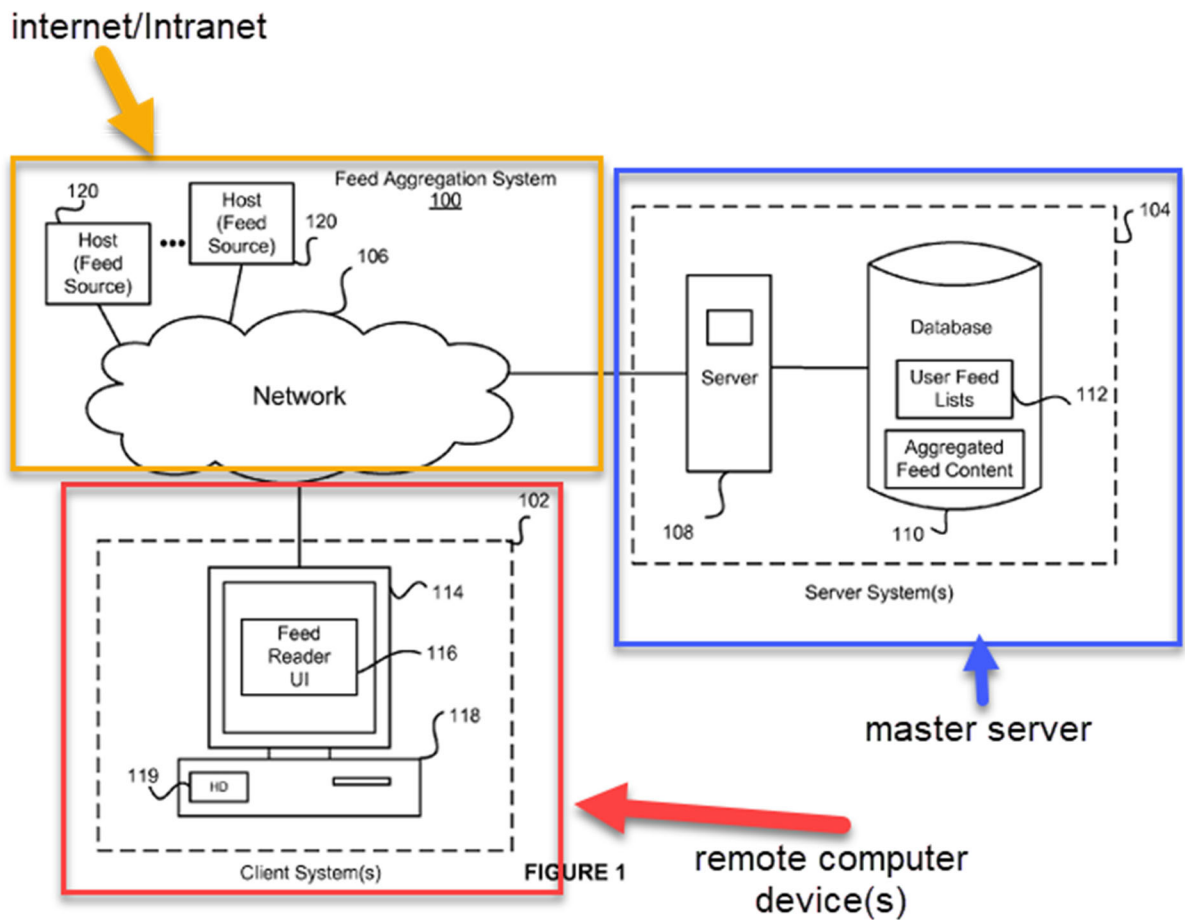
Darnell, Fig. 5.

162. Further, Darnell explains that server system 104 may be implemented on a single computer, housing both the memory and CPU in a unified configuration. *Id.*, ¶35. This single-server configuration would support the functionality of a “master server” as described in the ’658 patent, where a master server facilitates centralized control over content data and provides access to client systems through a unified interface. ’658 Pat., 4:7-10.

163. Importantly, Darnell also describes how users can access the server system 104 remotely through client devices, indicating its accessibility over “network 106,” which may be any combination of LAN, WAN, or the Internet. Darnell, ¶32. This accessibility over a network enables Darnell’s server to meet the requirement of a master server being “accessible over the internet/Intranet from

remote computer devices by a user,” as recited in the ’658 patent claim. Darnell’s server system 104 is accessible to users operating a client system 102, which may be a “computer 118 or computer controlled device, such as a personal digital assistant (PDA), cellular telephone, or the like.” *Id.*, ¶39. This structure allows users to connect with the server and view content on their respective devices, demonstrating that Darnell’s server system is indeed configured for remote accessibility through the Internet or intranet.

164. I also annotated Figure 1 of Darnell, which shows the master server highlighted in blue, the internet/Intranet highlighted in yellow, and the remote computer devices highlighted in red.



Darnell, Fig. 1.

165. Therefore, in light of the teachings in Darnell, it is my opinion that a POSITA would have recognized Darnell's server system 104 as a "master server with a processor and memory" that is accessible over the internet/intranet from remote computing devices by a user.

- c. **Claim 1, limitation [b1], "a master application stored in the memory of the master server and that executes on the processor of the master server and that includes software coded for organizing content data and content applications from multiple content**

sources or service providers pre-selected by the user”

166. Darnell discloses or suggests a “master application” that is stored in the memory of a master server and executed by a processor, containing software designed to organize content data and applications sourced from multiple content providers. Darnell describes server system 104 as equipped with memory 504, which stores several programs, modules, and data structures, collectively functioning as a master application. Darnell, ¶¶70–80; Fig. 5 (annotated below).

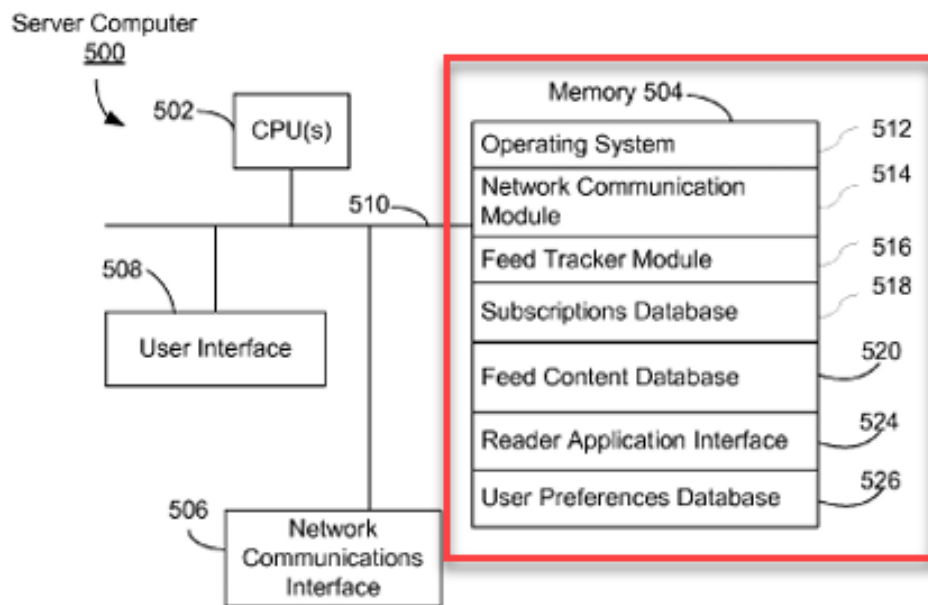


Figure 5

Id., Fig. 5.

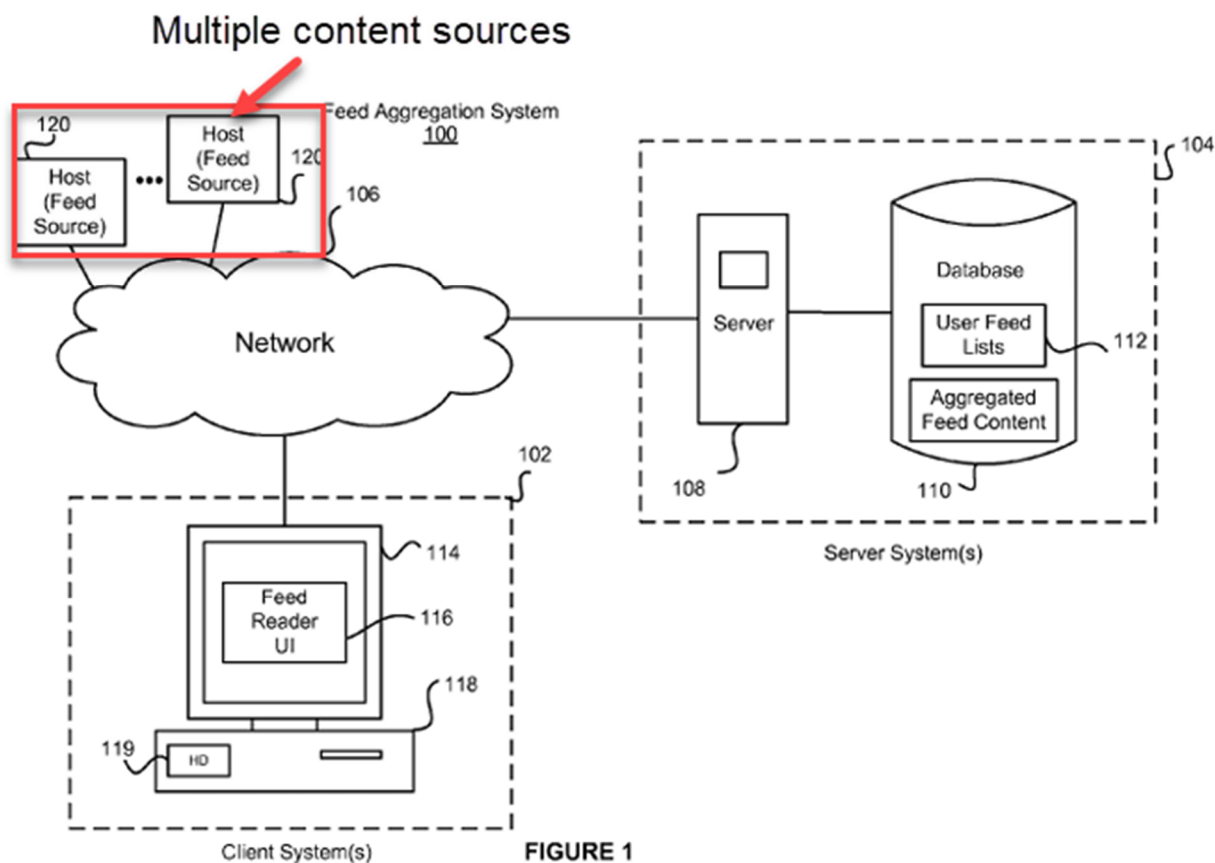
167. Darnell’s feed content database 520 “stor[es] content items from content feeds and user information about content items.” Darnell, ¶76. Such content “may be any machine-readable data, including but not limited to web pages, images,

digital audio, digital video, Portable Document Format (PDF) documents, and so forth.” Darnell, ¶36. Furthermore, a content item “may include the content associated with the item itself..., a title..., and/or a description of the content, a network location or locator (e.g., URL) of the content, or any combination thereof.” *Id.* The content data is received at the server and stored remotely in a database before being sent over the Internet to the user through the feed reader interface. *Id.*, ¶¶33 (“Server 108, sometimes called a front end server, retrieves the content feeds from the feed sources 120, and also provides an interface between the server system 104 and the client systems 102.”), 34 (“The server system 104 stores content items in the content feeds in the database 110.”), 40, 76, 81, 114; ’658 Patent, 4:1-3.

168. These programs and modules correspond to distinct sets of instructions performing specific functionalities. The server also includes a database 520 that stores content items and user information about these items. Darnell, ¶76. Additionally, Darnell discloses a subscriptions database 518, which retains information regarding users’ content feed subscriptions, and a user preferences database 526 that stores user-specific interface preferences. *Id.*, ¶¶75, 78. Subscriptions database 518 “stor[es] information about the content feed subscriptions of users of the system.” *Id.*, ¶75. Reader application interface 524 is “for exchanging information with the feed reader modules in one or more client computers,” and user preferences database 526 is “for storing user-specific

information, including user preferences with regard to the user interface.” *Id.*, ¶¶77-78. Furthermore, Darnell’s server includes an operating system to run the programs stored in memory. *Id.*, ¶72. Darnell’s Figure 5, for instance, illustrates this setup, where memory 504 and CPU 502 interact to carry out the functions required of a master server application. *Id.*, Fig. 5. A POSITA would have understood that Darnell’s modules and data structures on the server collectively are a master application that resides in memory 504 (memory) of server system 104 (the master server) and executes on the CPU 502 (the processor).

169. Darnell’s server system 104 also communicates with multiple feed sources 120, which represent various content sources or service providers. The specification defines “content feed” as a resource or service that offers lists of content items present or updated at feed sources. *Id.*, ¶36. Darnell explains that “[a] content feed (or channel) ... provides a list of content items that are present, recently added, or recently updated at a feed source 120.” *Id.* “A user interfaces with the server system 104 and views content items at a client system or device 102” via “a feed reader user interface 116 that may be presented on the display 114” and that allows the user to “create a list of feed subscriptions” (preselected by the user). *Id.*, ¶¶39-40. The feed sources identified in Figure 1 are the “multiple content sources” of claim 1 in the ’658 patent.



Id., Fig. 1.

170. Users can interact with the server to view these content items on client systems (i.e., remote computing devices) and can further refine the content they wish to see by selecting specific feed subscriptions. *Id.*, ¶39. A user interacts with Darnell’s system using a feed reader interface that allows them to create a personalized list of feed subscriptions, thus effectively preselecting content sources or providers based on individual preferences. *Id.*, ¶40. This list is displayed to the user through a web-based user interface and includes items from the feeds to which the user has subscribed. *Id.*, ¶41. Darnell’s webpages present an “ordered list” of

items to the user that is organized either by timestamp or “a score that takes into account both the content of the items and the timestamps of the items” (organizing content data and content applications). *Id.*, ¶¶43-45, 47, 49, 51-58, Figs. 3A-D. A POSITA would have understood that the program operating on server 104 that sends webpages to the remote computer performs functions to create the ordered list.

171. Darnell’s content feeds are broad in scope, covering formats like web pages, images, digital audio, video, and PDF documents, with each item containing details such as titles, descriptions, and URLs. *Id.*, ¶36. The stored data at server system 104 serves as the foundation of content items displayed to users through the feed reader interface. *Id.*, ¶¶33–34, 40, 76, 81, 114. A POSITA would have identified this broad variety of content as “content data,” with each item serving as data stored on the server and accessible through the server’s service provider infrastructure. ’658 Pat., 4:1–3.

172. The user interface in Darnell presents these content items in either a compact or expanded format, with JavaScript enabling client-side tasks, such as marking items as read or updating user preferences, by manipulating the HTML or XML coding of the displayed webpage. Darnell, ¶¶30, 41, 53-54. The JavaScript programming also enables the rendering and layout of content in compact or expanded views, a feature that a POSITA would have identified as part of the “content application” that accesses, displays, and organizes content data as required.

Darnell's user interface uses "Frames" to partition the webpage and a "JavaScript application" that "receives content items from the server system 104, manages the rendering of those content items at the client, and also performs the client-side aspects of other tasks, such as marking items as read, adding and removing content feed subscriptions, labeling content feed subscriptions and/or individually selected content items, selecting or revising user preferences, and the like." *Id.*, ¶41. A POSITA would have understood the compact and expanded views of the content data in the user interface is controlled by the HTML or XML programming of the webpage, including the JavaScript.

173. In Darnell, the use of "Frames" and a "JavaScript application" in the user interface allows for the efficient display, interaction, and organization of content on the client side while relying on server system 104 to retrieve and deliver the actual content items. Darnell's approach of using JavaScript to manage "Frames" enables content to be dynamically loaded, rendered, and refreshed within different sections of a single webpage without the need to reload the entire page, which improves both performance and user experience.

174. Frames are a method for partitioning a webpage into separate, independently scrollable sections, each capable of loading different content. By using frames, Darnell's system can maintain distinct areas within the user interface for various types of information, such as content feeds, control options, and settings,

without disruption to other sections when updates occur. JavaScript plays a crucial role in managing these frames, as it allows the system to asynchronously update specific sections by fetching content from the server without requiring a full-page refresh. This means users can view newly retrieved content or interact with updated data while preserving the context of their existing view.

175. Technically, JavaScript can access the Document Object Model (DOM) of the webpage, which represents the structure and content of the interface. By doing so, the JavaScript application in Darnell is able to control each frame's content individually, programmatically load and display new content items, and perform various client-side functions, such as marking items as read or filtering based on user preferences. This architecture offloads much of the display and interaction logic to the client side, which not only minimizes server load but also enhances responsiveness from the user's perspective.

176. For example, when a user interacts with the interface by marking an item as read or subscribing to a new feed, the JavaScript application registers this action, updates the relevant frame, and sends the request back to the server asynchronously. Darnell's use of JavaScript here is equipped to handle these asynchronous tasks, ensuring that interactions such as adding or removing subscriptions or labeling content are reflected in real-time on the client side without reloading the entire interface.

177. Thus, Darnell's server system 104, housing memory and CPU that execute stored programs to aggregate, organize, and display data from preselected content providers, meets the requirements of claim element 1[b1]. A POSITA would have understood the software displaying the compact and expanded format of the content item is a "content application" able to change the display of the content when selected. *See* '658 patent, 4:4-6 ("Content applications are software applications that accesses display and organized content data [sic], generally from a web-sites."). A POSITA would have also understood these content items and webpage programming as "content data and content applications" because Darnell's user interface contains data accessible from the server software application to display and organize the content.

d. Claim 1, limitation [b2], "and streamed to the master server simultaneously in real time from their respective servers on a master web-page and are viewable from the remote computer devices through the master server on the master web-page"

178. Darnell discloses or suggests the elements recited in claim limitation 1[b2]. Claim 1 specifies that "content data and content applications" are "streamed to the master server simultaneously in real time from their respective servers on a master web-page and are viewable from the remote computer devices through the master server on the master web-page." The '658 patent does not define "simultaneously" or "real time" in the specification; however, a POSITA would have

understood that, in the context of the '658 patent specification and file history, these terms to mean that the content from more than one source is sent to the server when it is available. In other words, these terms signify that the content from multiple sources is transmitted to the server when it becomes available, without significant delay between content sources.

179. For example, in column 1, lines 43-46, the '658 patent describes a cloud-based computer system that “can include the delivery of any kind of content data and/or services in real-time, which extend the capabilities to a remote computing device, typically over the internet.” This description of a cloud-based system means that content updates and transmissions occur with minimal lag, which a POSITA would have recognized to ensure users can access the most current data from multiple sources in real time.

180. Further, in column 4, lines 1-3, the '658 patent refers to “content data” as “data or information or information that is stored remotely on servers that are generally accessible over the internet/intranet via service provider servers.” This setup indicates that content is dynamically available and must be readily retrievable by the master server. The expectation of prompt retrieval suggests that content updates from each service provider would appear on the master server concurrently, given the system’s design to integrate various feeds into a cohesive, accessible interface.

181. Moreover, in column 4, lines 4-6, the '658 patent discusses “content applications” as software that “accesses[,] display[s,] and organize[s] content data, generally from a web-site.” This description shows that as content updates on a source server, the master application, which organizes and displays content on the master web page, retrieves this content promptly to ensure users experience minimal delay when viewing updated data.

182. Therefore, a POSITA would have understood “simultaneously” and “real time” as describing a system where content from multiple sources is updated and displayed as soon as it becomes available. The cloud-based architecture, as described in the patent, relies on minimized latency to fulfill its purpose of providing access to up-to-date content across various sources.

183. Darnell's Figure 1, alongside Darnell's description, depicts content items streamed to server system 104 (the master server) from their respective hosts or feed sources 120. Darnell, ¶32. The terms “server” and “host” are frequently used interchangeably and mean the same thing. A “host” of information “serves” that information to other computers; hence the similarity between the two terms. Darnell discloses that “server system 104 accesses content feeds from the feed sources 120” through server 108, which not only “retrieves the content feeds from the feed sources 120” but also provides “an interface between the server system 104 and the client systems 102.” *Id.*, ¶33. Accordingly, Darnell reveals that content from multiple feeds

reaches the server and is then delivered to the user, presented in the user's feed with timestamps organized chronologically. This is evident in Darnell's Figures 3A through 3D, which depict content feeds arriving and appearing sequentially to the user. *Id.*, Figs. 3A-3D, ¶¶51, 109.

184. Additionally, Darnell describes server 108's "feed tracker module 516," which retrieves content items "periodically and/or as content feeds are updated." *Id.*, ¶74. Given that the feed tracker fetches items when they are updated, Darnell's system receives content in "real time," as would have been understood by a POSITA. Darnell reinforces this concept by explaining that "feed items from a content feed source can be considered to be a stream of items," thus conveying the "streaming" nature of these content feeds. *Id.*, ¶31. Accordingly, Darnell's system streams content from multiple sources as updates become available, meeting the '658 patent's claim of content feeds streamed "simultaneously in real time."

185. As further explained above with regard to limitation 1[b1], Darnell enables the user to view content via a feed reader user interface 116. *Id.*, ¶41. This user interface 116, implemented as a web-based display, includes "Frames" for partitioning content and utilizes "JavaScript" for rendering and organizing content items at the client side. *Id.* When a user scrolls to the end of the visible list of content items, the system triggers a "fetch" command to retrieve additional items from the server, allowing the new content items to be appended to the list as the user scrolls.

Id., ¶112. The content items are thus displayed to the user in a continuous, updated stream, meaning that the server system 104 receives, retrieves, and then displays new content items to the user at the client system (remote computing device) in “real time” as they scroll through the feed. Darnell's Figure 3A exemplifies this, with a scroll bar shown on the user interface webpage for navigating through content. *Id.*, ¶54, Fig. 3A (annotated below).

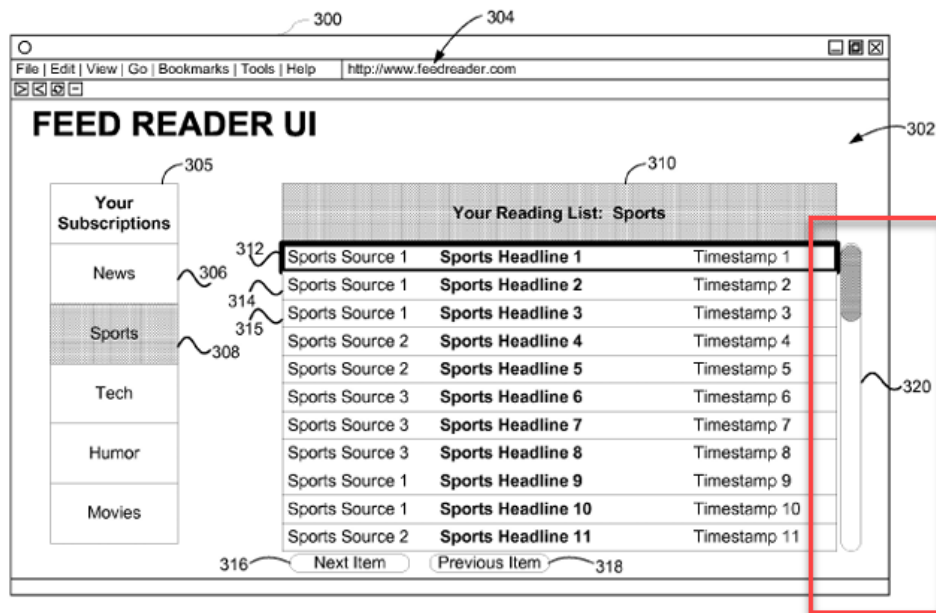


Figure 3A

Id., Fig. 3A.

186. A POSITA would have recognized that Darnell’s user interface is a “master web page,” meeting the ’658 patent’s description of a web-based interface that allows for scrolling or paging through “content data and/or content application

from multiple content data sources and/or service providers in real time” (’658 patent, 4:24-26). *Id.*, ¶41.

187. Therefore, Darnell’s cloud-based system discloses streaming content from multiple feed sources 120—wherein multiple content sources are “streamed... simultaneously... from their respective servers”—and sent to server system 104 (the master server). The content is rendered viewable through the feed reader user interface webpage, which features a scroll bar on a client system (remote computing device). Darnell’s feed reader user interface webpage is a “master web page” that presents content “simultaneously in real time” as updates occur.

- e. **Claim 1, limitation [b3], “wherein the master application programs a processor on the remote computer devices to operate and execute a slide tool program for scrolling or paging through the content data or the content applications from the multiple content sources or service providers simultaneously and through the master server in real time from the master web-page and from the remote computer by operating the slide tool program from the remote computer; and”**

188. It is my opinion that Darnell discloses or suggests this limitation. Darnell’s feed reader UI 116 is received from server system 104 executing modules and data structures (the master application) as a program (e.g., webpage) for execution on the processor (programs a processor) of the client system(s) (the remote computer devices). The ’658 patent describes “a slide tool that scrolls or pages through content data.” ’658 patent, 4:24-26. The claim does not limit the type of

slide tool that must be used; it only requires that the tool be suitable “for scrolling or paging through” content. A POSITA would have understood that “a slide tool program” is software implementing a slide tool, and a POSITA would have further recognized that this term includes a scrollbar. At the time of invention, scrollbars were widely known and commonly used in web interfaces as tools that allow users to navigate through displayed content.

189. Darnell’s disclosure supports this understanding, describing a feed reader user interface that includes a scrollbar 320 used to navigate an ordered list 310 of content items from various feeds. Darnell, ¶54. This scrollbar is embedded within the user interface webpage and operates as a slide tool that enables a user to scroll through the content in real time as the feed reader dynamically updates and displays content from multiple sources.

190. Darnell also explains that its user interface is constructed in a web-compatible language, such as HTML or XML, and that the webpage includes embedded scripts, such as JavaScript or ECMAScript, allowing the interface to interact with the server system. *Id.*, ¶41. These scripts, embedded in the webpage served from the server system 104, are executed when the webpage is rendered. Specifically, the feed reader module 420 script within the HTML enables a user to scroll through the content items served from the server. *Id.*, ¶67. This architecture is like the architecture described in the ’658 patent, where the master server

“programs” the client-side device to display the feed reader user interface, which includes a scroll bar allowing continuous, real-time access to updated content.

191. Moreover, as Darnell explains, the scrollbar’s function operates from the client system, enabling the user to interact with the feed interface as updated content is dynamically loaded and displayed. Darnell’s server architecture programs the remote client device to interact with the feed content on the master webpage and provides a scrolling tool for viewing multiple content sources through the master server in real time. Accordingly, Darnell’s feed reader user interface, delivered by the server to the client device, includes a scrolling tool that allows for real-time navigation of aggregated content that meets the claim limitation 1[b3].

f. Claim 1, limitation [c1], “a remote tool box feature application that allows the user to write code, select code or delete code to enable and disable tool features that control how sub-sets of the content data or the content applications from the multiple content sources or service providers are received, displayed or organized,”

192. Darnell discloses or suggests a “remote tool box feature application that allows the user to write code, select code or delete code to enable and disable tool features that control how sub-sets of the content data or the content applications from the multiple content sources or service providers are received, displayed or organized.” The ’658 patent defines a toolbox as “a set of sub-features...that allow a web-page layout and/or operation to be customized by linking tools with content

applications.” ’658 patent, 4:16-20. Darnell similarly discloses that the feed reader interface includes a JavaScript application, which manages the rendering of content items and “performs the client-side aspects of other tasks, such as marking items as read, adding and removing content feed subscriptions, labeling content feed subscriptions and/or individually selected content items, selecting or revising user preferences, and the like.” Darnell, ¶41. Darnell’s server also includes “a subscriptions database 518 for storing information about the content feed subscriptions of users of the system” and “a user preferences database 526 for storing user-specific information, including user preferences with regard to the user interface.” *Id.*, ¶¶75, 78.

193. As shown in Figure 6C, Darnell provides a “data structure for user preferences,” which is stored in a table format or through sparse table storage mechanisms. *Id.*, ¶¶84, 86. A POSITA would have understood that these data structures and databases in Darnell would be implemented in a database language such as SQL.

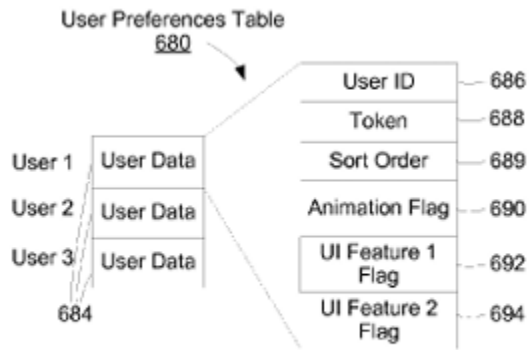


Figure 6C

Id., Fig. 6C.

194. Accordingly, Darnell’s system discloses a subset of features in the user interface (a remote tool box feature) that allows users to add (write) or modify (delete) user preferences in a database (code), which enables the customization of user interface functionalities. These preferences include fields like Animation Flag 690, UI Feature 1 Flag 692, and UI Feature 2 Flag 694, which can control aspects such as “formatting and color of a user interface,” “sort order,” and “interface feature flags.” *Id.*, ¶85. A POSITA would have understood that these user preferences interact with the server-side application to modify what content items are displayed and how they are displayed, thus enabling and disabling tool features that control the reception, display, and organization of content data or applications from multiple sources.

195. Darnell also discloses that users can “create filters or labels to be applied to content feeds and/or content items.” *Id.*, ¶40. These filters and labels are

transmitted to the server system and written in the database, which, as explained above, corresponds to writing code as claimed because a user creates (writes) the filters and labels. Darnell, ¶40. This interpretation is consistent with the plain and ordinary meaning of the claim term, which Patent Owner seems to agree with based on Patent Owner's infringement contentions. EX1014. I do not take any position with respect to written description or indefiniteness issues related to this claim limitation.

196. For example, as shown in Figure 6D, Darnell discloses the ability for users to write labels and apply checkmarks to certain content feed items within a “user subscription table.” Darnell, ¶¶87-90, Fig. 6D. The labels created by the user are stored in the subscriptions database on the server, and the server application subsequently utilizes these labels to construct the webpage, which displays them in the feed reader interface. The feed subscriptions (e.g., Feed ID) and labels written by the user (writing code) and stored in the subscriptions database are used by the server application to construct the webpage using those labels. *Id.*, ¶51.

User Subscription Table
625

640	Stream 1 630	Stream2 632	Stream3 634	Stream4 636	Stream5 638	...
642	Feed ID: URL1	Feed ID: URL2	Feed ID: User1:tech	Feed ID: User2:friends	Feed ID: User25:tech	...
User 1	√ label:tech	√ label:tech				
User 2			√ label:friends		√ label:friends	
User 3	√			√		
644	646	⋮				

Figure 6D

Id., Fig. 6D

197. Darnell further explains, for instance, that “Feed reader user interface 302 includes a list 305 of labels, also referred to as folders, associated with content feeds to which a user has subscribed.” *Id.*, ¶51. As illustrated in Figure 3A, these labels are displayed on the webpage and can be selected to adjust the display of the ordered list of content items. *Id.*, ¶51. I’ve annotated Figure 3A below to highlight the labels.

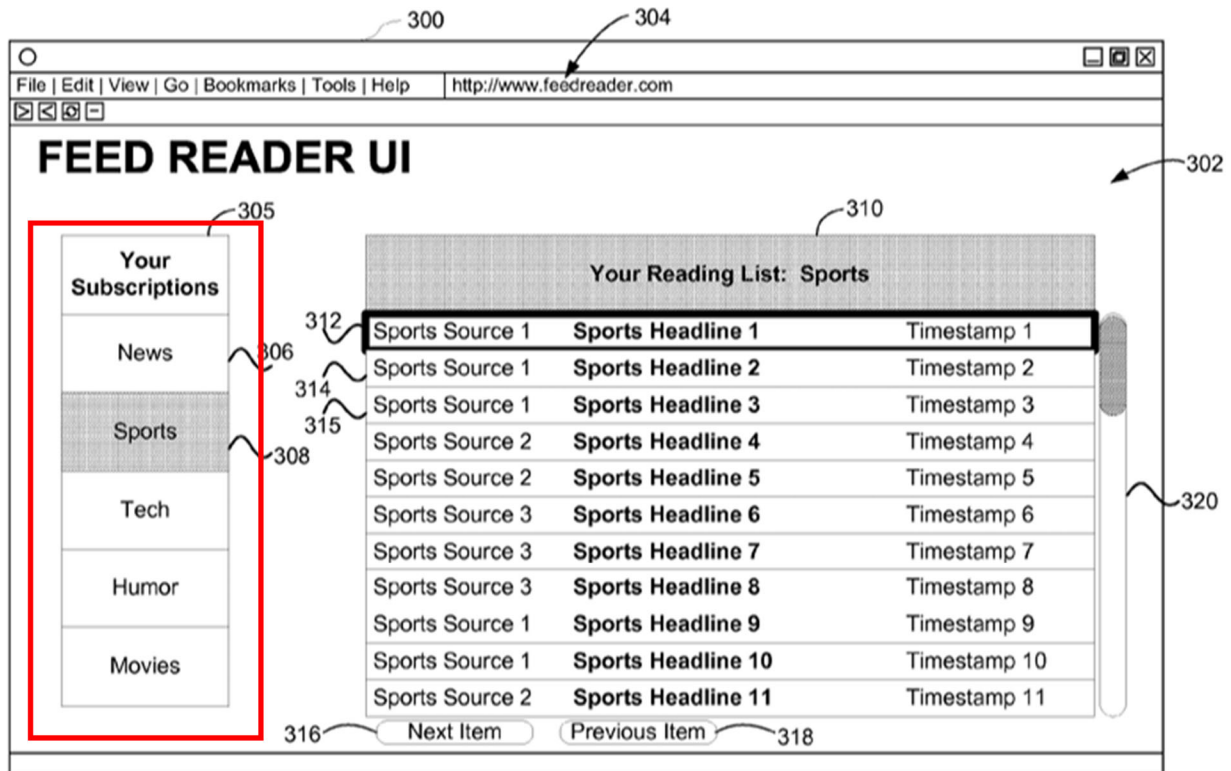


Figure 3A

Id., Fig. 3A.

198. Accordingly, Darnell discloses a user interface webpage with JavaScript applications (remote tool box feature application) allowing users to create and modify feed labels and user preferences (write, select, or delete code to enable or disable tool features) that alter the content and display of the user interface webpage (that control how sub-sets of the content data or the content applications from the multiple content sources or service providers are received, displayed or organized).

- g. **Claim 1, limitation [c2], “wherein the master server collects analytics based which tool features are enabled to target specific products, services or**

information to the user.”

199. In my opinion, Darnell discloses or suggests the claimed “analytics” functionality that collects user data based on the enabled tool features to tailor the display of products, services, or information to the user. Specifically, Darnell’s system includes a user preferences database 526, stored on the server (the master server), which accumulates user-specific information for purposes of customization. Darnell, ¶84. The user preferences table 680 illustrated in Figure 6C contains fields like user data 684, user ID 686, and token 688, which authenticate and identify user-specific activities. *Id.* The preferences stored here also include a sort order flag 689, which defines the order in which content items are presented to the user—whether sorted chronologically or by a ranking based on quality and timeliness of content. *Id.*, ¶85 (“[T]he user data optionally may include a sort order flag 689, for specifying the order in which to display content items on a user interface, such as user interface 302 (FIG. 3A).”). These customization fields and flags collect data on the user’s interactions with the interface, which a POSITA would have recognized as analytics. I’ve annotated Figure 6C of Darnell below to show the User Data 684 that is part of the user preferences table 680.

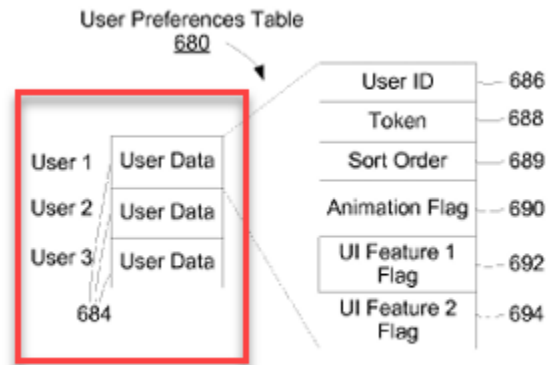


Figure 6C

Id., Fig. 6C.

200. Further, Darnell’s user preference database provides information about what specific content types the user accesses, allowing for targeted recommendations. By capturing data related to the “recency” or “quality” of content items, Darnell’s server gathers insights about the user’s preferences and interest patterns, which can guide the system’s targeted display. *Id.*, ¶51. User preferences and labels (enabled tool features), for example, determine the displayed content. *Id.*, ¶¶51-52. Thus, Darnell’s user preferences database 526 collects information about the user and content (analytics) and uses that information to display content items based on the user’s preferences (enabled to target specific...services or information to the user). A POSITA would have understood that these characteristics—such as recency and quality—constitute analytic data since they provide insights from the data about user interaction patterns and preferences. A POSITA would have

understood that keeping track of content quality and freshness of the content corresponds to “collecting analytics” as claimed.

201. Darnell also discloses the enabled tool features on content presentation. As discussed in ¶85, the user preferences that Darnell collects impact the presentation of content items, allowing for a personalized user experience based on preferences stored in the server database. These preferences include flags like Animation Flag 690, UI Feature 1 Flag 692, and UI Feature 2 Flag 694, which can be used to adjust aspects such as “formatting and color of a user interface,” sort order, and other interface settings. These flags act as data points regarding which features are enabled by the user and, thus, are “analytics” collected by the server to determine the user’s preferences. Darnell, ¶85.

202. In Darnell, the word “regency” appears to be a typographical error and, instead, should be “recency.” “Recency” of data means how recent a user accessed that data. A POSITA would have understood that more recently accessed content is likely more relevant to a user than content that has not been recently accessed by a user.

203. In Darnell’s system, the notion of a “score” that “takes into account both the content of the items and the timestamps of the items” (Darnell, ¶51) could incorporate a type of regency where content is evaluated based on predefined criteria that might include the importance of the content in the overall context, its value to

the user, or its weight within the aggregation structure. This scoring system could prioritize or order content not only by how recent it is but also by how “regent” it is according to the system's prioritization metrics.

204. To the extent it might be argued that Darnell alone does not disclose all aspects of the analytics functionality, Kim provides a complementary teaching that would render this element obvious. Kim’s system collects and uses a user’s *usage history*, which tracks the user’s interaction with content feeds—such as how often each feed is accessed and the read-to-unread item ratio. Kim, ¶¶69-70. A POSITA would have recognized that tracking a user’s page interactions, such as visits and read/unread status, constitutes analytics since these actions reflect analytic data commonly collected to understand and optimize user engagement.

205. A POSITA would have understood “analytics” as a process involving the collection, examination, and interpretation of data to derive meaningful insights, particularly in enhancing and personalizing the user experience. In the context of the ’658 patent, this would extend to tracking a user’s interactions with specific features and tools to adapt content presentation dynamically based on user behavior.

206. For example, as the ’658 patent describes, users have options to configure their content experience by enabling or disabling features, which then influence the organization, display, and prioritization of content data on the interface. Tracking these configurations and user interactions provides a feedback

loop—analytics—on user preferences, which can then inform subsequent content organization and feature customization, targeting specific content items, services, or applications to the user based on inferred interests.

207. This is consistent with definitions of “analytics” from Oxford Languages and other dictionaries and reflects how a POSITA would have recognized the importance of tracking user engagement to optimize system responses and interface functionality. For example, these dictionaries state:

208. **Oxford English Dictionary (OED):** “[A]nalytics” refers to “[t]he systematic computational analysis of data or statistics.” EX1047. A POSITA would have understood this definition to include the computational tracking of user content views and device usage on the server, which Darnell and Kim disclose as methods to tailor content delivery to user interactions.

209. **Cambridge Dictionary:** “[A]nalytics” is defined as “a process in which a computer examines information using mathematical methods in order to find useful patterns.” EX1046. This definition suggests that analytics involves gathering historical data on content views and device usage patterns, as seen in Darnell’s user account tracking and Kim’s usage history feature, to tailor content presentation to user behaviors.

210. Furthermore, Kim organizes content based on *traits* that users can select, as detailed in ¶58 and depicted in Figure 10. Traits serve as indicators of

content type (e.g., images, text, links), and the user's engagement with these traits provides insight into preferred content categories. Kim, ¶58. A POSITA would have understood that these content traits qualify as tool features under the '658 patent's description, since traits indicate specific content attributes, allowing the system to tailor presented information based on user engagement. A POSITA would have understood a user's selection and viewing of the content items is determined by the selected traits.

211. Kim's Figure 13 presents a further example of user-driven content organization, where summaries of content feeds are ordered based on factors such as frequency of user engagement and other interaction-based metrics. Kim, ¶¶67-71. Here, Kim's system presents headlines in a particular order based on a user's history with the content—an example of content being targeted to the user based on collected analytics. *Id.*, ¶69. For example, the feed application “presents headlines for content items in chronological order for the five feeds with the strongest usage history in region.” Thus, the order of the content feeds in Figure 13 are organized based on a user's usage history (to target specific...information to the user). A POSITA would have applied Kim's teachings to Darnell's ordered list system to create a more refined sorting methodology based on traits and engagement history.

212. Given the analogous teachings in Darnell and Kim, a POSITA would have been motivated to apply Kim's tracking of usage history and traits to Darnell's

user preference and display systems, expecting to achieve enhanced customization of content displays based on user-specific data. For example, a POSITA would have understood that Kim’s usage history (master server collects analytics) is based on which content and traits a user selects and views (based which tool features are enabled), which then determines, for example, the order of the summary content feeds (target specific...information to the user) and applied Kim’s teaching to Darnell. Darnell discloses the “items in ordered list 310 are ordered in accordance with a score that takes into account both the content of the items and the timestamps of the items.” Darnell, ¶51. Darnell, however, does not disclose how the score is calculated or what aspects of the content are “tak[en] into account.” But Kim discloses “analyz[ing] a feed's content for its traits... such as whether or not the feed content is primarily links, text, inline photos, enclosed media, and the like.” Kim, ¶¶25, 38, 58. For the reasons I explained in § XI.A.1, a POSITA would have been motivated to apply Kim’s analysis of content traits to Darnell’s “score” system in order to better define how Darnell’s “score” is calculated.

213. A POSITA would have had a reasonable expectation of success in using Kim’s traits and usage history to control how Darnell displays feeds based on scores because Kim shows that using such a technique to organize displayed feeds was known by a POSITA. Furthermore, organizing and presenting content to a user based on previous viewing of the content would have been a straightforward application

of known, conventional content aggregation methods used according to their known functions to yield predictable results. Applying this approach would have been a routine exercise in combining known techniques for user analytics, leading to predictable improvements in delivering targeted content to the user.

3. Independent Claim 7

214. Independent claim 7 is analogous to independent claim 1, but also includes a few additional terms that I explain below. In my opinion, the combination of Darnell and Kim renders obvious all of the claim limitations from independent claim 7 for the reasons I previously discussed in § XI.A.2 and the reasons I provide below. To streamline my discussion, I will incorporate analysis in § XI.A.2 through references to that section rather than repeat the entirety of my previous opinions.

215. I have also created this table to show the limitations of Claim 7 and their analogous limitations in Claim 1.

Claim 7	
Limitation	Substantially Similar Limitation(s)
7[pre]	1[pre]
7[a]	1[a], 1[b1]
7[b]	1[b3]-1[c1]
7[c]	1[b2]
7[d]	1[c1]
7[e]	1[c2]

a. Preamble of claim 7, “A computer system with”

216. As I explained in § XI.A.2.a, Darnell discloses a computer system. Darnell’s “content feed aggregation system” is a “computer system.”

b. Claim 7, limitation [a], “a master server with a memory and a processor for storing and executing a master application”

217. As I explained in §§ XI.A.2.b-XI.A.2.c, Darnell’s “content feed aggregation system” comprises a server system 104 (master server) that contains a memory and a processor for storing and executing a master application. I highlighted these components in Figure 5 of Darnell:

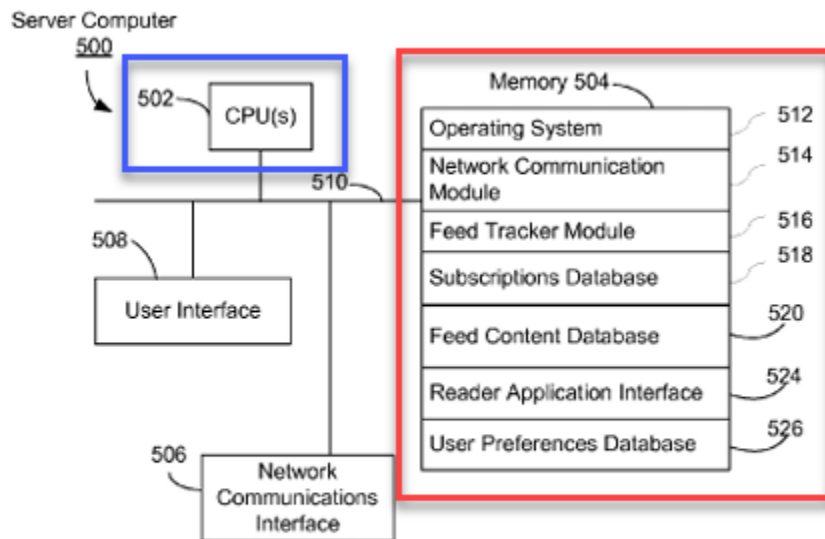


Figure 5

Darnell, Fig. 5.

c. Claim 7, limitation [b], “with a remote tool box feature that programs a processor on a remote computer with a slide tool program for scrolling or

paging through content data or content applications from multiple content data sources or service providers enabled by a user simultaneously on a web-page created by the master application and hosted on the master server using hardware on remote computer,”

218. To address claim limitation 7[b], it is important to first understand its similarities and distinctions from claim limitation 1[b3]. Both limitations reference functionalities that program the remote computer, yet claim limitation 7[b] specifies that a “remote tool box feature” performs this programming rather than the “master application.” A POSITA would have recognized that the “remote tool box feature” is a part of the master application, rather than an entirely separate entity. In other words, a POSITA would have understood that the “remote tool box feature” was part of the “master application” because a “tool-box” is a “set of sub-features” of an application. ’658 patent, 4:16-19. As I explained above in 1[b3], scripts (e.g., JavaScript or ECMAScript) program the remote computer, and as explained in 1[c1], a script is a tool box. This understanding is consistent with the ’658 patent’s specification, which describes the tool box feature as a “set of sub-features” within the application that allow customization of the layout and operation of a web page by linking tools with content applications. *Id.* This distinction clarifies that the tool box operates within the broader master application, directing functionality specifically at the remote device.

219. As explained in my analysis of claim limitation 1[b3] in § XI.A.2.e, Darnell discloses a user interface that utilizes embedded scripts, such as JavaScript or ECMAScript, to manage the rendering of content items on the client side. These scripts execute on the remote computer to manage content display and enable interactions, including marking items as read, adding or removing content subscriptions, and adjusting preferences for the user interface display. Darnell, ¶41. By running these scripts, Darnell’s server “programs” the remote computer to perform the functions set by the user’s preferences and interactions. This process is consistent with the remote tool box feature in claim limitation 7[b], which allows user-controlled content display and organization through such client-side execution of embedded scripts.

220. A POSITA would have understood this client-side programming as a form of a “remote tool box feature” within the master application, given that the scripts operate on the remote computer to control which content features are visible and accessible to the user. In summary, Darnell discloses claim limitation 7[b]. The embedded scripts provide the remote programming required, allowing the remote device to be customized according to user preferences, as facilitated by the master application’s remote tool box sub-feature.

- d. Claim 7, limitation [c], “wherein the content data or content applications are simultaneously streamed to the master server from servers of multiple content data sources or service providers over the internet in**

real time to an account belonging to the user,”

221. In my opinion, Darnell discloses or suggests claim limitation 7[c]. Although similar to claim limitation 1[b2], claim limitation 7[c] specifically requires that content is streamed “in real time to an account belonging to the user” rather than being “viewable from the remote computer devices through the master server on the master web-page.” In my analysis, a POSITA would have understood that Darnell teaches content streaming in real time to a user’s account. Darnell discloses a “user’s feed reader account” which consolidates content items that a user has either marked as read or unread and categorized or tagged with specific labels and characteristics, thereby creating a distinct “account” experience. Darnell, ¶¶84, 92, 99, 105.

222. A POSITA would have understood Darnell’s disclosure of the user’s account as supporting the claim’s concept of an account uniquely tied to the user’s preferences and activities, based on Darnell’s description that, upon login, users access their customized feed reader account with content tailored to their specifications. Darnell explains that the remote computer initiates content access by sending a “login cookie” to the server, which authenticates the user and grants access to the user’s feed reader account. This “login cookie” acts as a key, identifying and authorizing the specific account to receive content updates in real time. *Id.*, ¶92, ¶¶99-100, Fig. 7C. This real-time transmission and authentication process, which

directly links content to an individual account, fulfills the requirement for content to be streamed “in real time to an account belonging to the user.”

223. Moreover, Darnell’s content management, which allows the user’s account to categorize and interact with feed items in real time based on the user’s status and tags, is precisely what limitation 7[c] envisions: real-time streaming of content directly to a user-specific account, differentiating it from limitation 1[b2] where content is simply “viewable” through a master server. Therefore, Darnell’s detailed disclosure of user-specific feeds and real-time updating through authenticated accounts satisfies this claim element as would have been understood by a POSITA.

- e. **Claim 7, limitation [d], “wherein the remote tool box feature allows the user to write code, select code or delete code to enable and disable tool features that control how sub-sets of the content data or the content applications from the multiple content sources or service providers are received, displayed or organized”**

224. Limitation 7[d] is substantially similar to limitation 1[c1]. For the same reasons as I explained in § XI.A.2.f, Darnell discloses claim limitation 7[d].

- f. **Claim 7, limitation [e], “and wherein the master server collects history analytics based which tool features are enabled to target specific products, services or information to the user.”**

225. Limitation 7[e] is substantially similar to limitation 1[c2]. For the same reasons as I explained in § XI.A.2.g, Darnell discloses claim limitation 7[e].

4. Claim 2: “The cloud-based computer system of claim 1, wherein the content sources or service providers include one or more personal content account and at least one or more social media account belonging to the user.”

226. Claim 2 depends from claim 1. Claim 2 specifies a content aggregation system capable of integrating content from a user’s personal accounts, such as email and social media, into a unified feed. Darnell in combination with Kim discloses or suggests these additional limitations. Kim, for example, describes a content aggregation system that collects various forms of content, including both email and blog posts, which are commonly associated with user accounts and social media. Kim, ¶¶21-22, 33, 36-39, 45, 58-59, 72, Figs. 3-9, 11-13. A POSITA would have understood that this email content originates from a user-specific email account, thereby meeting the limitation of “personal content” associated with the user. Kim further specifies that content can be categorized based on metadata, indicating its source or nature, with certain blog posts categorized as “personal.” Kim, ¶42. This organization by metadata would have been understood by a POSITA as a categorization scheme, wherein personal blog posts could be identified as part of the user’s social media activity, therefore fulfilling the requirement for receiving both personal and social media content into the aggregator. A POSITA would have understood that a blog post with a comment section (as shown in Kim) is social media because in order for a user to “comment” on a blog post, the user needs an account to interact socially with other commentators on the blog post. For example,

an author of a blog post in Kim is able to socially interact with readers who leave comments. Furthermore, it was well-known that social media includes “public and private messages, blog posts, status updates and other communications on social media systems.” EX1008, ¶3. And “social media sources...include messaging networks such as Twitter™ and email, blogs, social networking sites such as Facebook™ and LinkedIn™, media sharing sites such as Flickr™ and YouTube™, forums and comment streams.” *Id.*, ¶24. Kim further explains that blog posts can be “categorized as personal in metadata associated with the feed source.” Kim, ¶42. The blog posts thus come from a “personal” source (e.g., a personal travel blog) that may include the user’s own blogs (belonging to the user) that originate from the user’s own account. A POSITA would have understood that the blog posts are the user’s own blogs, i.e., a social media account belonging to the user.

227. The blog posts thus come from a “personal” source (e.g., a personal travel blog) that may include the user’s own blogs (belonging to the user) that originate from the user’s own account. The user of Kim’s feed aggregator could thus monitor, for example, personally authored comments or comment activity on the user’s personally authored blogs. A POSITA would have been motivated to combine Kim’s email and blog feed content with Darnell’s feed reader UI to provide the user of Darnell’s system with additional content sources in a unified UI so that the user would not need to toggle back and forth between a blog UI and the feed reader UI.

228. Indeed, in my personal experience, I could access my email on my television prior to the priority of the '658 Patent. It was well known at the time of the priority date of the alleged invention that a user could access a web browser on a television, and a POSITA would have known that email is accessed via a web browser. *See, e.g.,* WebTV (EX1049, <https://webtv.zone/> and EX1050, <http://www.chicagonet.net/webtv.htm>, archived June 21, 2010).

229. Additionally, Kim's feed aggregator enables these personalized content streams to be integrated into the feed, allowing both email and blog content to be viewed within the same unified interface. A POSITA would have found this integration compatible with Darnell's system, which already utilizes a feed reader interface to consolidate multiple content streams. Darnell's approach, which describes content delivery in machine-readable formats such as HTML, PDF, and XML, is well-suited to the receipt of diverse media types, including email and blog data, given their compatible data structure. Darnell, ¶36. Moreover, Darnell's interface presents this aggregated content to users in a singular, accessible view, enabling users to avoid toggling between separate email or blog interfaces, thereby enhancing user experience by centralizing all relevant content. Receiving email and blog content would have been a straightforward application of known, conventional content aggregation methods used according to their known functions to yield predictable results.

230. Kim's use of metadata tagging and categorization for social media posts, when combined with Darnell's feed reader, would yield a predictable and reliable interface where users could receive and view personal and social media content without navigating multiple platforms. This functionality would be especially useful for users managing both email and blog content, as it simplifies access to a broader set of media types within Darnell's content aggregation framework.

231. Additionally, as I explain below for claim 4, the user would have to log into the Darnell-Kim feed reader. A POSITA would have therefore understood that user login would be a prerequisite to receiving blog posts and leaving comments, meaning the received content in the feed reader (blog posts) would be associated with a "social media account belonging to the user" as claimed. As shown in Kim's Figure 4, the comments section has individual user comments associated with names ("Memet" as one example in the figure), which a POSITA would have understood as being associated with a social media account.

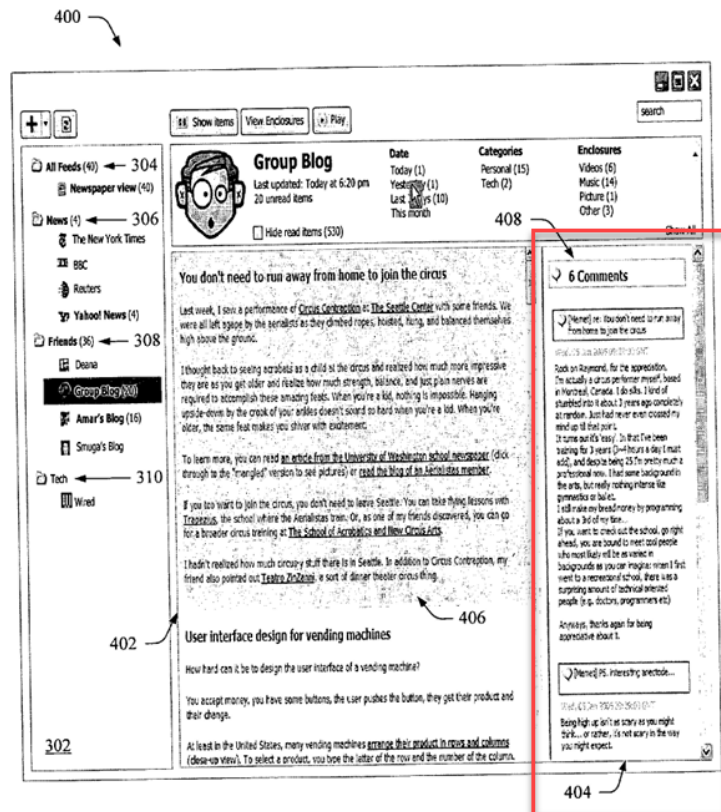


FIG. 4

5. **Claims 3 & 11: [Claim 3] “wherein the personal content account is an e-mail account”; [Claim 11] “wherein one of the content sources or service providers includes an e-mail web-site belonging to the user.”**

232. Claim 3 also depends from claim 1, and claim 11 also depends from claim 7. As I showed in my table in § V.C.1 above, these dependent claims are substantively identical, so I will address them here together instead of separately.

233. Claims 3 and 11 are closely related in that both address the functionality of receiving email content within the claimed cloud-based computer system but do so with slight variations in their scope. Claim 3 specifies that the “personal content account” disclosed in claim 2 is an “e-mail account,” whereas claim 11 specifies that

the content source or service provider includes an “e-mail web-site belonging to the user.” Both claims relate to the system’s ability to integrate and display email content.

234. The Darnell-Kim combination discloses or suggests the additional limitations of dependent claims 3 and 11. For example, Kim describes “a user’s email account” as a type of personal content account, and POSITAs would have recognized an email account as a standardized and common type of personal content account. Kim, ¶¶21-22, 33, 36-39, 45, 58-59, 72, Figs. 3-9, 11-13.

235. Dependent claim 11 recites that one of the content sources or service providers is an “e-mail web-site belonging to the user.” While this claim is similar to claim 3, POSITAs would have understood the term “e-mail web-site” to refer to a content source for email, such as a web-based interface for accessing email content. This includes webmail services, which enable users to access and manage email directly from their browsers.

236. Moreover, a POSITA would have understood that the method of receiving email content does not vary substantially based on its source. Whether from a dedicated email client or an email website, email content is standardized in formats like IMAP and POP, ensuring compatibility across systems. Darnell-Kim’s system, which retrieves content from various sources and displays it in a unified

interface, would support web-based email content as disclosed in claims 3 and 11. See Kim, ¶¶21-22, 33, 36-39, 45, 58-59, 72, Figs. 3-9, 11-13.

6. Claim 4: “The cloud-based computer system of claim 1, wherein the master application includes tools for enabling security features that limits access to one or more of the content sources or service providers based a user log-in belonging to the user.”

237. In my opinion, Darnell and Kim disclose or suggest this additional limitation. Darnell explains that “[f]or security reasons, it is desirable to verify the request is, in fact, an authorized request from a respective user of the feed reader system before processing the request.” Darnell, ¶99. This verification process ensures that a user’s access to content feeds on the feed reader system is restricted based on their authorization status. As part of this security mechanism, the user must provide credentials that the system authenticates before granting access to content from multiple sources. *Id.*, ¶¶91-102. Therefore, before any request for feed content (representing access to content sources or service providers) is approved by the server software (the master application), the user must confirm their identity to the system.

238. Darnell further discloses specific methods for user authentication, which a POSITA would have recognized as “tools for enabling security features.” For example, Darnell details the use of a “login cookie,” which includes a “token that uniquely corresponds to the user identifier” and a “login web page” displayed

to the user on the client computer. *Id.*, ¶¶99-102. Through this interface, the user logs into the feed reader system, confirming their identity and granting authorized access to content. *Id.* Darnell’s approach to securing content access is consistent with the concept in the ’658 patent, which describes a “tool” as a feature that enables the ability “to access, display and/or organize content data on a web-page.” ’658 patent, 4:11-15. By logging in, the user confirms authorization through an interface comprising a login cookie, token, and login screen, which are tools to manage security features on the server.

239. Figure 7C of Darnell illustrates this login protocol, confirming the security process flow. Upon attempting access, the user must present a “login cookie” with a unique token or log in manually through the login interface. Darnell, ¶100, Fig. 7C. This mechanism ensures that content access is restricted based on the login credentials associated with the user’s account. In this way, Darnell discloses a login-based security measure (user log-in belonging to the user) that is required by claim 4. *Id.*, ¶100.

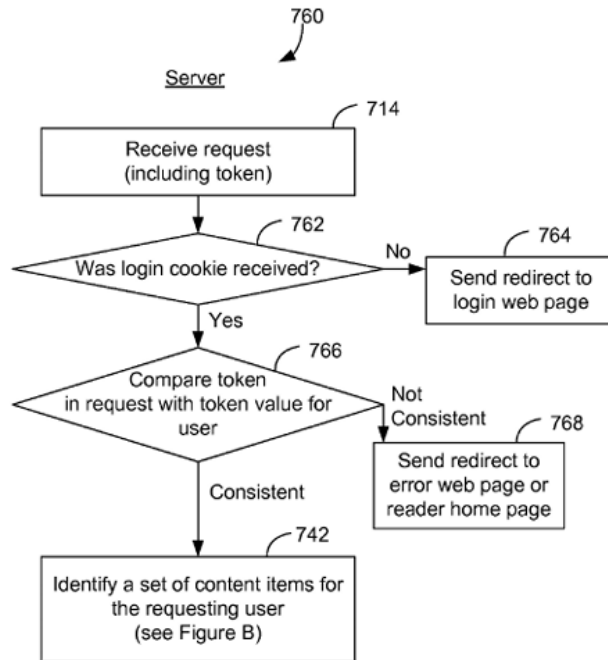


Figure 7C

Darnell, Fig. 7C

240. Thus, Darnell discloses software on the server (master application) that implements a secure login mechanism by combining a login cookie, unique token, and login screen (tools for enabling security features) to prevent unauthorized access to the feed reader and its associated content items. This structure limits access to content sources or service providers based on the user's verified login credentials, fulfilling the requirements of claim 4.

7. **Claims 5 & 8: [Claim 5] “wherein the master server collects and stores history analytics based content data and content applications viewed by the user”; [Claim 8] “wherein the master server collects and stores history analytics based on**

usage of the remote computer.”

241. Darnell and Kim disclose or suggest the additional limitations in claims 5 and 8. In particular, both references describe collecting and storing analytics based on a user’s interaction with content, whether through the viewing of specific items or general usage patterns. Darnell, ¶¶84, 92, 99-100, 105; Kim, ¶¶69-71. As explained above, Darnell’s feed reader system captures and stores details regarding a user’s interactions with content in the server’s databases. Specifically, Darnell’s user preferences database 526 stores information that tracks which content items a user has accessed or marked, such as read or unread status and labels or tags. Darnell, ¶¶84-86. This information is accessible through a user’s feed reader account, which stores history analytics based on the user’s viewing patterns and labels applied to content items, further refining the feed presented to the user.

242. Similarly, Kim’s usage history feature discloses tracking and storing analytics regarding content items a user has accessed or favored based on traits identified through “usage history.” Kim explains that this “usage history” data allows content feeds with higher frequency of user interaction to appear more prominently, which is as an analytics system that captures and organizes the user’s historical interaction with various content items. Kim, ¶¶69-71. In both Darnell and Kim, the captured history data is stored on the server (master server), and each

reference discloses the required tracking and storing of “history analytics based [on] content data” and usage as viewed by the user.

243. Claims 5 and 8 are closely related in that both address the “history analytics” feature of the ’658 patent, with claim 5 focused on the history of “content viewed by the user,” while claim 8 emphasizes “usage of the remote computer.” Dependent claim 8 is substantially similar to claim 5, except that it recites “history analytics based on usage of the remote computer.” A POSITA would have understood that the viewing history of the user on the client device results from using the client device. A POSITA would have also recognized that these claims overlap substantially in scope, as “viewing content” involves the usage of a computing device to access and display that content. Tracking a user’s history of content viewing therefore includes analytics based on the device’s usage patterns—such as frequency of access to particular content, time spent on different feed items, or specific filtering options engaged by the user. By storing these analytics on the server and presenting customized content to the user based on viewing history, Darnell and Kim meet the additional limitations of both claims 5 and 8.

244. In further support of this understanding, industry practices align with how analytics is understood to encompass both user-specific data—like viewing history—and device usage metrics. Dictionary sources as cited in § XI.A.2.g define “analytics” as data analysis focused on identifying patterns, which encompasses both

user actions on content (viewing patterns) and interactions with a device (usage behavior). Consequently, a POSITA would have understood that “history analytics” includes metrics on content items accessed and device interactions to tailor content delivery more effectively, as Darnell and Kim disclose. Accordingly, Darnell-Kim teaches the additional limitations recited in dependent claims 5 and 8.

8. Claim 9: “The computer system of claim 8, wherein the master server uses the history analytics to select content data or the content applications based on a user log-in belonging to the user.”

245. Claim 9 is similar to claims 1[c2] and 8, as it recites the use of *history analytics* to selectively provide *content data or content applications* to a user based on login authentication. Darnell’s disclosure of a server that organizes and transmits feed content to the user discloses this claim requirement. Specifically, as I described in claims 1[c2] and 8, Darnell’s server sends customized content via a webpage based on a user’s preferences or viewing history. Darnell, ¶¶84-85. Darnell discloses that the server's database stores user preferences and feed-specific labels, which are used to manage and present tailored content to the user in their feed reader interface (content data or content applications). *Id.*, ¶¶51, 84-85.

246. Further, Darnell incorporates login-based security features that restrict access to content based on user authentication, as discussed in claim 4. To access content through Darnell’s system, a user must log in to the server, which verifies the user’s identity through mechanisms like a “login cookie” or token associated with

their feed reader account. *Id.*, ¶99. Darnell’s login process ensures that each user’s personalized content feed (e.g., history analytics) is directly tied to their authenticated session. Thus, when a user logs into their account, the server tailors the displayed content data and applications based on that user’s preferences and past interactions.

247. Darnell emphasizes the importance of security and user authentication by noting, “[f]or security reasons, it is desirable to verify the request is, in fact, an authorized request from a respective user of the feed reader system before processing the request.” *Id.*, ¶99. This log-in protocol ensures that the server tailors content selection based on the user’s unique identifier, reflecting the limitation of claim 9 that calls for history analytics-driven selection of content based on a “user log-in belonging to the user.” Accordingly, a user can only access Darnell’s content feed aggregation system if the user is a “registered user of the system,” or, in other words, the user logs in successfully into the user’s feed reader account (based on a user log-in belonging to the user).

248. In addition, a POSITA would have understood that using history analytics to select content for an authenticated user, as Darnell describes, is a known, effective method for improving user experience and providing targeted content. By integrating a log-in feature with user-specific history analytics, the system achieves a tailored and secure content aggregation, disclosing the requirements of claim 9.

9. Claim 10: “The computer system of claim 7, wherein the user interface is an icon-based master web-page.”

249. Claim 10 specifies that the user interface must be “an icon-based master web-page.” Darnell discloses features that support this limitation by incorporating icons and menu items within its feed reader user interface, enabling users to manage their content subscriptions. Darnell describes that the feed reader interface “typically include[s] icons, menu items or the like for reviewing and managing the user’s content feed subscriptions, including removing and adding content feed subscriptions, and optionally labeling a respective content feed subscription with one or more user defined labels (or, alternately, assigning the respective content feed subscription to one or more user defined folders).” Darnell, ¶60. As shown in Figures 3A-3D, Darnell’s interface includes interactive icons that operate as navigational tools, providing a streamlined experience within the master web-page. I’ve highlighted the icons in Darnell’s figure 3A below:

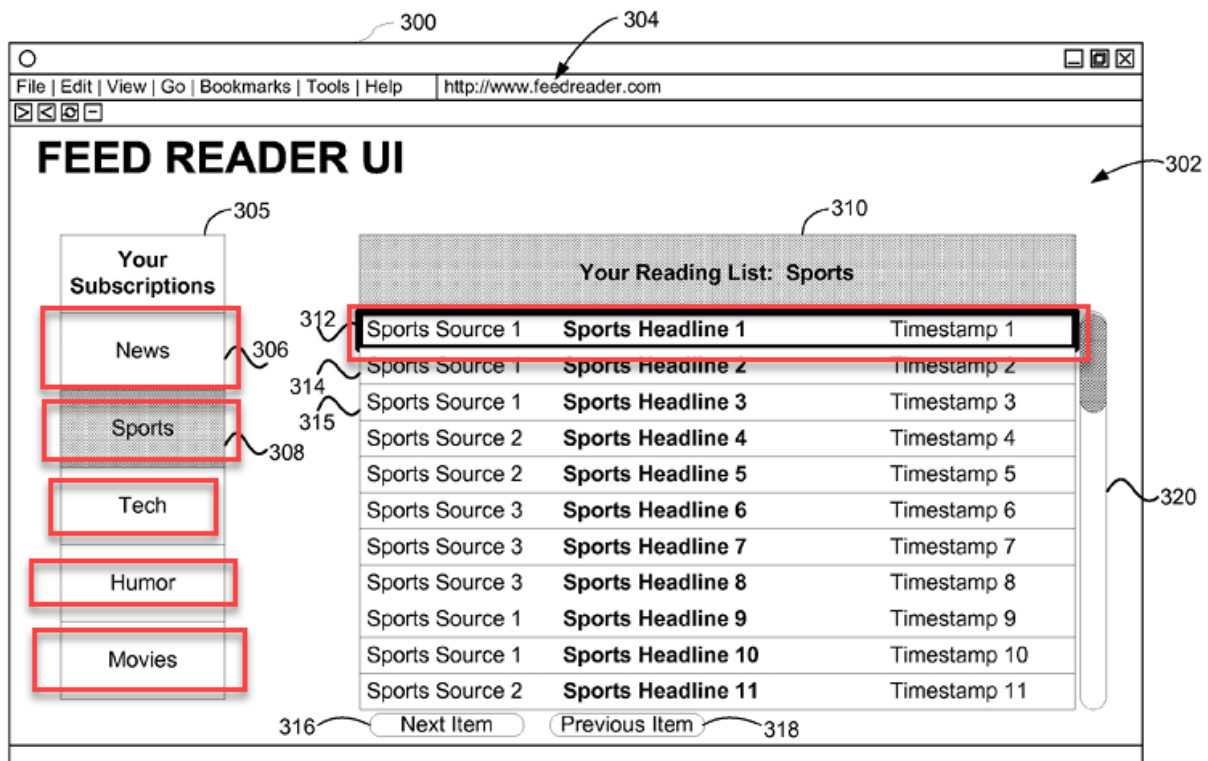


Figure 3A

Darnell, Fig. 3A.

250. Kim also teaches the use of icons in a user interface to organize and access content based on certain traits or attributes. Kim explains that “traits may be indicated by graphics, icons, or other indicia and be selected with a single user action.” Kim, ¶58. These icons allow users to group and view content items according to shared characteristics, facilitating an intuitive, icon-based organization of content selection and display. *Id.* I’ve highlighted some of the icons from Kim’s Figure 4 below:

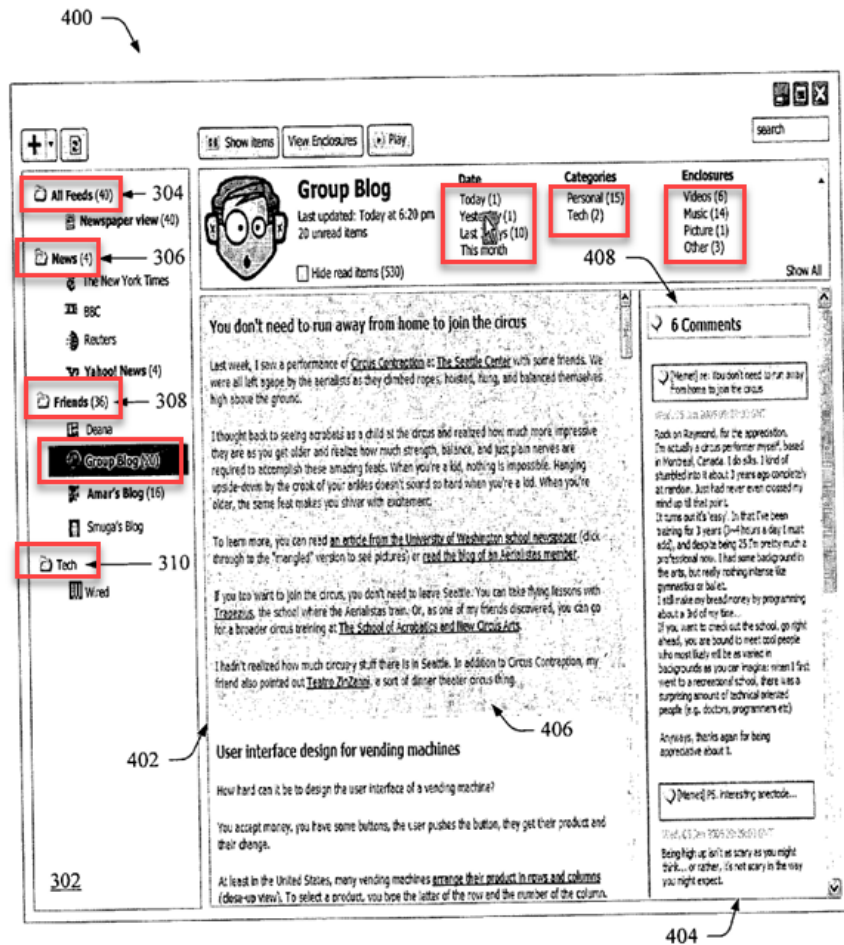


FIG. 4

Kim, Fig. 4.

251. A POSITA would have understood from the combined teachings of Darnell and Kim that an icon-based master web-page offers an organized, efficient approach for managing and accessing content feeds, consistent with the requirements of claim 10. By linking functions and content attributes with graphical icons, Darnell and Kim disclose a design that enhances usability and streamlines content management on a web-based user interface. Accordingly, the combined

Darnell-Kim system discloses or suggests the icon-based structure of the master web-page as claimed in claim 10.

B. GROUND 1B: DARNELL, KIM, AND DASGUPTA RENDER CLAIMS 6 AND 12-18 OBVIOUS

252. It is my opinion that the combination of Darnell and Kim renders claims 1-5 and 7-11 obvious. For claims 6 and 12-18, however, Dasgupta combined with Darnell and Kim meets the additional limitations of these claims. As I explain below, a POSITA would have been motivated to combine Dasgupta with Darnell and Kim. As such, it is my opinion that the Darnell-Kim-Dasgupta combination renders claims 6 and 12-18 obvious for the following reasons.

1. Motivation to Combine Dasgupta with Darnell and Kim

253. A POSITA would have been motivated to extend the Darnell-Kim combination to incorporate Dasgupta's advertising and/or virtual store teachings. A POSITA would have recognized that both Darnell's and Kim's systems for aggregating content—such as news, sports, blog entries, and email on a cloud-based server—would be compatible with Dasgupta's cloud-based media content access system because Darnell, Kim, and Dasgupta each supports content aggregation and presentation. Darnell, Abstract; Kim, Abstract; Dasgupta, Abstract. Moreover, Both Darnell and Dasgupta aggregate content at a server and send the aggregated content to a remote client. Darnell, ¶¶33-34, Fig. 1; Dasgupta, 6:52-61, 7:44-55, Fig. 2. A POSITA would have recognized the efficiency and functionality gained by

combining these teachings because Darnell and Kim's frameworks center on collecting and organizing different content types on a server before presenting them to users through a unified interface (Darnell, Abstract, ¶¶11-14, 30-32, Fig. 1; Kim, Abstract, ¶¶20-23, Fig. 1), and Dasgupta similarly describes a system where diverse media content (e.g., digital video and audio) is aggregated and streamed from a server to the client, providing users with a unified graphical user interface (GUI) for interaction. Dasgupta, 7:3-67, Fig. 2. Dasgupta's known techniques would have improved Darnell (and Kim) because they share similar architectural models and combining their features would provide a more versatile and comprehensive media content aggregation system in similar ways.

254. A POSITA would have also found these references' common reliance on Internet-based protocols, specifically for content streaming and organization, as a motivation to combine them. Darnell, for instance, uses RSS feeds and similar protocols to organize and stream digital content (Darnell, ¶¶8, 32, 63, 73, 96), while Kim describes the feed delivery of media content, which is organized and displayed to users (Kim, ¶20). Dasgupta also relies on Internet protocols to deliver content over HTTP, facilitating dynamic user interaction with various content types through the GUI. Dasgupta, 7:3-29, 10:10-24, 12:48-62. This technological overlap means that combining Darnell, Kim, and Dasgupta's systems to streamline and enhance the

management and display of content would have been an obvious and straightforward application of known media content delivery techniques over the Internet.

255. Darnell, Kim, and Dasgupta share the goal of aggregating large amounts of content and organizing it for users in a single, accessible interface. Dasgupta's system is aimed at addressing the challenge of managing extensive media content by employing a unified GUI for easy navigation. Dasgupta, 1:7-29; § XI.A.1. Darnell is aimed at solving the same problem by aggregating feeds for streamlined user access. Darnell, Abstract, ¶¶11-14, 30-32. Likewise, Kim's approach tailors the display of content based on its traits to improve usability and user interaction. Kim, Abstract, ¶¶20-23. POSITAs would have recognized that combining these references provides a cohesive and intuitive solution to the same problem: aggregating various content types (e.g., audio, video, news feeds) in a way that simplifies user interaction.

256. Darnell explains that its "content may be any machine-readable data, including but not limited to web pages, images, digital audio, digital video, Portable Document Format (PDF) documents, and so forth" (Darnell, ¶36) and Dasgupta similarly explains that "'media content'...may refer generally to any content that may be accessed and/or presented by a media content access system for experiencing by a user of the media content access system" (Dasgupta, 2:25-38). Kim also discloses receiving feeds, email, audio and video content, and visual content. Kim,

¶21. A POSITA would have thus been motivated to combine Dasgupta's advertising content, for example, with Darnell and Kim because such content falls within the ambit of Darnell and Kim and such a combination would have improved Darnell and Kim in the same way as Dasgupta.

257. Finally, a POSITA would have viewed the combination of Darnell, Kim, and Dasgupta as a straightforward application of known techniques to improve similar content aggregation systems. Combining Dasgupta's advertising framework with the structured content and feed handling capabilities of Darnell and Kim would have allowed users to interact with different types of content (RSS feeds, email, blogs, advertisements, etc.) through a unified, adaptable GUI accessible from any client device. Such a combination was well within a POSITA's understanding of conventional design.

258. As I also explained below, Applying Dasgupta's teachings to Darnell-Kim would have been nothing more than applying Dasgupta's known techniques to improve the similar content aggregation systems already taught by Darnell and Kim in the same manner. For example, combining Dasgupta's advertising framework (*see* claim 6 below) with the structured content and feed handling capabilities of Darnell and Kim would have allowed users to seamlessly interact with different types of content (RSS feeds, email, blogs, advertisements, etc.) through a unified, adaptable GUI accessible from any client device. As explained above in 1[c2], both Darnell

and Kim collect analytics and display information based on the analytics. § XI.A.2.g. A POSITA would have found it obvious to extend the information Darnell and/or Kim display based on their analytics to include advertisements, as described in Dasgupta, because advertisement data “may include, without limitation, data representative of advertisement content (e.g., advertisements images, artwork, and text) as well as advertisement metadata, which may be descriptive of one or more attributes of advertisement content.” Dasgupta, 3:63-4:8. A POSITA would have understood Darnell and Kim were ready for improvement because of their wide range of information display capabilities and that displaying advertisement content would have improved Darnell and Kim by providing more information to the user of the feed reader based on the analytics collected by the feed reader. A POSITA would have also been motivated to apply Dasgupta’s virtual store to Darnell’s feed reader UI to access content, such as other subscriptions or feeds that cost money. A POSITA would have understood that the ability to purchase content in Darnell’s feed reader UI would have extended its usefulness and streamlined operation.

259. A POSITA would have had a reasonable expectation of success in using Dasgupta’s advertisement facility or virtual store to enhance Darnell’s feed reader to allow for purchasing additional content from the content sources because Dasgupta shows such a technique in an aggregated feed display was well within the skill of a POSITA. Furthermore, providing advertisements and opportunities to

purchase goods and services in a feed reader UI, such as Darnell, was well-known in the art and would have been a straightforward application of known, conventional content aggregation methods used according to their known functions to yield predictable results. *See, e.g.*, EX1010 and EX1011.

2. Claim 6: “The cloud-based computer system of claim 5, the master application includes a purchase tool that allows the user to purchase goods or services from the multiple content sources or service providers from master web-page.”

260. Claim 6 adds a “purchase tool” to the cloud-based system described in claim 5, allowing a user to purchase goods or services from various content sources directly from the master webpage. The Darnell-Kim-Dasgupta combination discloses this purchase tool functionality, as illustrated by Dasgupta’s advertisements and “virtual store” feature and Darnell’s and Kim’s user interfaces.

261. In Dasgupta, the “virtual store” (purchase tool) enables the user to acquire products and services by selecting options accessible from within a main interface. Dasgupta explains that this virtual store is launched via prompts within the channel guide, which inform the user of specific actions needed to access purchase options. Dasgupta, 13:52-14:2 (The virtual store includes “one or more tools that may be utilized by the user to acquire (e.g., purchase, rent, subscribe to) media content such as videos, music, and games, or to acquire any other product or service offered in the virtual store.”). For example, prompt 708-1, annotated below in Figure 7, allows users to access a virtual store directly from the interface. The virtual store

includes a variety of selectable options that the user can utilize to purchase media content, such as videos, music, games, or other items. *Id.*, 16:33-44 (“a plurality of selectable options are presented to a user” that are “associated with categories of products and/or services that are available for acquisition by the user in the virtual store”). Thus, Dasgupta details the steps to purchase media content using the virtual store, such as a video, and discloses a user-friendly interface that allows for product acquisition within a unified display.

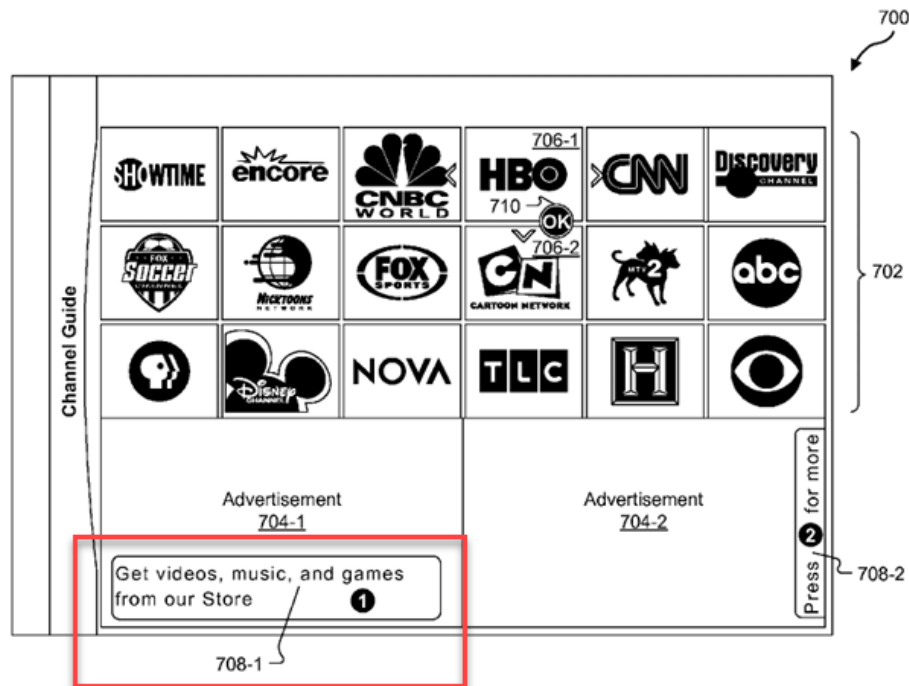


Fig. 7

Dasgupta, Fig. 7.

262. Dasgupta also details several steps for purchasing content, such as video selections, within the virtual store. *Id.*, 16:33-17:42. The store’s options are

organized by product category, and each option is linked to content sources that provide access to the desired goods or services, enabling a purchasing process from the main user interface. *Id.*, 16:33-44. As I explained in discussing claim 1[b1], the media content accessible through the virtual store includes offerings from multiple feed sources. *Id.*, 13:52-14:2, 16:33-44; Darnell, ¶¶30-32, 36; Kim, ¶¶21, 38.

263. Dasgupta's purchase tool functionality is further enhanced by embedded advertisements, which also act as access points to specific products and services. These advertisements are personalized based on the user's interests, as determined by metadata and user interaction history, and are tailored to the content displayed within the main interface. Dasgupta, 3:63-4:8, 13:9-23, 13:41-51, 14:3-30. For example, "advertisement facility 106 may analyze advertisement metadata and information about the user and select an advertisement that is likely of interest to the user based on the analysis." *Id.*, 4:9-23. Advertisements are linked to the content being provided (from the multiple content sources or service providers from master web-page). *Id.*, 5:41-56 ("content may include information, media content, advertisements, other content, services, service offers, and/or any other material specific to the media content channel and/or a media content provider associated with the media content channel"), 14:31-44 ("advertisements that are specific to a particular media content channel"). A POSITA would have understood that advertisements as displayed with prompts (a purchase tool) direct the user to

purchase what is being advertised (allows the user to purchase goods or services from the multiple content sources or service providers). In other words, a POSITA would have understood that this system of displaying advertisements as part of the main interface creates a streamlined shopping experience, allowing users to make purchases directly from prompts on the webpage that direct the user to purchase relevant goods or services.

264. In combination with Darnell and Kim, Dasgupta's advertisement and virtual store tools fulfill the requirements of claim 6 by providing a purchase tool from a unified, icon-based interface. As discussed for claim 1[b2], Darnell's feed reader UI operates as a web-based interface, accessible to users for selecting and viewing content in real time from a variety of sources. Darnell, ¶¶41, 74. This combined system not only allows for the targeted display of content based on user preferences and history but also facilitates purchasing actions directly from the interface, enhancing the utility of Darnell's feed reader UI.

265. A POSITA would have been motivated to incorporate Dasgupta's virtual store and advertisement facility into Darnell's system to improve user engagement by providing accessible purchase options directly within the content display. This would allow for the purchase of additional content or subscriptions in Darnell's feed reader UI, thus making it more comprehensive and streamlining operations for users. Applying Dasgupta's advertisement facility or virtual store as

part of Darnell's aggregated content feed system would have been a straightforward and predictable approach within a POSITA's skill level, given that using such integrated purchase tools was known in the art for content aggregation systems.

266. It was well-known in the field of content aggregation that integrating advertisements and offering direct purchasing options within a user interface (UI) could enhance user engagement and streamline access to relevant products or services. *See* EX1010 and EX1011 (U.S. Pat. Appl. Pubs. 2005/0165615 and 2004/0093327). Systems designed to provide aggregated content to users frequently incorporated targeted advertisements and purchase links, which allowed users to interact with or buy the advertised products without needing to navigate away from the primary interface. In fact, this approach can be seen across various fields, including news aggregation, online marketplaces, and media streaming services, where integrated purchasing and advertisement links were widely used at the time of the earliest priority date of the '658 patent.

267. For example, a news aggregator might include sponsored content and advertisements within its interface, where users could click on articles or product links to purchase or learn more about items relevant to the news content they're viewing. Similarly, early online shopping platforms used in the 2000s often featured a sidebar of targeted recommendations for additional products that users could purchase directly from the interface. This principle of embedding purchase options

within an interface became common, as it offered both a cohesive user experience and a predictable, streamlined pathway for monetizing content.

268. In the media content and entertainment domains specifically, services such as digital music or video platforms provided users with recommendations for content that could be purchased or rented directly from the interface, such as Apple iTunes. This was often enabled through metadata and user preferences, similar to the “virtual store” and advertisement facilities disclosed by Dasgupta. For instance, a user watching a video might be presented with options to buy or rent related media, leveraging a recommendation engine to serve targeted advertisements and purchase options directly within the interface. This method allowed users to interact with and purchase media without navigating to a separate store, aligning with Dasgupta’s approach to providing a “virtual store” accessible through prompts embedded in the content guide.

269. In Darnell, the feed reader UI consolidates content from multiple sources into a single interface, offering a natural place for integrating advertisements and purchasing options. By combining Darnell’s feed aggregation approach with Dasgupta’s method of presenting personalized advertisements and purchase prompts, a user could be shown a product, service, or subscription directly relevant to the content they’re viewing. For example, if a user frequently accessed certain content feeds, the system could present targeted offers for premium content

subscriptions or products directly tied to those feeds, allowing users to purchase them directly within the feed reader interface. This integration would improve user experience by eliminating the need to switch applications or open separate pages to complete transactions.

270. Thus, combining Darnell's feed reader UI with Dasgupta's virtual store or advertising prompts would have been a logical and predictable step for a POSITA, applying conventional aggregation and monetization techniques known in the art to achieve a more efficient, user-friendly system that both displayed and monetized content.

3. Independent Claims 12

- a. **Preamble: [claim 12]: "A method of navigating content data or content applications from multiple servers of multiple content sources or service providers through a master web-page hosted and created on a cloud-based master server from a remote computer, the method comprising:"**

271. To the extent the preamble of claim 12 is deemed limiting, Darnell discloses or suggests this limitation for the same reasons previously articulated in §§ XI.A.2.a-XI.A.2.c. Darnell describes a content feed aggregation system designed to organize and display content data or content applications sourced from multiple servers or content providers in a centralized interface accessible from a client device. Darnell discloses a method that allows a user to navigate various content items through a structured feed reader interface, providing users the ability to view,

organize, and scroll through multiple feed items. Darnell, ¶¶11, 43-49, Figs. 3A-3D, cls. 1-8. This corresponds to “navigating content data or content applications from multiple servers of multiple content sources or service providers through a master web-page” articulated in claim 12.

272. Darnell further discloses that this feed reader interface, corresponding to the claimed “master web-page,” is both created and hosted on Darnell’s server system, satisfying the requirement of a “cloud-based master server” that supports and manages aggregated content data from various sources. Content is accessible through the feed reader interface hosted on the server and becomes viewable and interactable on a client device, thereby allowing users to navigate and access content from multiple servers in real time “from a remote computer,” as claimed. Darnell, Fig. 1, ¶¶32, 35, 39.

273. Additionally, Darnell’s interface design and structure demonstrate an organized presentation of content feeds, which is consistent with the preamble’s requirement for navigating through a “master web-page hosted and created on a cloud-based master server.”

- b. Claim 12, limitation [a], “setting up an account on the cloud-based master server which streams the content data or content applications from the multiple servers of the multiple content sources or service providers simultaneously to the cloud-based master server;**

and”

274. In my opinion, Darnell discloses or suggests the limitations of claim 12[a] for at least the same or similar reasons as I articulated in §§ XI.A.2.d and XI.A.6. Darnell describes the use of a “user’s feed reader account” that serves as a mechanism for identifying the user by login credentials associated with a specific user account. Darnell, ¶99. Darnell’s feed reader UI is only accessible to “registered user[s] of the system,” indicating that users must first set up an account to engage with the content feed aggregation functions. *Id.* ¶81. Once this account is set up, Darnell’s server can authenticate the user via a “login cookie” or other token mechanisms, allowing access to the system and secure identification during content retrieval and interaction. *Id.* ¶¶92, 99-100.

275. The requirement in claim 12[a] to “set up an account on the cloud-based master server” is disclosed by Darnell’s process of user account creation and registration, which establishes user credentials that persist across sessions and allow access to the feed reader system. In Darnell, setting up this account on the server enables the simultaneous streaming of content data or applications from multiple sources directly to the cloud-based server, which is part of the content aggregation system. A POSITA would have recognized that by registering a user account, Darnell’s system can efficiently stream selected content to each unique user feed

interface in real time, organized according to individual account settings and preferences.

276. Thus, the setup of an account in Darnell is necessary to provide the intended cloud-based aggregation of multiple content feeds and personalized content viewing. This account structure further allows for customized user interactions such as tagging and labeling of content items, which are only possible within an authenticated account. In the context of claim 12[a], this account setup feature effectively enables and manages the delivery of real-time, multi-source content data and applications as claimed.

- c. **Claim 12, limitation [b1], “constructing the master web-page on the cloud-based master server from the remote computer via a tool box software stored on memory of the cloud-based master server and executed on a processor of the cloud-based master server and that is accessible from the remote computer,”**

277. In my opinion, Darnell discloses or suggests the limitations of claim 12[b] for at least the same or similar reasons as I articulated in §§ XI.A.2.e-XI.A.2.f. This limitation is similar to limitations 1[b3]-1[c1], with the addition that it specifies the “construction of the master web-page on the cloud-based master server” being achieved “via a tool box software.”

278. In Darnell, server system 104, including the feed reader user interface 116, is expressly described as providing a “web-based content aggregation service”

to users. Darnell, ¶41. A POSITA would have understood that this language indicates Darnell's feed reader user interface (the master web-page) is created and maintained on server system 104, which is the "cloud-based master server." The fact that the webpage is constructed and then transmitted to the client device for viewing confirms that it is both "constructed" and "hosted" on the server. *Id.*, ¶¶35, 39-41, 112, Figs. 3A-3D.

279. Moreover, as explained in the analysis of limitation 1[c1], Darnell's system includes embedded scripts (e.g., JavaScript) that operate client-side and interact with the server-side components to create and display user-specific web content based on interactions and preferences set by the user. A POSITA would have recognized these embedded scripts as part of the toolbox software responsible for the dynamic aspects of webpage construction, enabling customization of how content is organized and displayed, depending on user interactions with the feed reader interface. *Id.*, ¶41.

280. Darnell's configuration, where the feed reader UI and associated scripts facilitate user-driven customization of the master web-page's appearance and content, describes a "tool box software" operating from the cloud-based server to render the master web-page viewable on a client device. This approach is further supported by Darnell's disclosure that the interface includes "tools" allowing users

to add, remove, and label content feeds—functionality that operates as a web-based tool kit, adapting the display of content based on user input. *Id.*, ¶¶40, 60.

281. Therefore, Darnell’s system discloses the claimed “construction of the master web-page” on a cloud-based server with an accessible tool kit that operates via software stored and executed on that server, as claimed in 12[b].

- d. **Claim 12, limitation [b2], “wherein a user navigates through the content data or content applications from the multiple content sources or service providers through master web-page in real time from the remote computer using a slide tool program executed from the remote computer,”**

282. Darnell, Kim, and Dasgupta, alone or in combination, disclose limitation 12[c] for the same or substantially similar reasons as I articulated for claim elements 1[c1] and 1[c2] in §§ XI.A.2.f-XI.A.2.g.

- e. **Claim 12, limitation [b3], “wherein the tool box software allows the user to write code, select code or delete code to enable and disable tool features that control how sub-sets of the content data or the content applications from the multiple content sources or service providers are received, displayed or organized;”**

283. Darnell, Kim, and Dasgupta, alone or in combination, disclose limitation 12[d] for the same or substantially similar reasons as I articulated for claim element 1[c1] in § XI.A.2.f.

- f. **Claim 12, limitation [c], “simultaneously searching analytics based which tool features are enabled to target specific products, services or information to the**

user; and”

284. Kim discloses limitation 12[e] for the same or substantially similar reasons as I articulated for claim element 1[c2] in § XI.A.2.g. The one difference is this limitation requires “simultaneously searching analytics.” A POSITA would have understood that Darnell-Kim’s server would have read (searching) the user preferences and/or usage history (analytics) in order to use them to display the specific content (target specific products, services, or information to the user) based on which tool features were enabled.

g. Claim 12, limitation [d], “purchasing the specific products, services or information from at least one of the multiple content sources or service providers from the master web-page.”

285. Dasgupta discloses limitation 12[f] for the same or substantially similar reasons as I articulated for dependent claim 6 in § XI.B.2.

4. Dependent Claim 13: “The method of claim 12, wherein setting up the account includes one or more of enabling security features and selecting personal settings.”

286. Darnell discloses claim 13 for the same or substantially similar reasons as I articulated for dependent claim 4 in § XI.A.6.

5. Dependent Claim 14: “The method of claim 12, wherein one of the multiple content sources or service providers is an e-mail [sic] content source or service provider.”

287. Kim discloses dependent claim 14 for the same or substantially similar reasons as I articulated for dependent claims 3 and 11 in § XI.A.5.

288. This claim appears to have a typographical error: “c-mail.” A POSITA would have understood this to mean “e-mail,” since no mention of “c-mail” is in the specification and the term “c-mail” is otherwise undefined and not known in the technological art area of content aggregation systems.

6. Dependent Claims 15: “The method of claim 12, wherein the user interface is an icon-base web-page.”

289. Darnell and Kim, alone or in combination, disclose dependent claim 15 for the same or substantially similar reasons as I articulated for dependent claim 10 in § XI.A.9.

7. Dependent Claim 16: “The method of claim 15, further comprising logging the cloud-based master server from a remote computer, wherein the cloud-based master server initiates a down load of software on to the remote computer that enables the slide tool program.”

290. Darnell discloses dependent claim 16 for the same or substantially similar reasons as I articulated for claim limitation 1[b3] and dependent claim 4 in §§ XI.A.2.e and XI.A.6.

291. In my opinion, Darnell discloses or suggests the additional limitations of claim 16. As described in the analysis of limitation 1[b3] and claim 4, Darnell’s feed reader interface (UI) includes a scrollbar 320 that is a “slide tool program” to enable scrolling or paging through content on the client system. Darnell discloses that the feed reader UI is web-based, meaning that the feed reader UI—including the scrollbar—is provided to the client system as a webpage accessed from server

system 104 (cloud-based master server). Darnell, ¶41. A POSITA would have understood that upon logging into the feed reader (see claim 4) the webpage containing the feed reader UI would be sent (down load of software) to a client system from the server system, the scrollbar feature (slide tool program) would be initiated on the client system (on to the remote computer).

292. Upon accessing the feed reader UI, a user is required to log in through Darnell's authentication system. This log-in feature uses mechanisms such as a login cookie and a login webpage displayed on the client computer, verifying the user's identity as an authorized user before allowing access to the feed reader. *Id.*, ¶¶91–102. Once the user's identity is authenticated, Darnell's server initiates the download and rendering of the feed reader UI webpage, which includes embedded JavaScript and other scripts that execute on the client device to activate and enable features such as the scrollbar. *Id.*, ¶¶41, 67.

293. Darnell discloses that upon logging into the cloud-based server, the master server initiates a download of necessary software elements onto the remote client device, enabling the scrollbar (slide tool program) on the client device. A POSITA would have understood that this process allows the server system to effectively provide the interactive elements of the feed reader UI directly to the remote device, where they are then executed. This approach is consistent with claim 16's requirement that logging into the cloud-based master server from a remote

computer initiates a download of software that enables the slide tool program. The claim language, “initiates a download of software on to the remote computer that enables the slide tool program,” is thus met by Darnell’s disclosure of scripts, such as JavaScript, which are executed on the client side and enable scrolling functions once downloaded from the server system.

294. Further, the execution of Darnell’s slide tool program on the client device would have been understood by a POSITA as initiated by the cloud-based server, since the server controls access to the feed reader UI and directs its rendering on the client side through the embedded scripts. This structure is common in web-based applications where a server provides functionality that is then enabled and controlled locally on a user’s remote device, yielding predictable and efficient results through known methods of content aggregation and web interface implementation.

8. Dependent Claim 17: “The method of claim 16, further comprising selectively viewing the content data or content applications from the multiple content sources or service providers by scrolling or paging through the content data or content applications using the slide tool program.”

295. Darnell and Kim, alone or in combination, disclose dependent claim 17 for the same or substantially similar reasons as I articulated for claim limitation 1[b3] in § XI.A.2.e.

9. Dependent Claim 18: “The method of claim 12, further comprising storing the analytics in the memory of the cloud-

based master server.”

296. Kim discloses dependent claim 18 for the same or substantially similar reasons as I articulated for claim limitation 1[c2] and dependent claims 5 and 8 in §§ XI.A.2.g and XI.A.7.

C. GROUND 2A: DASGUPTA AND DARNELL RENDER CLAIMS 1 & 4-10 OBVIOUS

297. In Ground 1, I applied Darnell as the primary reference. Here, in Ground 2, I will apply Dasgupta as the primary reference. In my opinion, Dasgupta in view of Darnell renders obvious claims 1 and 4-10 for the reasons that follow.

1. Motivation to Combine Dasgupta and Darnell

298. In this section I expand on my reasons for combining Dasgupta and Darnell, explained in § XI.B.1.

299. A POSITA would have recognized that integrating these systems provides considerable advantages. Specifically, Darnell’s methods for managing feed subscriptions and authenticating users complements and enhances Dasgupta’s existing media access capabilities. Darnell discloses a system that enables users to select specific content feeds, securely access them, and personalize their content viewing experience according to their preferences, as reflected in Darnell’s subscription and feed reader structure. This structured selection of content allows users to focus on chosen sources while preventing unauthorized access to sensitive or private information. Darnell, ¶¶40, 84, 99-102. Integrating these capabilities into

Dasgupta's system would provide users with a more organized, user-focused interface while leveraging Darnell's robust selection and security functionalities to ensure that only authenticated, authorized users can view selected content.

300. Dasgupta's media content system, in turn, is designed to provide extensive access to a wide range of media channels and sources, and explicitly supports customization for the presentation and control of media content access using "any technologies suitable for accessing media content." Dasgupta, 3:14-28. By adopting Darnell's subscription techniques, Dasgupta's system would benefit from displaying content specifically tailored to each user's subscriptions and preferences, rather than presenting the entire breadth of available content without distinction. A POSITA would have recognized that Dasgupta's "access facility" could readily accommodate Darnell's selection and security features, adding value by offering a more personalized media experience that aligns with individual user profiles.

301. Furthermore, combining Darnell's feed selection and authorization methods with Dasgupta's system would have been well within the knowledge of a POSITA. Systems that organize and restrict access based on user authentication and predefined subscriptions were well-understood in the field at the time. Therefore, the combination would be a straightforward application of these known methods to yield predictable results, offering users a streamlined and customizable interface.

Incorporating Darnell's methods would merely apply recognized techniques for enhancing control over content access and personalization to the content management features in Dasgupta, resulting in a unified and secure media access system without introducing undue complexity.

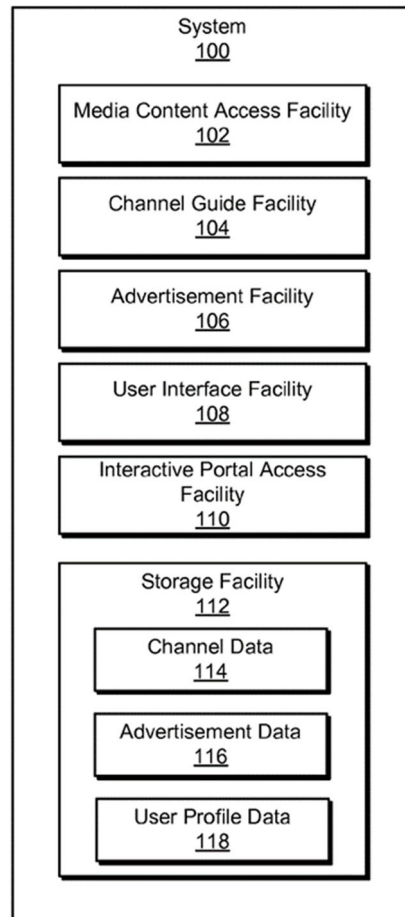
2. Independent Claim 1

a. Preamble of claim 1, "A cloud-based computer system comprising:"

302. To the extent it is limiting, the preamble of Claim 1, "a cloud-based computer system," is disclosed or suggested by Dasgupta's media content access system, which comprises various components for accessing, delivering, and presenting media content from multiple sources through network connections such as the internet or intranet. A POSITA would have recognized that Dasgupta's media content access system is a cloud-based computer system, as required by the claim.

303. Dasgupta's disclosure describes a comprehensive media content access system configured for retrieving and delivering media content, supported by both software and hardware elements. Dasgupta explains that system 100 includes components like media content access facility 102, channel guide facility 104, advertisement facility 106, user interface facility 108, interactive portal access facility 110, and storage facility 112. Dasgupta, 3:4-13. Each of these facilities contributes to the system's overall function of providing and managing content accessible through a centralized networked environment, qualities intrinsic to a

cloud-based system. Figure 1 of Dasgupta effectively illustrates this architecture, emphasizing the interactive and content-rich nature of the system, similar to what is expected from cloud-based implementations.



Dasgupta, Fig. 1.

304. Moreover, the system's configuration allows media content access facility 102 to retrieve and aggregate content from various providers, making it accessible to end-users through Dasgupta's user interface facility 108. This architecture is facilitated by components like the media content delivery subsystem 202 and media content access subsystem 204, which communicate over network

206, enabling delivery and access across a variety of network types, including the internet or intranets. *Id.*, 7:3-19. This design, involving a “network cloud” as depicted in Figure 2, is a cloud-based computer system, where content is managed, processed, and transmitted through cloud-network infrastructures.

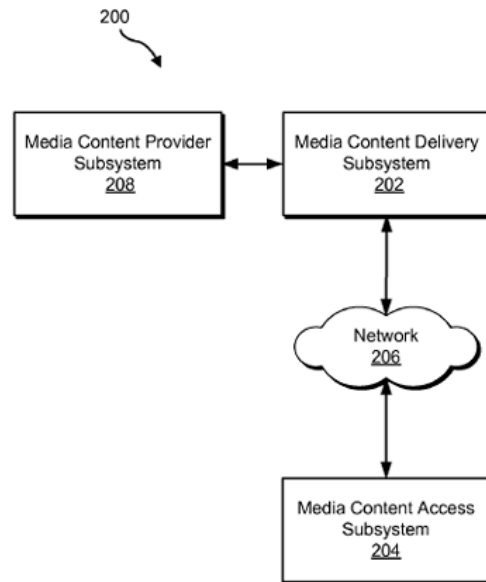


Fig. 2

Id., Fig. 2.

305. The access facility “process[es] and present[s] media content for experiencing by a user,” whereas the channel guide “maintain[s] channel data representative of one or more media content” accessible by the access facility and

stored as channel data in the storage facility. *Id.*, 3:14-51. The UI is “configured to generate and launch one or more GUIs for display to a user.” *Id.*, 4:33-44.

306. Dasgupta further emphasizes that the system’s software is responsible for generating and providing GUIs (graphical user interfaces) to users, including end-user devices, for interacting with media content. *Id.*, 2:64-3:3. The ability of Dasgupta’s media content access system to generate and distribute GUIs demonstrates its role as a cloud-based environment, as it supports centralized content delivery, user interaction, and control across remotely connected devices. This configuration discloses the requirements of a cloud-based computer system as specified in Claim 1.

307. A POSITA would therefore have understood that Dasgupta’s media content access system qualifies as a “cloud-based computer system” as described in the preamble of Claim 1.

b. Claim 1, limitation [a], “master server with a processor and memory, wherein the master server accessible over the internet/Intranet from remote computer devices by a user”

308. In my opinion, Dasgupta discloses or suggests this limitation 1[a]. As detailed in Dasgupta, the media content access system operates through one or more computing devices, which include “a processor (e.g., a microprocessor)...from a computer-readable medium (e.g., a memory, etc.).” Dasgupta, 6:27-61, 12:15-39.

This description corresponds to a master server containing both a processor and memory, as claimed.

309. Dasgupta's delivery subsystem 202 includes a processor 304 and storage device 306, such as memory or storage drives. Figure 3 of Dasgupta illustrates this delivery subsystem, which is designed to manage operations for the media content access system. Subsystem 202 performs functions, including data processing and storage, which, in my opinion, meet the claimed "master server with a processor and memory." Dasgupta, 7:56-8:67. I've outlined the relevant components of subsystem 202 below:

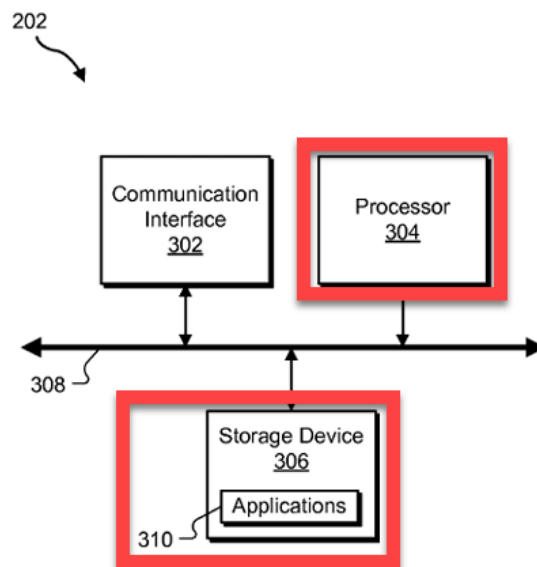


Fig. 3

Id., Fig. 3.

310. Further, Dasgupta explains that any or all of the facilities described may be implemented within delivery subsystem 202, which encompasses functionalities such as media content access, channel guide management, and advertisement handling. Dasgupta, 8:54-67.

311. Additionally, Dasgupta discloses a second subsystem, subsystem 204, which contains a processor 404 and storage device 406. This subsystem may be implemented on remote computing devices, including “a personal computer,” and facilitates user access to content. Dasgupta, 9:1-18, Fig. 4. Dasgupta further explains that subsystem 204 receives content from subsystem 202 over network 206, which includes internet and intranet connectivity options. Dasgupta, 7:3-43. In my view, this arrangement establishes subsystem 202 as the “master server” that is accessible from remote devices (e.g., subsystem 204) via the internet or intranet, thereby fulfilling the requirements of Claim 1[a].

312. Based on the description provided in Dasgupta, a POSITA would have understood delivery subsystem 202, equipped with processing and memory capabilities and communicating over network 206, to perform as the claimed “master server with a processor and memory, accessible over the internet/intranet from remote computer devices by a user.”

- c. **Claim 1, limitation [b1], “a master application stored in the memory of the master server and that executes on the processor of the master server and that includes software coded for organizing content data**

**and content applications from multiple content
sources or service providers pre-selected by the user”**

313. In my opinion, Dasgupta discloses or suggests this limitation 1[b1]. Dasgupta describes “one or more applications 310” stored in storage device 306 within subsystem 202, which contains processor 304. As shown in Dasgupta’s Figure 3, applications 310 are housed in memory on the master server and are “configured to direct processor 304 to perform any of the operations described herein,” including functions associated with the access, guide, and advertisement facilities. Dasgupta, 8:27-67. These applications direct processor 304 to perform tasks integral to the system, establishing that Dasgupta’s system includes a “master application” capable of organizing content in a manner consistent with the claim requirements. I annotated Figure 3 below to show Applications 310 (the master application):

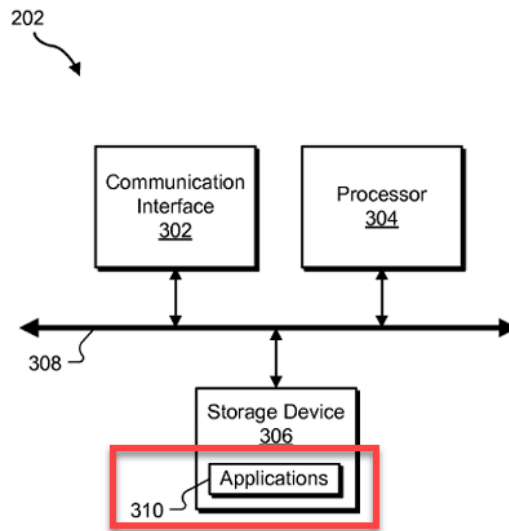


Fig. 3

Id., Fig. 3.

314. “[A]pplications 310 [are] configured to direct processor 304 to perform any of the operations described herein” and “may be configured to direct processor 304 to perform one or more processes or functions associated with” the facilities as described in 1[pre] (executes on the processor of the master server). Dasgupta, 7:39-67. Dasgupta also explains that channel guide facility 104, advertisement facility 106, and user interface facility 108 interact to arrange and present content to users, particularly through a graphical user interface (GUI) displaying a channel guide with icons representing various media content channels. Dasgupta, 1:64-2:24, 4:33-44, 13:9-23, 17:43-63. In addition to representing channel guide and advertisements in

this UI, Dasgupta describes how this interface serves as a centralized platform for accessing content. These icons are configured as launching points to external interactive portals, providing users access to diverse content sources directly within the system. Dasgupta, 2:17-24, 6:1-6, 13:24-40, 18:29-38. The process of assembling channel icons and advertisements into a unified, organized structure meets the claim's requirement for "organizing content data and content applications from multiple content sources or service providers."

315. Dasgupta's storage facility 112 within subsystem 202 holds channel and advertisement data, which organizes and manages content within the GUI. Dasgupta, 3:63-4:8, 6:7-12, 8:54-67, 13:41-14:44. In my opinion, Dasgupta's channel icons, serving as representations of multiple media content sources and stored in subsystem 202, would have been understood by a POSITA to represent "content data and content applications," providing the building blocks for the organized, user-friendly UI display specified by the claim.

316. Dasgupta's Figures 7 and 16 provide visual examples of the organized display of a channel guide featuring "a plurality of channel icons associated with a plurality of media content channels" originating from multiple content providers, such as the inclusion of "HBO." Dasgupta, 17:43-63, Figs. 7 & 16.

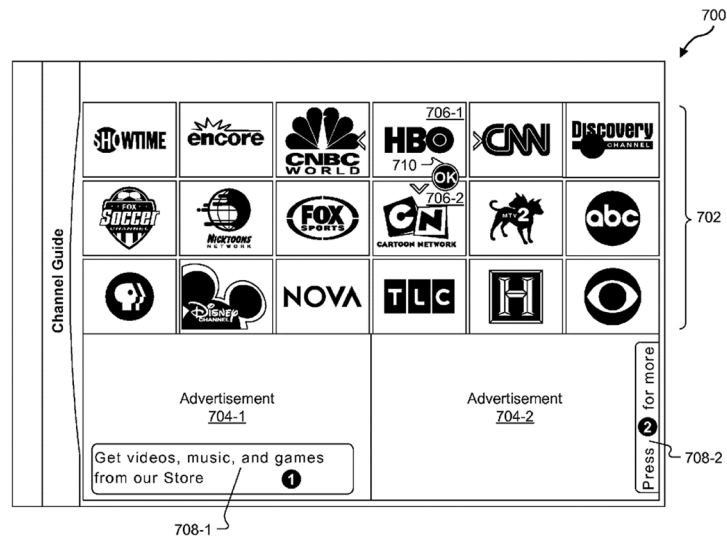


Fig. 7

Id., Fig. 7.



Fig. 16

Id., Fig. 16.

317. Regarding the phrase “preselected by the user,” Dasgupta explains that channel guides contain content accessible only by “subscribers,” suggesting user

selection. In describing a user as a “subscriber to one or more services,” Dasgupta discloses that such services, including premium channels like HBO, require a user-initiated selection and purchase. Dasgupta, 2:64-3:3, 7:3-19, 14:3-30. A POSITA would have understood that a user must actively subscribe to certain channels, which are then displayed in the channel guide, meeting the requirement that these content sources are “preselected by the user.” It was well-known that HBO is a subscription channel that can only be viewed if it is chosen and paid for by the user, such as in a package that contains the channel. A POSITA would have also understood that the content from a subscription channel, such as HBO, would be purchased by the user before it could be presented to the user in the UI.

318. Additionally, Dasgupta specifies that the UI system can “populate a GUI with content,” modify it by “adding new content,” and subsequently organize content within the GUI according to user selections and interactions. Dasgupta, 3:14-28, 5:6-40. Dasgupta discloses that the access facility “may be configured to access, process, and present media content for experiencing by a user,” using “any technologies suitable for accessing media content.” *Id.*, 3:14-28. Dasgupta’s UI may also be “configured to control the contents of a GUI.” *Id.*, 5:6-19. For example, it “may initially populate a GUI with content such as a graphical representation of a channel guide” and “[s]ubsequently...may modify the contents of the GUI such as by adding new content to the GUI, removing content from the GUI, modifying

content in the GUI, and replacing content in the GUI with new content.” *Id.* Furthermore, the UI accepts user input to control the GUI. *Id.*, 5:20-40. A POSITA would have understood that a user is operating the UI and selecting the content to be received.

319. Dasgupta further clarifies how the system allows users to navigate through the UI in both “navigation mode” and “launch mode,” where users select specific media content channels such as HBO. For example, as the user navigates and highlights a channel icon, content relevant to that icon is displayed and updated, allowing for preselection and engagement with the chosen content. Dasgupta, 14:45-16:17, Figs. 7-9.

320. To the extent PO argues that “preselected by the user” would not be obvious in view of Dasgupta alone, POSITAs would have been motivated to incorporate Darnell’s feed subscription methods into Dasgupta’s access facility to enhance user interaction and organization of media sources. In Darnell, the system presents subscribed feeds through the user interface, filtering content according to a list created by the user. By applying Darnell’s feed selection and management to Dasgupta’s system, the user could preselect and organize specific content sources within the UI, enhancing the user experience. Dasgupta, 3:14-28; Darnell, ¶¶40, 87-90.

321. In my opinion, incorporating Darnell's method would be a predictable application of known techniques to improve the organization and customization of content in Dasgupta's media content access system. This enhancement would provide the expected results of displaying only the preselected content sources based on user subscription, while maintaining a straightforward application of conventional aggregation methods.

322. In my opinion, Dasgupta alone discloses this limitation or Dasgupta in combination with Darnell discloses this limitation.

- d. **Claim 1, limitation [b2], "and streamed to the master server simultaneously in real time from their respective servers on a master web-page and are viewable from the remote computer devices through the master server on the master web-page"**

323. In my opinion, Dasgupta discloses or suggests this limitation 1[b2]. Dasgupta describes how provider subsystem 208 transmits media content to delivery subsystem 202, which is then distributed to media content access subsystem 204 via network 206. Dasgupta, 7:3-19, Fig. 2. Provider subsystem 208 includes sources operated by media providers, such as HBO, configured to send media data to subsystem 202, thereby establishing a direct streaming of content from these sources to the master server. *Id.*, 7:3-19.

324. In the context of cable systems, POSITAs would have understood that content transmitted over such systems relies on channels organized by distinct

frequency bands. Wideband signals carry numerous channels simultaneously, each differentiated by its assigned frequency. A receiver device “tunes” to a specified frequency to isolate and present the content associated with that frequency, such as specific channels on cable systems. Dasgupta describes a similar arrangement where media channels are defined by frequency bands, with subsystem 202 configured to access these “channels” based on frequency availability. Dasgupta, 2:25-50, 3:14-28, 9:25-10:24, 12:40-13:4. Given that each frequency band corresponds to a separate content stream available concurrently across all channels, a POSITA would have understood that content could be delivered “simultaneously in real time” from these respective sources to the master server.

325. Dasgupta also discloses that subsystem 202, in conjunction with network 206, delivers content from multiple providers to subsystem 204, which operates as the remote client device. To facilitate this communication, Dasgupta explains the use of the Hypertext Transfer Protocol (HTTP) within the Transmission Control Protocol/Internet Protocol (TCP/IP) framework to manage data transmission. Dasgupta, 7:20-29. HTTP is the foundational protocol for the World Wide Web, defining how data, including web pages, is transmitted from servers to client devices. Using HTTP, subsystem 202 can provide web-based content aggregation directly to subsystem 204 by sending structured data (such as HTML or other markup data) that is rendered on the client device. HTTP governs the

interaction between web servers and web clients, such as browsers, allowing users to access and interact with content over the internet, which is particularly relevant when web-based interfaces serve as content aggregation platforms. *See, e.g.,* Darnell, ¶¶41, 96, 102. A POSITA would have thus recognized that subsystem 202, acting as the master server, organizes and distributes content as web pages viewable on subsystem 204 over HTTP, creating a “master web-page” accessible on remote devices through the master server. At the time of the alleged invention, it was well known that webpages included HTML and JavaScript programs to create user interfaces. A POSITA would have understood that subsystem 202 sends webpages to subsystem 204 using HTTP (i.e., “on a master web-page”).

326. Additionally, Dasgupta explicitly describes how media content becomes viewable on subsystem 204 (the remote computer device), which “present the media content for experiencing (e.g., viewing) by a user.” Dasgupta, 7:36-43. Subsystem 202 serves as a central distribution point for content received from provider subsystem 208, thereby ensuring that the aggregated media content is accessible for viewing on subsystem 204. In my opinion, a POSITA would have thus understood Dasgupta’s system as streaming content in real time to the master server, accessible from a master web-page and ultimately viewable on remote devices through that server-based interface. This understanding is reinforced by Dasgupta’s disclosure of subsystem 202 as a central server facilitating the distribution and

display of media content on user-accessible interfaces, hosted and processed centrally and streamed to client devices for remote viewing.

- e. **Claim 1, limitation [b3], “wherein the master application programs a processor on the remote computer devices to operate and execute a slide tool program for scrolling or paging through the content data or the content applications from the multiple content sources or service providers simultaneously and through the master server in real time from the master web-page and from the remote computer by operating the slide tool program from the remote computer; and”**

327. In my opinion, Dasgupta discloses or suggests this limitation 1[b3]. As explained above, Dasgupta provides a method in which user interfaces, such as a channel guide or advertisement display, are transmitted from subsystem 202 to subsystem 204 using HTTP and displayed on a user’s client device. Dasgupta, 7:20-29. The ’658 patent discloses that a “slide tool” scrolls or pages through content data and does not limit the form of the slide tool as long as it allows scrolling or paging. ’658 patent, 4:24-26. A POSITA would have understood that a “slide tool program” could include any software program that enables scrolling, whether a scroll bar, arrow key navigation, or similar visual tool in the user interface. This understanding is further supported by the use of JavaScript or HTML to execute and present user interfaces with scroll functionality on client devices as disclosed in Dasgupta. A POSITA would have thus found it obvious that Dasgupta’s user interface system would enable scrolling or paging through content data in a client

device interface using software-based tools like scroll bars or similar prompts. Dasgupta, 8:39-67.

328. Dasgupta describes subsystem 202 as hosting the user interface in a way that provides client devices (subsystem 204) with a programmatic user interface. For example, Dasgupta discloses applications 310, stored in storage device 306 of subsystem 202, that configure processor 304 to execute various functionalities, including the graphical user interface (GUI) transmitted to the client computer. Dasgupta, 8:27-67, Fig. 3. By delivering the user interface over HTTP, subsystem 202 “programs” the remote device, allowing the remote processor to display and interact with the interface locally, effectively executing the channel guide or advertisement display on subsystem 204. Dasgupta, 7:20-29.

329. Dasgupta specifically enables scrolling functionality through its channel guide and advertisement interface. For instance, Dasgupta’s advertisement facility 106 enables users to scroll horizontally through advertisements within the interface. A visual prompt 708-2, as illustrated in Figure 7, allows users to “access more advertisements 704” by initiating scrolling. Dasgupta, 14:3-30, Fig. 7. When activated, the prompt scrolls a strip of advertisements “such that one or more of the currently displayed advertisements 704 is moved off-screen... and one or more other advertisements 704 is moved on-screen.” *Id.* This disclosure meets the slide tool program requirement, as it allows a user to view advertisements sequentially by

scrolling within the user interface on the client device. A POSITA would have recognized this functionality as a form of scrolling or paging through content data or applications, since the advertisements are accessed and navigated in real time through the master server interface and viewed on the remote computer device. I annotated Figure 7 below to show the scrolling advertisements in Dasgupta's user interface:

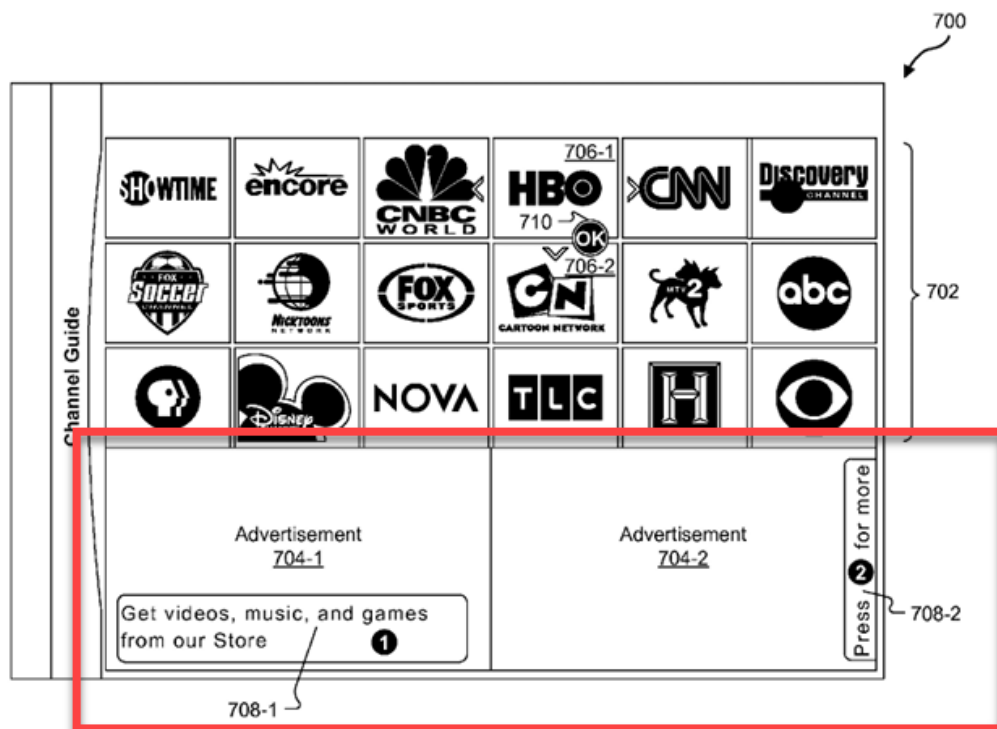


Fig. 7

Id., Fig. 7.

330. In another embodiment, Dasgupta discloses a channel guide 1602, shown in Figure 16. In this configuration, Dasgupta enables a three-dimensional grid of media channels that users can scroll through directionally. When a user initiates

a navigation event, Dasgupta describes how the system “respond[s] by directionally scrolling the three-dimensional grid matrix of channel icons relative to stage 1604.” Dasgupta, 18:49-19:3. This channel guide thus provides a similar scrolling experience, allowing users to navigate channel options using the remote interface and view available media content channels from multiple providers in real time. This scrolling action on the remote device interface demonstrates a “slide tool” within the meaning of the ’658 patent, as it allows the user to navigate and select content from multiple providers directly from the master server interface.



Fig. 16

Id., Fig. 16.

331. As outlined above, Dasgupta teaches that such scrolling or paging features may occur on the remote device (subsystem 204) and are executed via programmatic instructions delivered from the master server (subsystem 202) over

HTTP. In my opinion, A POSITA would have understood that these features in Dasgupta disclose or suggest a “slide tool program” that enables scrolling through content sourced from multiple providers in real time via a server-managed interface, aligning directly with the requirements of limitation 1[b3].

- f. **Claim 1, limitation [c1], “a remote tool box feature application that allows the user to write code, select code or delete code to enable and disable tool features that control how sub-sets of the content data or the content applications from the multiple content sources or service providers are received, displayed or organized,”**

332. In my opinion, Dasgupta, in combination with Darnell, discloses or suggests limitation 1[c1]. Dasgupta describes the ability of the user interface to enable navigation within a channel guide in “navigation mode,” where a user can browse through channel icons, selecting and highlighting various media content such as “HBO” or “CNN.” Dasgupta, 14:45-15:44. The channel icons represent the media content channels in Dasgupta’s system (as discussed for limitation 1[b1]) and would have been understood by a POSITA as corresponding to the content data and applications provided by multiple content sources or service providers. A POSITA would have therefore recognized that this ability to navigate and highlight specific media content meets the “remote tool box feature application” as disclosed in the ’658 patent, which defines the tool box as an application that “allow[s] a web-

page layout and/or operation to be customized by linking tools with content applications.” ’658 patent, 4:16-20.

333. Further, a POSITA would have understood that when a user interacts with the channel guide and highlights a channel icon, additional content specific to the selected icon is displayed on-screen, thus modifying the display of content in real time. In “navigation mode,” Dasgupta discloses that when a channel icon is highlighted, channel content 902 is displayed, “inform[ing] a user of a specific input that may be provided to initiate access to information about one or more deals associated with the channel content.” Dasgupta, 15:45-16:16, Fig. 9. By highlighting a particular channel icon, the system triggers (or “selects”) additional code or program instructions that enable the display of specific content, thereby activating additional functionalities within the UI. A POSITA would have recognized that in selecting a particular content icon and displaying related content, Dasgupta’s system effectively “selects code” to enable specific display features, in line with the requirement for selecting code “to enable and disable tool features that control how sub-sets of the content data or the content applications...are...displayed.”

334. Furthermore, Dasgupta’s system includes an “advertisement facility 106,” configured to personalize and control the display of advertisements within the UI. This facility “analyzes advertisement metadata and information about the user” and “select[s] an advertisement that is likely of interest to the user based on the

analysis.” Dasgupta, 4:9-23. A POSITA would have understood this feature to function as a “tool” that determines which specific advertisements are displayed based on the user’s data and preferences, dynamically adjusting the advertisement content available to each user. This advertisement facility thus embodies a tool feature that manages which “sub-sets of the content data or the content applications...are received, displayed or organized” on the user interface, as it controls both the selection and organization of advertisements delivered to the user.

335. Accordingly, it is my opinion that a POSITA would have recognized Dasgupta, in combination with Darnell, as disclosing a “remote tool box feature application” that allows user interaction to selectively enable or disable tool features affecting the receipt, display, and organization of various content sub-sets, consistent with the requirements of limitation 1[c1].

g. Claim 1, limitation [c2], “wherein the master server collects analytics based which tool features are enabled to target specific products, services or information to the user.”

336. In my opinion, Dasgupta discloses or suggests limitation 1[c2]. Dasgupta describes an “advertisement facility 106,” which dynamically selects and displays advertisements to users by considering user-specific data. Dasgupta explains that the advertisement facility analyzes both “advertisement metadata and information about the user” to “select an advertisement that is likely of interest to the user based on the analysis.” Dasgupta, 4:9-23. This disclosure suggests that the

advertisement facility targets specific products, services, or information to users based on data about their individual preferences or behavior, aligning with the requirement in the claim to “target specific products, services or information to the user.” By using this approach, the system customizes the presentation of advertisements, selecting content tailored to the user’s preferences and data profile. A POSITA would thus have understood that the advertisement facility acts to target specific products and information by analyzing user-related data within the system.

337. Dasgupta also discloses “user profile data 118” stored in the system’s storage facility 112, which includes data relevant to customizing the user experience, such as “one or more preferences of a user, historical media content access data associated with the user (e.g., data indicating one or more media channels and/or media content programs accessed by the user), and any other information about the user that may be helpful for selecting a personalized advertisement for presentation to the user.” Dasgupta, 4:9-23. This user profile data is maintained within the storage device 306 on subsystem 202, which a POSITA would have recognized as the “master server” described in claim limitation 1[c2]. By collecting and storing this user profile data, the master server gathers “analytics” regarding user preferences and historical interactions with media content, which allows the advertisement facility to make informed decisions on which products or services to target based on these analytics. Dasgupta, 8:54-67. A POSITA would have understood that when a

user selects a channel in Dasgupta's UI, particular code would be selected and enabled to, for example, "initiate access to information about one or more deals associated with the channel content." Dasgupta, 15:58-65. A POSITA would have further understood that when a user selects channels (i.e., "tool features are enabled"), the selections are stored in the user's profile data. Dasgupta, 4:9-23 ("historical media content access"). That profile data (e.g., showing the television channels to which the user subscribed) allows Dasgupta to provide targeted advertisement data to that user (target specific products, services or information to the user).

338. Furthermore, the '658 patent defines a "tool feature" as a function that can initiate software sequences to display content or control the display of data on the UI. '658 patent, 4:11-15. The analytics gathered within Dasgupta's advertisement facility utilize user profile data—such as historical access data—to make decisions about advertisement selection and targeting, which a POSITA would have recognized as an enabled "tool feature" that directs specific content to the user based on the collected analytics. Consequently, the system in Dasgupta both gathers and utilizes analytics based on user-interaction data, stored and processed on the master server, to effectively target advertisements to users according to their preferences and historical actions within the platform.

339. Thus, it is my opinion that Dasgupta, in view of the disclosures of the '658 patent, adequately suggests that the master server collects analytics based on enabled tool features to target specific products, services, or information to users, as required by claim limitation 1[c2].

3. Independent Claim 7

340. Independent claim 7 is analogous to independent claim 1, but also includes a few additional terms that I explain below. In my opinion, the combination of Dasgupta and Darnell renders obvious all of the claim limitations from independent claim 7 for the reasons I previously discussed in § XI.C.2 and the reasons I provide below. To streamline my discussion, I will incorporate analysis in § XI.C.2 through references to that section rather than repeat the entirety of my previous opinions.

341. I have also created this table to show the limitations of Claim 7 and their analogous limitations in Claim 1.

Claim 7	
Limitation	Substantially Similar Limitation(s)
7[pre]	1[pre]
7[a]	1[a], 1[b1]
7[b]	1[b3], 1[c1]
7[c]	1[b2]
7[d]	1[c1]
7[e]	1[c2]

a. Preamble of claim 7, “A computer system with”

342. As I explained in § XI.C.2.a, Dasgupta discloses a computer system. Dasgupta’s media content “system 100” is a “computer system.”

b. Claim 7, limitation [a], “a master server with a memory and a processor for storing and executing a master application”

343. As I explained in §§ XI.C.2.b-XI.A.2.c, Dasgupta’s delivery subsystem 202 includes a processor 304 and storage device 306, such as memory or storage drives. Figure 3 of Dasgupta illustrates this delivery subsystem that covers limitation 7[a]:

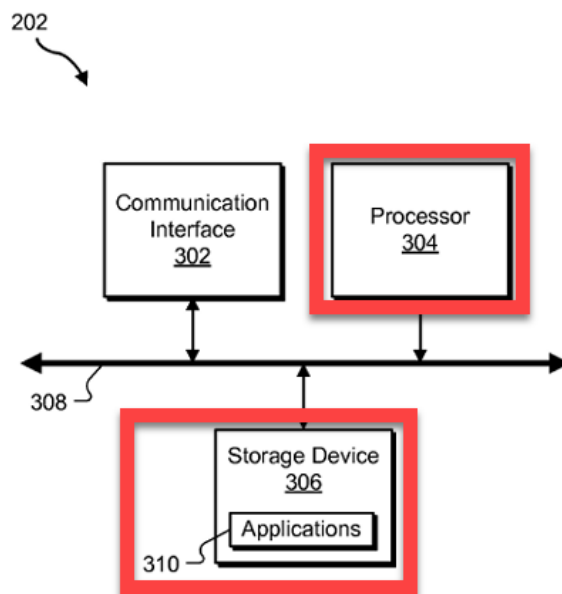


Fig. 3

Id., Fig. 3.

- c. **Claim 7, limitation [b], “with a remote tool box feature that programs a processor on a remote computer with a slide tool program for scrolling or paging through content data or content applications from multiple content data sources or service providers enabled by a user simultaneously on a web-page created by the master application and hosted on the master server using hardware on remote computer,”**

344. To address claim limitation 7[b], it is important to first understand its similarities and distinctions from claim limitation 1[b3]. I described these similarities and distinctions previously in § XI.A.3.c. Claim 7[b] is substantially similar to claim limitation 1[b3] except that it recites a “remote tool box feature” that programs the remote computer instead of the “master application.” A POSITA would have understood that the “remote tool box feature” was part of the “master application” because a “tool-box” is a “set of sub-features” of an application. ’658 patent, 4:16-20.

345. In my opinion, Dasgupta discloses or suggests claim limitation 7[b]. Dasgupta describes a user interface with functionality that enables users to navigate through a channel guide or advertisement interface that dynamically responds to user interactions. For instance, Dasgupta explains that the user interface, accessed remotely, permits a user to scroll through and select media content channels or advertisements. This functionality is detailed in Dasgupta’s description of a “three-

dimensional grid matrix” of channel icons, which users can navigate to view, highlight, or scroll through various icons, thereby organizing and displaying specific content data. Dasgupta, 14:45-15:44, 18:49-19:3, Figs. 7-9, 16-18. In this setup, a user’s selection and navigation through the channel guide icons act as selections of content data and applications from various sources, displaying them directly on the client’s device.

346. Regarding programming of the remote computer, Dasgupta discloses that applications, such as advertisement facility 106, are hosted on subsystem 202 and facilitate user interaction by prompting subsystem 204 to scroll through content. Dasgupta, 14:3-30; Fig. 7. This configuration demonstrates that Dasgupta’s system, through the “remote tool box feature,” allows remote interaction with and modification of displayed content, thus controlling how content from multiple sources is presented to the user on the remote device. A POSITA would have recognized this programming as the remote tool box feature programming the remote computer to interact with content data via a slide tool.

347. Dasgupta’s advertisement facility, which selects and personalizes advertisements displayed to the user based on user interactions, further exemplifies the “tool-box” concept disclosed in the ’658 patent, allowing for control over specific data sets. For instance, the advertisement facility programs and adjusts the display of a scrollable advertisement strip (Fig. 7), which users can scroll through or

otherwise adjust. Dasgupta, 14:3-67. This scrollable advertisement feature, functioning as a tool to modify the display based on user input, directly reflects the intended operation of a remote tool box feature.

d. Claim 7, limitation [c], “wherein the content data or content applications are simultaneously streamed to the master server from servers of multiple content data sources or service providers over the internet in real time to an account belonging to the user,”

348. In my opinion, Dasgupta discloses or suggests claim limitation 7[c]. Although similar to claim limitation 1[b2], claim limitation 7[c] specifically requires that content is streamed “in real time to an account belonging to the user” rather than being “viewable from the remote computer devices through the master server on the master web-page.” I explained this previously in § XI.A.3.d.

349. In my opinion, Dasgupta discloses or suggests claim limitation 7[c]. Dasgupta describes content that is streamed “in real time to an account belonging to the user” by implementing a user profile system that stores user-specific data, such as content viewing history and user preferences. For example, Dasgupta’s user profile, stored within subsystem 202, includes a range of user-specific data that governs content selection and display. This profile data, as Dasgupta explains, may comprise “one or more preferences of a user, historical media content access data associated with the user (e.g., data indicating one or more media channels and/or media content programs accessed by the user), and any other information about the

user that may be helpful for selecting a personalized advertisement for presentation to the user.” Dasgupta, 4:9-23. This setup is consistent with streaming content in real-time to “an account belonging to the user,” as the stored user data directly impacts the delivery and personalization of the media content.

350. Furthermore, Dasgupta demonstrates that this user profile information is used by subsystem 202 to tailor content specific to the individual’s account. By streaming personalized advertisements and selected media items directly based on the user’s stored data and preferences, Dasgupta’s system discloses the ’658 patent’s claim language of streaming “to an account belonging to the user.” Dasgupta’s user profile data, therefore, is the targeted account-specific data repository that curates content selection in real time for the user, as required by claim limitation 7[c].

- e. **Claim 7, limitation [d], “wherein the remote tool box feature allows the user to write code, select code or delete code to enable and disable tool features that control how sub-sets of the content data or the content applications from the multiple content sources or service providers are received, displayed or organized”**

351. Limitation 7[d] is substantially similar to limitation 1[c1]. For the same reasons as I explained in § XI.C.2.f, Dasgupta in combination with Darnell disclose claim limitation 7[d].

- f. **Claim 7, limitation [e], “and wherein the master server collects history analytics based which tool features are enabled to target specific products,**

services or information to the user.”

352. Limitation 7[e] is substantially similar to limitation 1[c2]. For the same reasons as I explained in § XI.C.2.g, Dasgupta discloses claim limitation 7[e].

4. Claim 4: “The cloud-based computer system of claim 1, wherein the master application includes tools for enabling security features that limits access to one or more of the content sources or service providers based a user log-in belonging to the user.”

353. I understand that claim limitation 4 requires “tools for enabling security features that limit access to one or more of the content sources or service providers based on a user log-in belonging to the user.” In my opinion, a POSITA would have found this limitation disclosed or suggested by Dasgupta in combination with Darnell for several reasons.

354. First, Dasgupta discloses a channel guide that is presented specifically to a “subscriber to one or more services,” indicating that the system caters to individual users who have specific content subscriptions and preferences. Dasgupta, 2:64-3:3, 7:3-19, 13:51-14:30. Although Dasgupta provides a channel guide to these subscribed users, it does not describe the specific mechanisms used for managing or securing user access based on their subscriptions. In my opinion, a POSITA would have been motivated to enhance Dasgupta’s system by incorporating Darnell’s feed subscription model, which allows users to actively “create a list of feed subscriptions via the feed reader user interface 116.” Darnell ¶40. Combining Darnell’s feed

selection approach would allow users to subscribe to specific content sources and thereby customize the channel guide displayed to them, providing a more personalized and restricted access experience.

355. In addition to personalized subscriptions, Darnell further details specific security measures to verify authorized requests, such as user authentication protocols involving a user ID and token to “authenticate requests from the user.” *Id.*, ¶84. In Darnell, the user’s ID and token are stored in a “user preference table” on the server along with other user-specific data, thereby ensuring that only authenticated users can access their feed subscriptions and related content. *Id.*, ¶¶84-86. In my opinion, a POSITA would have recognized the utility of this method in Dasgupta’s system because it would provide a secure method for ensuring that only the authorized user can access the content associated with their account.

356. Moreover, Dasgupta already maintains user profile data, such as user preferences, in storage facility 112 on the server. Dasgupta, 8:54-67. In my opinion, a POSITA would have been motivated to store additional user-specific security features in Dasgupta’s profile data, which would align with Darnell’s approach of maintaining authentication tokens and user IDs within a secure, server-based preference table. By combining Darnell’s approach, Dasgupta’s system would thereby ensure valid requests for media content access and enable display of the appropriate channel guide specifically authorized for each user.

357. Further, in my opinion, a POSITA would have found it desirable to incorporate Darnell's security features into Dasgupta's media content system to enhance the privacy and security of the user's account. As explained above, Dasgupta already provides a system that stores user profile data, which could include sensitive information, such as historical media content access data. Dasgupta, 4:9-23. By incorporating Darnell's secure authentication measures, Dasgupta's system would prevent unauthorized access to this stored user data, thereby protecting users' preferences, subscriptions, and viewing history from unauthorized access or manipulation.

358. A POSITA would also have had a reasonable expectation of success in implementing Darnell's user ID and token in Dasgupta's media content access system. This is because Dasgupta already stores user profile data, and the addition of user authentication data, such as a user ID or token, would be a straightforward enhancement to the existing system's storage structure. Additionally, Darnell demonstrates that its security technique successfully limits content source access to authorized users in the user interface. A POSITA would have recognized this as a conventional, predictable method for limiting access in content aggregation systems, following the standard practices in the art.

5. **Claims 5 & 8: [Claim 5] "wherein the master server collects and stores history analytics based content data and content applications viewed by the user"; [Claim 8] "wherein the master server collects and stores history analytics based on**

usage of the remote computer.”

359. In my opinion, claims 5 and 8, which are substantially similar, would be disclosed or suggested by Dasgupta based on its description of a server-stored database that maintains “historical media content access data.” As I explained above for limitation 1[c2], Dasgupta discloses that advertisement facility 106 displays advertisements to users based on “information about the user,” which includes stored user profile data containing “historical media content access data associated with the user” and user preferences. Dasgupta, 4:9-23. This profile data is stored in storage facility 112 on the server, which is a database for maintaining information about users’ past interactions with media content. *Id.*, 4:54-67. In my opinion, a POSITA would have understood this profile data to include, among other things, details about the user’s content-viewing history and access habits.

360. Dasgupta’s storage of user profile data, including historical access data, serves as a tool for analyzing user behavior to determine which content may be most relevant or engaging. In my opinion, a POSITA would have recognized this historical media content access data as part of the server’s analytics capabilities, allowing the system to record, analyze, and ultimately act on the user’s engagement with various content types over time. This historical data enables targeted advertising and content recommendations based on the user’s past media

interactions, which meets the claimed requirement that the server collects and stores “history analytics” for content.

361. A POSITA would have further recognized that Dasgupta’s database includes information specific to the user’s profile and engagement history and that this data would be valuable for personalizing the content or advertisements displayed to the user. By storing this historical information, Dasgupta’s system enables analytics-driven content targeting based on stored user data. This would meet the limitations set forth in claims 5 and 8, as it allows the server to record user interactions and organize content options based on a user’s historical behavior with content and applications.

6. Claim 6: “The cloud-based computer system of claim 5, the master application includes a purchase tool that allows the user to purchase goods or services from the multiple content sources or service providers from master web-page.”

362. As mentioned previously in § XI.B.2, this dependent claim adds the “purchase tool.” In my opinion, claim 6’s limitations are disclosed or suggested by the combination of Dasgupta and Darnell. Specifically, as I have described above for limitations 1[c2] and in the analysis of claim 6 in Ground 1, Dasgupta’s system enables advertisements and a virtual store that would serve as a “purchase tool” as claimed. Dasgupta describes how advertisement facility 106 not only displays advertisements to the user but also includes a virtual store accessible via user prompts on the channel guide interface, where products and media content, such as

videos, are made available for purchase, rent, or subscription. Dasgupta, 13:52-14:2, 14:31-44. A POSITA would have understood that this virtual store feature enables users to directly acquire goods or services within Dasgupta's media content system, which meets the claimed "purchase tool."

363. Additionally, Dasgupta describes media content provider subsystem 208, which supplies media content—such as videos and other items—for presentation within Dasgupta's media content access system, illustrating the provision of content from multiple content sources or service providers. *Id.*, 7:3-19, 17:64-18:3. In my opinion, a POSITA would have recognized that subsystem 208 provides access to various content sources, making Dasgupta's virtual store a hub where a user can obtain goods or services from these multiple sources, as required by claim 6.

364. As I explained for limitation 1[b2], Dasgupta's channel guide is streamed to subsystem 204 (the remote computer) via the HTTP protocol, providing the user with access to the channel guide interface on a webpage. *Id.*, 7:20-29, 7:36-43. In my opinion, this streamed webpage allows the user to access the virtual store interface remotely from subsystem 204 through subsystem 202, or the "master webpage," thus aligning with the claimed requirement that content is accessed from a master web-page.

365. In combining Dasgupta's system with the security methods disclosed in Darnell, a POSITA would have recognized that Dasgupta's advertisement facility could further be enhanced by integrating Darnell's secure feed subscription and authorization features, ensuring that only authorized users can access and purchase the content provided through the virtual store. This integration of Darnell's security mechanisms would enhance the functionality of Dasgupta's virtual store as a secure, user-specific purchase tool.

7. Claim 9: "The computer system of claim 8, wherein the master server uses the history analytics to select content data or the content applications based on a user log-in belonging to the user."

366. In my opinion, claim 9's limitations are disclosed or suggested by Dasgupta. As I discussed for claim limitation 1[c2], Dasgupta discloses that "historical media content access data" is stored in a user's profile and used by advertisement facility 106 to select personalized advertisements for display to that user. Specifically, Dasgupta explains that user profile data, including historical media content access data and preferences, is stored in storage facility 112. Dasgupta, 4:9-23. This data enables advertisement facility 106 to analyze and tailor advertisement selections based on information specific to the user, effectively using the stored historical data as "history analytics" that guides which advertisements are targeted to the user. A POSITA would have thus recognized that the user's media content access data is integral to the "user profile," and that this data inherently

requires user-specific attributes tied to the individual user, or “based on a user log-in belonging to the user,” as claimed.

367. Additionally, as I explained in claim limitation 4, incorporating Darnell’s secure login features into Dasgupta’s system would further enhance the security of the user’s profile by ensuring that only authorized users could access or modify the stored profile data. Darnell describes implementing security measures, such as logging into the system to verify that content requests are authorized, thereby preventing unauthorized access to feed content. Darnell, 17:30-42. By integrating Darnell’s login and token features, Dasgupta’s system would effectively link each user profile to a unique, authenticated user login, which meets the requirement that history analytics are based on “a user log-in belonging to the user.”

8. Claim 10: “The computer system of claim 7, wherein the user interface is an icon-based master web-page.”

368. In my opinion, dependent claim 10 is disclosed or suggested by Dasgupta. As discussed in claim limitation 1[b2], Dasgupta teaches a web-based media content access system that includes a graphical user interface (GUI) in the form of a channel guide. This channel guide contains “channel icons” representing various media content channels, which are selectable by the user. Dasgupta, 1:64-2:16, 13:9-23. A POSITA would have understood these channel icons to provide users with a convenient, visual means of selecting and accessing content channels within the interface, thus making the GUI “icon-based.”

369. The system described in Dasgupta presents these icons as part of a structured, web-based GUI that serves as a centralized interface or “master web-page” for media content access. In particular, Dasgupta explains that the channel guide organizes the icons in a grid or matrix layout, which can even be displayed in three-dimensional form on a “stage,” providing an intuitive and visually organized user experience. Dasgupta, 17:43-63, 18:49-19:3; Figs. 7 & 16. This configuration allows the user to interact with the icon-based interface by navigating through the channel guide to select specific content channels, which meets the “icon-based master web-page” as claimed.

370. Furthermore, Dasgupta’s disclosure of the web-based GUI interface, which streams content to remote computer devices over a network, would reinforce that this channel guide operates as a central, server-based “master web-page” accessible to users. The icons within this interface, each representing a media content channel or service, would thus constitute the claimed “icon-based” functionality for navigating and accessing content as required by claim 10.

D. GROUND 2B: DASGUPTA, DARNELL, AND KIM FURTHER RENDER CLAIMS 2, 3, AND 11 INVALID AS OBVIOUS

371. In Ground 2B, I combine Kim with Dasgupta and Darnell to meet the limitations of claims 2, 3, and 11. In my opinion, Dasgupta in view of Darnell and Kim renders obvious claims 2, 3, and 11 for the reasons that follow.

1. Motivation to Combine Kim with Dasgupta and Darnell

372. As explained previously in § XI.B.1, a POSITA would have combined Dasgupta with Darnell and Kim for several reasons. A POSITA would also have combined Kim with Dasgupta and/or Darnell for several reasons.

373. A POSITA would have been motivated to combine the teachings of Dasgupta, Darnell, and Kim for several reasons, each aimed at enhancing the functionality and user experience of a unified media content access system.

374. First, a POSITA would have recognized that integrating Kim's email and blog content with Dasgupta's media content system is consistent with an established trend at the time of the priority date of the '658 patent toward unified digital interfaces that bring together diverse types of media content into a single user interface. Dasgupta's system already discloses a user-friendly interface that aggregates various media channels into a single, accessible GUI, providing a centralized location for users to navigate media content. Combining Kim's teachings, which include an email and blog feed aggregation system, would streamline user access to multiple forms of content within this single interface, eliminating the need for users to switch between separate applications for television, email, and social media feeds. By consolidating these functionalities into a single, cohesive interface, a POSITA would have recognized that the user experience is

enhanced, creating a comprehensive home entertainment and communication hub that meets evolving user expectations for simplicity and integration.

375. Furthermore, a POSITA would have understood that the addition of Kim's email and social media content reduces the cognitive load on users by simplifying the experience of interacting with multiple types of content. In environments where users may be passively viewing television content, such as home or office settings, enabling access to email and social media within the same interface would allow users to monitor important communications, including critical emails or notifications, without needing to divert attention from the main screen. This configuration would be particularly advantageous in contexts where users benefit from seeing notifications or communications alongside other media content without the friction of accessing a separate device. By applying Kim's teachings in this way, Dasgupta's media content system would have provided users with an accessible and highly efficient interface, which would have been a desirable improvement from the perspective of a POSITA at the time of the alleged invention of the '658 patent.

376. Additionally, incorporating Kim's content aggregation techniques into Dasgupta's system would increase user engagement and retention by offering additional features that keep users within the same interface for extended periods. By enabling users to check email, read blog feeds, and watch television content

without leaving the unified interface, the system would become a more versatile tool, reducing the reliance on external devices and applications. A POSITA would have recognized that providing access to diverse content types in a single interface allows for continuous interaction and engagement, supporting the intended design and functionality of Dasgupta's user interface as a multi-functional media access system.

377. A POSITA would have further understood that incentivizing a user to maintain interaction with an application that aggregates email, blog, and other media content increases the monetization opportunities of that application (i.e., the longer a user spends on an application, the more the application can provide tailored advertisements). Dasgupta already discloses that advertisements are tailored to the user, based on user metadata. Dasgupta, 3:63-4:23. The personalized advertisements become more effective as a user spends more time on the application, since the system can collect more user-specific data to serve more relevant advertisements. As such, a POSITA would have been motivated to integrate Kim's email and blog features to increase the active time a user spends on Dasgupta's system, so that Dasgupta's system could collect more user data and serve more relevant, personalized advertisements.

378. From a technical perspective, Kim's teachings would readily integrate with Dasgupta's system because Dasgupta discloses a media content access facility that aggregates and presents various media content types transmitted over

standardized protocols, such as HTTP. As Dasgupta notes, “any technologies suitable for accessing media content” may be used to access content (3:14-28), indicating that additional content types could be aggregated within its interface. Kim’s email content and blog feeds are transmitted over widely adopted protocols compatible with Dasgupta’s access methods. Thus, a POSITA would have had a reasonable expectation of success combining Kim’s additional media content (email and blogs) with Dasgupta’s system, particularly since Kim explicitly describes accessing email and blogs within a feed aggregation interface similar to Dasgupta’s GUI.

379. Incorporating these aspects of Kim’s system into Dasgupta’s media content access interface would be a straightforward application of known content aggregation techniques. A POSITA would have understood that combining these techniques, according to their known functions, would yield predictable results. Accordingly, applying Kim’s teachings to Dasgupta’s system would be nothing more than applying established methods to achieve a unified content aggregation system, enhancing the user experience in a manner consistent with both Dasgupta’s and Kim’s design objectives.

2. **Claim 2: “The cloud-based computer system of claim 1, wherein the content sources or service providers include one or more personal content account and at least one or**

more social media account belonging to the user.”

380. Claim 2 depends from claim 1. In my opinion, Dasgupta in combination with Kim discloses or suggests the limitations of claim 2. Dasgupta discloses that its media content access system is configured to aggregate and present “any content that may be accessed and/or presented by a media content access system for experiencing by a user.” Dasgupta, 2:25-38. A POSITA would have recognized that such a broad disclosure encompasses diverse forms of content, intended to provide users with a unified experience that includes any accessible media within the scope of the system’s display capabilities.

381. Furthermore, Kim teaches the aggregation of email and blog content, which it describes as “personal content accounts” and “social media accounts” associated with the user. Kim, ¶¶21-22, 33, 36-39, 45, 58-59, 72, Figs. 3-9, 11-13. A POSITA would therefore have understood that Dasgupta’s media content system could readily integrate Kim’s content types—namely, email and social media accounts—to achieve a more complete and user-centric content experience. As noted in my previous analysis, Kim’s system receives email and blog content in a manner consistent with a feed aggregation interface, thereby facilitating easy integration into Dasgupta’s framework.

382. From a usability standpoint, a POSITA would have been motivated to include Kim’s email and blog accounts within Dasgupta’s media content system to

simplify and enhance the user experience. Integrating these additional content sources directly into Dasgupta's unified interface would allow users to access television, email, and social media content without the burden of switching between multiple applications or devices. Such a consolidation of user activities within a single interface would have been consistent with design principles at the time of the alleged invention, focused on reducing cognitive load and providing an efficient, uninterrupted media consumption experience. A POSITA would have recognized that allowing users to access various content types in a single, cohesive user interface not only improves convenience but also encourages longer engagement with the system, as users can remain within the interface to complete a range of tasks. A POSITA would have further recognized that longer engagement with the system would result in more data collected on the user, resulting in the display of more relevant and personalized advertisements, as described in Dasgupta.

383. The technical feasibility of this combination is further underscored by Dasgupta's explicit disclosure that "any technologies suitable for accessing media content" may be used to access content, suggesting that the system is compatible with different transmission protocols. Dasgupta, 3:14-28. As Kim shows, email and blog feeds are transmitted via known data protocols and can be displayed within a feed aggregation interface. A POSITA would therefore have a reasonable expectation of success in incorporating Kim's email and blog content into

Dasgupta's media content access system, given that Dasgupta's framework is flexible enough to handle various content types and access methods.

384. In my view, integrating Kim's content types with Dasgupta's system would have been a straightforward application of conventional content aggregation methods, operating according to their intended functions to yield predictable results. A POSITA would therefore have understood that adding email and blog feeds to Dasgupta's media content system represents a logical and beneficial enhancement to the user experience, aligning with the overarching goal of Dasgupta's system to aggregate diverse media content into a unified, user-friendly interface.

3. Claims 3 & 11: [Claim 3] "wherein the personal content account is an e-mail account"; [Claim 11] "wherein one of the content sources or service providers includes an e-mail web-site belonging to the user."

385. Claim 3 also depends from claim 1, and claim 11 also depends from claim 7. As I showed in my table in § V.C.1 above, these dependent claims are substantively identical, so I will address them here together instead of separately.

386. In my opinion, the combination of Dasgupta, Darnell, and Kim discloses or suggests the limitations of dependent claims 3 and 11. As discussed in my analysis for claim 2, Dasgupta's media content access system is designed to aggregate and present various content types in a unified user interface. When combined with Kim's teaching of a "personal content account," which includes email, a POSITA would have understood that email functionality could be

incorporated into Dasgupta's system to enhance the user's experience by allowing access to personal communication services within the same interface.

387. Kim teaches that personal content accounts, such as email accounts, can be managed alongside other content types within a feed aggregation interface, enabling users to access email without navigating away from the primary system interface. Kim, ¶¶21-22, 33, 36-39, 58-59, 72, Figs. 3-9, 11-13. In Dasgupta's system, this would mean that email, as a "personal content account," could be accessible from the same master web-page where media content is aggregated. A POSITA would have found such integration beneficial, as it would eliminate the need to exit the system to check email, supporting a more efficient and consolidated user experience. A POSITA would have understood that an "e-mail web-site" is a content source for email, and receiving email from a website would be substantially the same as receiving email from another source (e.g., a standalone program like Outlook).

388. Claim 11 is substantively similar to claim 3, differing only in specifying that the email account originates from an "e-mail web-site belonging to the user." In my view, a POSITA would have understood this variation as a straightforward implementation within the same integrated user interface. Kim's disclosure of accessing user-specific email accounts would have complemented Dasgupta's system's flexibility in accommodating multiple content sources. This combination

would allow users of Dasgupta's system to view emails directly from the content sources they are subscribed to, whether accessed from a personal or publicly accessible email platform.

389. Thus, the combination of Dasgupta, Darnell, and Kim, for reasons set forth in my analysis of claim 2 and reiterated here, would disclose or suggest the limitations of dependent claims 3 and 11, and would have been understood by a POSITA as a practical and predictable addition to Dasgupta's system, supporting the inclusion of personal email within a consolidated content access interface.

XII. '068 PATENT: DETAILED EXPLANATION OF THE GROUNDS

A. GROUND 1: DARNELL AND KIM RENDER CLAIM 1 OBVIOUS.

390. In my opinion, as discussed below, the combination of Darnell and Kim renders claim 1 obvious to a POSITA at the time of the alleged invention on October 12, 2010.

1. Motivation to Combine Darnell with Kim

391. In my opinion, Kim's teachings of aggregating diverse content types, such as news, blogs, and emails, are suitable for use in Darnell's feed reader system, in which content sources are aggregated on a remote server rather than at a user device. Kim, Abstract, ¶¶2, 18; Darnell, Abstract, ¶¶30-32. Darnell and Kim address similar problems of aggregating and presenting multiple feed sources to a user. For example, Darnell addresses "[l]ooking through a dense list of content items to locate

items of interest,” which is cumbersome for a user. Darnell, ¶9. And Kim similarly addresses the problem of viewing multiple pieces of fragmented content from news sources, blog entries, email, and the like in a “consuming experience having significant drawbacks.” Kim, ¶1. Both Darnell and Kim utilize RSS or XML-based formats for feed content, which enables integration of feed items from different sources. Darnell, ¶¶36-37; Kim, ¶¶24, 37. For example, both Darnell and Kim may receive feed content using RSS or similar feeds, motivating a POSITA to combine complementary features in receiving feed content using RSS with a reasonable expectation of success. In my opinion, Kim’s feed application, email application, media application, and photo application could be implemented using Darnell’s server system 104 rather than on a user device, which would allow the user to access personalized content feeds on any user device (e.g., phone, computer, tablet, etc.) instead of only on a single user device. Moreover, Kim was cited by the Examiner during the prosecution of Darnell suggesting an underlying relevance between the two references, indicating that Kim’s disclosure was viewed as pertinent to Darnell’s system. EX1009, 20. Citing Kim in Darnell’s prosecution underscores that a POSITA would have reasonably been motivated to combine Kim’s teachings with Darnell.

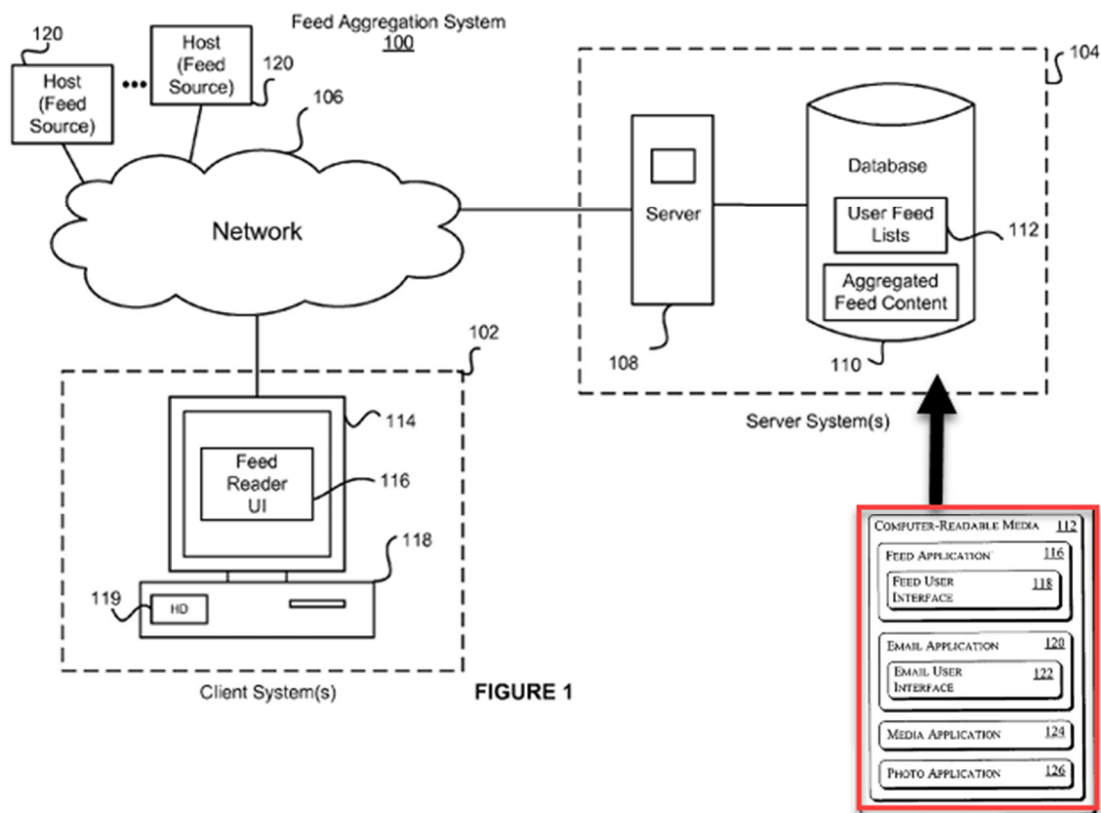
392. The inclusion of Kim’s classification and trait-based organization with Darnell’s server-oriented architecture would represent a straightforward extension

of Darnell's server system, where Kim's content analysis and tailored presentation functionalities would have been incorporated directly into Darnell's server system. This combination would bring the benefits of Kim's trait-organized interface to any user device accessing Darnell's server-based architecture, offering users a cohesive experience across multiple devices without dependence on a single local application. Darnell, ¶32, Fig. 1.

393. A POSITA would have been motivated to combine these teachings for several reasons. For example, both references are directed toward content aggregation systems that centralize content from multiple sources and present it to a user through a unified interface on a user device. Darnell, ¶¶8, 11-14, 30-32, Fig. 1; Kim, Abstract, ¶¶18-23, Fig. 1. Both references share the primary goal of simplifying content presentation from diverse feeds and specifically address the inefficiencies users face when attempting to locate specific items of interest within a large volume of content. For example, Darnell explicitly identifies the problem of users "looking through a dense list of content items to locate items of interest" as cumbersome (Darnell, ¶9), while Kim describes a "consuming experience having significant drawbacks" due to the fragmented presentation of content from news, blogs, and email (Kim, ¶1). By combining Kim's trait-based classification system with Darnell's server-based structure, a user would achieve a more efficient, organized, and user-friendly interface for presenting content.

394. Furthermore, a POSITA would have been motivated to incorporate Kim's aggregation of news, blogs, and email in a network computer and sending the feeds to a remote computer for display in a user interface, within Darnell's architecture to allow for centralized analysis and grouping based on content traits on the server before content is sent to a client. Kim, ¶¶37, 39. A POSITA would have seen value in implementing Kim's trait identification tools within Darnell's server system 104, because it would have allowed users to access personalized, trait-organized feeds from any device, removing the dependency on a single user device and enhancing the flexibility and portability of the system. Kim, ¶¶20-24, 33, 38-39. For example, Kim discloses that "[a] feed from Yahoo!™ News for instance, may predominantly comprise links to news articles." Kim, ¶24. Yahoo! News stores content on a remote server that is then served to a user device. As such, a POSITA would have recognized that Kim's content aggregation system was comparable with Darnell's server-based feed aggregation system. Such a combination would have been motivated by the increasing demand for cloud-based, device-agnostic services available at the time.

395. To illustrate the simplicity of combining Kim and Darnell, the following figure combines the architectures of Kim and Darnell, specifically moving Kim's computer readable media to Darnell's server system.



Darnell, Fig. 1.

396. Both Darnell and Kim disclose systems that aggregate content feeds and present them in user-friendly formats, each aiming to provide users with an organized and accessible interface. Combining Darnell and Kim would have been nothing more than using Kim’s known technique of detailed, trait-oriented organization in the same way as in Darnell’s remote server. Darnell, Abstract, ¶¶30-32; Kim, ¶¶20-24, Figs. 3-5. Specifically, a POSITA would have recognized the logical extension of Darnell’s aggregation capabilities by including Kim’s classification of items into “link,” “text,” “enclosed media,” and “conversation”

formats based on their content traits. Kim, ¶¶26. This combination would have provided users with a streamlined and tailored experience across content types, enhancing content accessibility and reducing time spent sifting through disorganized items.

397. Applying Kim's trait-based classification to would have been a predictable improvement to Darnell's server feed aggregation system. Both references disclose user-oriented content presentation, and in my opinion, integrating Kim's method of grouping by traits would achieve a more refined and customizable user experience without requiring significant modifications to Darnell's architecture. Using Kim's organization of feeds based on user history and relevance would have improved Darnell's similar system by tailoring content presentation to individual user preferences, making frequently accessed or popular feeds readily available. Kim, ¶¶68-70. The user-centric emphasis of both systems would have been retained and enhanced through such a combination, allowing the resulting interface to display feeds that are both aggregated remotely and organized for efficient user consumption.

398. Thus, in my opinion, a POSITA would have been motivated to combine Darnell and Kim not only due to their complementary goals and solutions but also because this combination would have involved merely applying known techniques (Kim's trait-based sorting) to an existing feed aggregation system using a remote

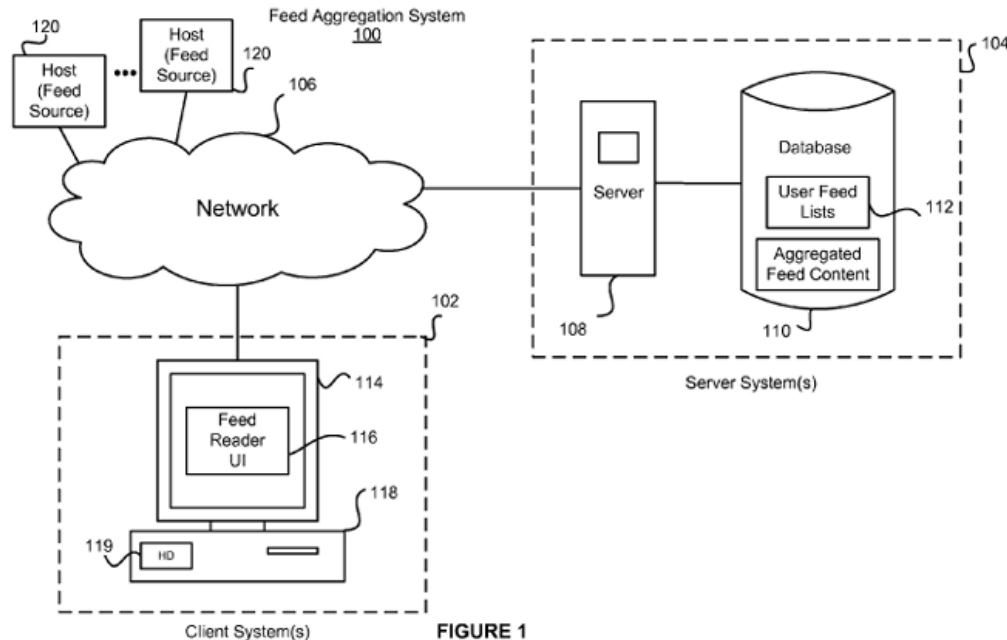
server architecture (Darnell). And using Kim’s organization of feeds based on user history and relevance would have improved Darnell’s similar system by tailoring content presentation to individual user preferences, making frequently accessed or popular feeds readily available. Kim, ¶¶68-70. This approach reflects routine skill in the art by implementing Kim’s trait identification and trait-based grouping as a means to enhance user accessibility within Darnell’s already centralized feed reader environment. By doing so, a POSITA would have successfully created a highly adaptive and cross-device content aggregation system, addressing both the user’s need for organized content and the industry’s shift toward centralized, server-based solutions.

2. Independent Claim 1

a. Preamble of claim 1, “A cloud-based computer system comprising:”

399. To the extent the preamble of Claim 1 is limiting, Darnell’s disclosure meets the definition of a “cloud-based computer system” as claimed in the ’068 patent. Darnell’s Figure 1 (below) illustrates a “content feed aggregation system” that includes “a server system 104” communicating with “one or more client systems 102” and “one or more hosts 120 (or ‘feed sources’) by network 106.” Darnell, ¶32. Both client and server systems are computers. *Id.*, ¶¶61-80. And Darnell further explains “network 106 may be any suitable wired and/or wireless network and may

include a local area network (LAN), wide area network (WAN), the Internet, metropolitan area network (MAN), or any combination of such networks.” *Id.*, ¶32



Darnell, Fig. 1.

400. This structure is illustrated in Darnell’s Figure 1, which depicts a server-client setup designed to aggregate and distribute content across a network, which a POSITA would have recognized as a cloud-based computer system. ’068 patent, 1:51-54, 4:38-41 (“Cloud-based computer system or architecture includes multiple content data sources or service provider servers that are linked to a log-in server or master server via the internet/intranet.”).

401. Darnell provides a comprehensive description of the server and client systems, making it clear that each is a computer with specific roles in the aggregation

and distribution of content to remote clients. For example, Darnell describes that client systems may be configured to receive, store, and organize feed content aggregated by the server. Darnell, ¶¶61-80. Darnell also explains that network 106 “may be any suitable wired and/or wireless network and may include a local area network (LAN), wide area network (WAN), the Internet, metropolitan area network (MAN), or any combination of such networks.” *Id.*, ¶32. This setup allows the server to aggregate content from various sources and deliver it to client devices across a geographically dispersed network, a configuration aligning with cloud-based architecture.

402. Darnell’s disclosure emphasizes that the server’s operations extend to managing and processing content for diverse feeds stored in a shared repository. Specifically, Darnell explains that the server system may include or have access to one or more databases configured to store the feeds, allowing dynamic retrieval and management of content stored centrally. *Id.*, ¶¶34-35. Darnell’s further description of using databases to manage content aggregation and access indicates a server-centric model that distributes content from a centralized location to various clients.

403. Darnell’s network structure also supports flexible and scalable access to the aggregated feed content. The description of the server system 104 accessing hosts 120 to collect feed content and aggregating it for user access via client systems would have aligned with a POSITA’s understanding of a cloud-based system. For

example, the '068 patent describes a “[c]loud-based computer system or architecture [that] includes multiple content data sources or service provider servers that are linked to a log-in server or master server via the internet/intranet.” ’068 Pat., 1:51-54, 4:38-41. This disclosure is like Darnell’s use of a central server to distribute aggregated content over a network to client devices and indicates a cloud-based approach to handling, storing, and distributing feed content for user access.

404. Finally, Darnell’s specification of a wide range of content types—digital audio, video, and text—that may be aggregated on the server for user access further solidifies its applicability as a cloud-based system. Darnell, ¶36. The ability of the server to manage multiple data types and serve them to various client devices over a broad network structure fulfills the general characteristics of a cloud-based computer system, providing centralized storage, dynamic content management, and networked access.

b. Claim 1, limitation [a], “master server with a processor and memory, wherein the master server accessible over the internet/Intranet from remote computer devices by a user”

405. In my opinion, Darnell discloses or at least suggests the limitation of a “master server with a processor and memory, wherein the master server [is] accessible over the internet/Intranet from remote computer devices by a user.” Darnell’s server system 104 is a “master server” as claimed, as it provides the components of processor and memory required to support the system’s aggregation

and distribution of content across multiple remote client systems. In Figure 1 of Darnell, server system 104 (master server) “includes a server 108 and a database 110.” Darnell, ¶33. Darnell further illustrates the server in Figure 5, which includes a CPU 502 and memory 504. *Id.*, ¶71. Memory 504 (memory) “includes high-speed random access memory, such as DRAM, SRAM, DDR, RAM or other random access solid state memory devices,” and CPU 502 (a processor) executes instructions. *Id.*, ¶71. Server system 104 may be implemented on a single computer or a distributed system of multiple computers. *Id.*, ¶35. A POSITA would have understood that Darnell’s server system 104, for example as implemented in a single computer, is a master server because, as explained below, it aggregates feeds from multiple sources and distributes them to clients. ’068 patent, 4:19-22.

406. I have annotated Darnell’s Figure 5 below, which shows memory 504 outlined in red and the processing unit outlined in blue.

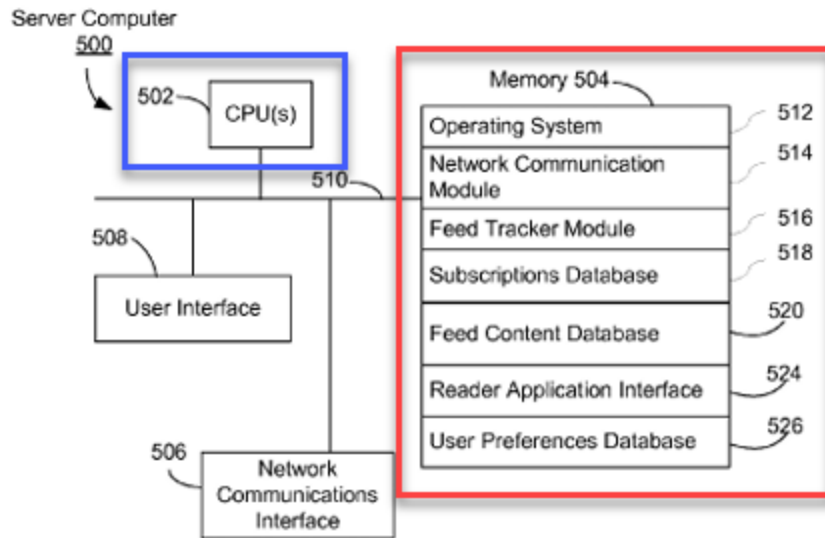


Figure 5

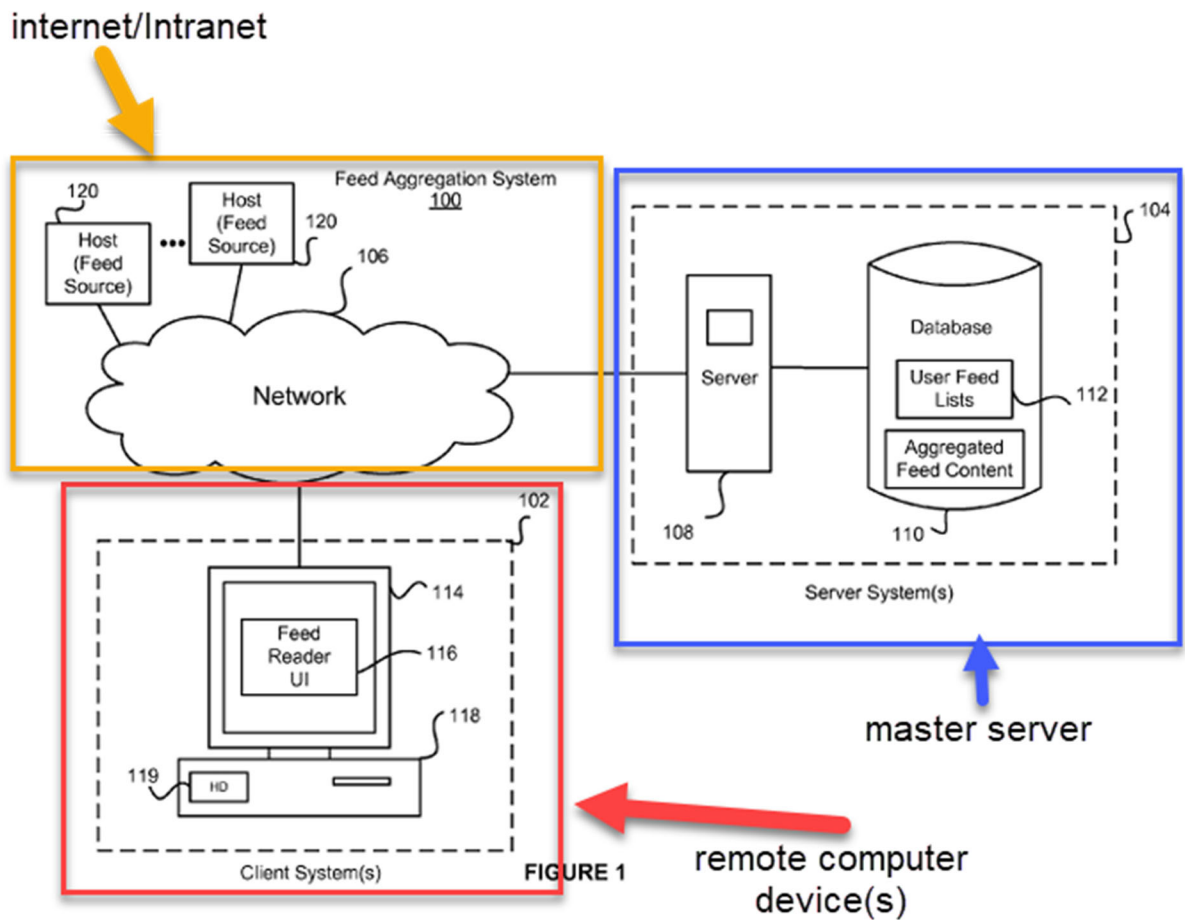
Darnell, Fig. 5.

407. Further, Darnell explains that server system 104 may be implemented on a single computer, housing both the memory and CPU in a unified configuration. *Id.*, ¶35. This single-server configuration would support the functionality of a “master server” as described in the ’068 patent, where a master server facilitates centralized control over content data and provides access to client systems through a unified interface. ’068 Pat., 4:19-22.

408. Importantly, Darnell also describes how users can access the server system 104 remotely through client devices, indicating its accessibility “over network 106,” which may be any combination of LAN, WAN, or the Internet. Darnell, ¶32. This accessibility over a network enables Darnell’s server to meet the requirement of a master server being “accessible over the internet/Intranet from

remote computer devices by a user,” as recited in the ’068 patent claim. Darnell’s server system 104 is accessible to users operating a client system 102, which may be a “computer 118 or computer controlled device, such as a personal digital assistant (PDA), cellular telephone, or the like.” *Id.*, ¶38. This structure allows users to connect with the server and view content on their respective devices, demonstrating that Darnell’s server system is indeed configured for remote accessibility through the Internet or intranet.

409. I also annotated Figure 1 of Darnell, which shows the master server highlighted in blue, the internet/Intranet highlighted in yellow, and the remote computer devices highlighted in red.



Darnell, Fig. 1.

410. Therefore, in light of the teachings in Darnell, it is my opinion that a POSITA would have recognized Darnell’s server system 104 as a “master server with a processor and memory” that is accessible over the internet/intranet from remote computing devices by a user.

- c. **Claim 1, limitation [b1], “a master application stored in the memory of the master server and that executes on the processor of the master server and that includes software coded for organizing content data and content applications from multiple content**

sources or service providers pre-selected”

411. Darnell discloses or suggests a “master application” that is stored in the memory of a master server and executed by a processor, containing software designed to organize content data and applications sourced from multiple content providers. Darnell describes server system 104 as equipped with memory 504, which stores several programs, modules, and data structures, collectively functioning as a master application. Darnell, ¶¶70–80; Fig. 5 (annotated below).

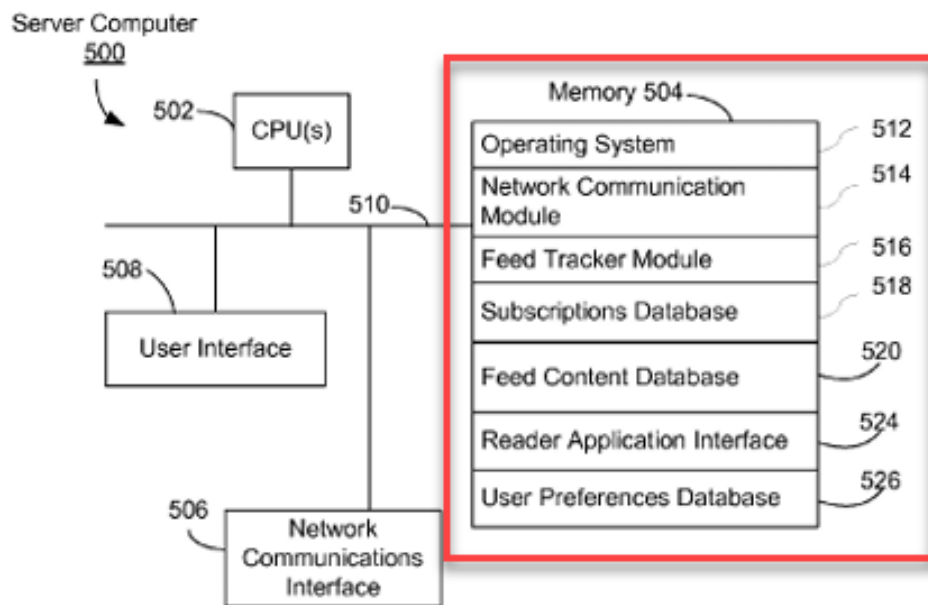


Figure 5

Darnell, Fig. 5.

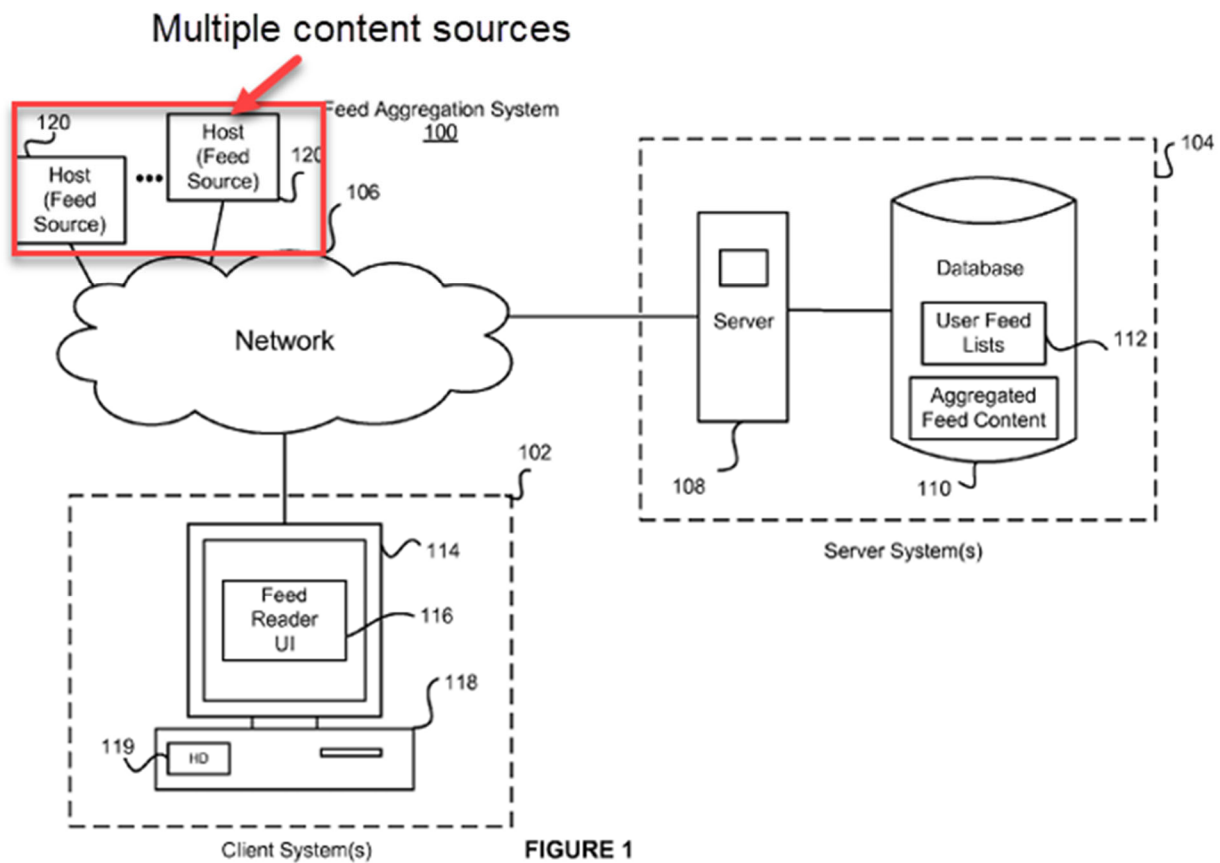
412. Darnell’s feed content database 520 “stor[es] content items from content feeds and user information about content items.” Darnell, ¶76. Such content “may be any machine-readable data, including but not limited to web pages, images,

digital audio, digital video, Portable Document Format (PDF) documents, and so forth.” Darnell, ¶36. Furthermore, a content item “may include the content associated with the item itself..., a title..., and/or a description of the content, a network location or locator (e.g., URL) of the content, or any combination thereof.” *Id.* The content data is received at the server and stored remotely in a database before being sent over the Internet to the user through the feed reader interface. *Id.*, ¶¶33 (“Server 108, sometimes called a front end server, retrieves the content feeds from the feed sources 120, and also provides an interface between the server system 104 and the client systems 102.”), 34 (“The server system 104 stores content items in the content feeds in the database 110.”), 40, 76, 81, 114; ’068 Patent, 4:13-15.

413. These programs and modules correspond to distinct sets of instructions performing specific functionalities. The server also includes a database 520 that stores content items and user information about these items. Darnell, ¶76. Additionally, Darnell discloses a subscriptions database 518, which retains information regarding users’ content feed subscriptions, and a user preferences database 526 that stores user-specific interface preferences. *Id.*, ¶¶75, 78. Subscriptions database 518 “stor[es] information about the content feed subscriptions of users of the system.” *Id.*, ¶75. Reader application interface 524 is “for exchanging information with the feed reader modules in one or more client computers,” and user preferences database 526 is “for storing user-specific

information, including user preferences with regard to the user interface.” *Id.*, ¶¶77-78. Furthermore, Darnell’s server includes an operating system to run the programs stored in memory. *Id.*, ¶72. Darnell’s Figure 5, for instance, illustrates this setup, where memory 504 and CPU 502 interact to carry out the functions required of a master server application. *Id.*, Fig. 5. A POSITA would have understood that Darnell’s modules and data structures on the server collectively are a master application that resides in memory 504 (memory) of server system 104 (the master server) and executes on the CPU 502 (the processor).

414. Darnell’s server system 104 also communicates with multiple feed sources 120, which represent various content sources or service providers. The specification defines “content feed” as a resource or service that offers lists of content items present or updated at feed sources. *Id.*, ¶36. Darnell explains that “[a] content feed (or channel) ... provides a list of content items that are present, recently added, or recently updated at a feed source 120.” *Id.* “A user interfaces with the server system 104 and views content items at a client system or device 102” via “a feed reader user interface 116 that may be presented on the display 114” and that allows the user to “create a list of feed subscriptions” (pre-selected). *Id.*, ¶¶39-40. The feed sources identified in Figure 1 are the “multiple content sources” of claim 1 in the ’068 patent.



Darnell, Fig. 1.

415. Users can interact with the server to view these content items on client systems (i.e., remote computing devices) and can further refine the content they wish to see by selecting specific feed subscriptions. *Id.*, ¶39. A user interacts with Darnell’s system using a feed reader interface that allows them to create a personalized list of feed subscriptions, thus effectively preselecting content sources or providers based on individual preferences. *Id.*, ¶40. This list is displayed to the user through a web-based user interface and includes items from the feeds to which the user has subscribed. *Id.*, ¶41. Darnell’s webpages present an “ordered list” of

items to the user that is organized either by timestamp or “a score that takes into account both the content of the items and the timestamps of the items” (organizing content data and content applications). *Id.*, ¶¶43-45, 47, 49, 51-58, Figs. 3A-D. A POSITA would have understood that the program operating on server 104 that sends webpages to the remote computer performs functions to create the ordered list.

416. Darnell’s content feeds are broad in scope, covering formats like web pages, images, digital audio, video, and PDF documents, with each item containing details such as titles, descriptions, and URLs. *Id.*, ¶36. The stored data at server system 104 serves as the foundation of content items displayed to users through the feed reader interface. *Id.*, ¶¶33–34, 40, 76, 81, 114. A POSITA would have identified this broad variety of content as “content data,” with each item serving as data stored on the server and accessible through the server’s service provider infrastructure. ’068 Pat., 4:13–15.

417. The user interface in Darnell presents these content items in either a compact or expanded format, with JavaScript enabling client-side tasks, such as marking items as read or updating user preferences, by manipulating the HTML or XML coding of the displayed webpage. Darnell, ¶¶30, 41, 53-54. The JavaScript programming also enables the rendering and layout of content in compact or expanded views, a feature that a POSITA would have identified as part of the “content application” that accesses, displays, and organizes content data as required.

Id.; '068 Pat., 4:16-18. Darnell's user interface uses "Frames" to partition the webpage and a "JavaScript application" that "receives content items from the server system 104, manages the rendering of those content items at the client, and also performs the client-side aspects of other tasks, such as marking items as read, adding and removing content feed subscriptions, labeling content feed subscriptions and/or individually selected content items, selecting or revising user preferences, and the like." Darnell, ¶41. A POSITA would have understood the compact and expanded views of the content data in the user interface is controlled by the HTML or XML programming of the webpage, including the JavaScript.

418. In Darnell, the use of "Frames" and a "JavaScript application" in the user interface allows for the efficient display, interaction, and organization of content on the client side while relying on server system 104 to retrieve and deliver the actual content items. Darnell's approach of using JavaScript to manage "Frames" enables content to be dynamically loaded, rendered, and refreshed within different sections of a single webpage without the need to reload the entire page, which improves both performance and user experience.

419. Frames are a method for partitioning a webpage into separate, independently scrollable sections, each capable of loading different content. By using frames, Darnell's system can maintain distinct areas within the user interface for various types of information, such as content feeds, control options, and settings,

without disruption to other sections when updates occur. JavaScript plays a crucial role in managing these frames, as it allows the system to asynchronously update specific sections by fetching content from the server without requiring a full-page refresh. This means users can view newly retrieved content or interact with updated data while preserving the context of their existing view.

420. Technically, JavaScript can access the Document Object Model (DOM) of the webpage, which represents the structure and content of the interface. By doing so, the JavaScript application in Darnell is able to control each frame's content individually, programmatically load and display new content items, and perform various client-side functions, such as marking items as read or filtering based on user preferences. This architecture offloads much of the display and interaction logic to the client side, which not only minimizes server load but also enhances responsiveness from the user's perspective.

421. For example, when a user interacts with the interface by marking an item as read or subscribing to a new feed, the JavaScript application registers this action, updates the relevant frame, and sends the request back to the server asynchronously. Darnell's use of JavaScript here is equipped to handle these asynchronous tasks, ensuring that interactions such as adding or removing subscriptions or labeling content are reflected in real-time on the client side without reloading the entire interface.

422. Thus, Darnell's server system 104, housing memory and CPU that execute stored programs to aggregate, organize, and display data from preselected content providers, meets the requirements of claim element 1[b1]. A POSITA would have understood the software displaying the compact and expanded format of the content item is a "content application" able to change the display of the content when selected. *See* '068 patent, 4:16-18 ("Content applications are software applications that accesses display and organized content data [sic], generally from a web-sites."). A POSITA would have also understood these content items and webpage programming as "content data and content applications" because Darnell's user interface contains data accessible from the server software application to display and organize the content.

d. Claim 1, limitation [b2], "and streamed to the master server simultaneously in real time from their respective servers on a master web-page and are viewable from the remote computer devices through the master server on the master web-page"

423. Darnell discloses or suggests the elements recited in claim limitation 1[b2]. Claim 1 specifies that "content data and content applications" are "streamed to the master server simultaneously in real time from their respective servers on a master web-page and are viewable from the remote computer devices through the master server on the master web-page." The '068 patent does not define "simultaneously" or "real time" in the specification; however, a POSITA would have

understood that, in the context of the '068 patent specification and file history, these terms to mean that the content from more than one source is sent to the server when it is available. In other words, these terms signify that the content from multiple sources is transmitted to the server when it becomes available, without significant delay between content sources.

424. For example, in column 1, lines 51-54, the '068 patent describes a cloud-based computer system that “can include the delivery of any kind of content data and/or services in real-time, which extend the capabilities to a remote computing device, typically over the internet.” This description of a cloud-based system means that content updates and transmissions occur with minimal lag, which a POSITA would have recognized to ensure users can access the most current data from multiple sources in real time.

425. Further, in column 4, lines 13-15, the '068 patent refers to “content data” as “data or information that is stored remotely on servers that are generally accessible over the internet/intranet via a service provider servers.” This setup indicates that content is dynamically available and must be readily retrievable by the master server. The expectation of prompt retrieval suggests that content updates from each service provider would appear on the master server concurrently, given the system’s design to integrate various feeds into a cohesive, accessible interface.

426. Moreover, in column 4, lines 16-18, the '068 patent discusses “content applications” as software that “accesses[,] display[s,] and organize[s] content data, generally from a web sites.” This description shows that as content updates on a source server, the master application, which organizes and displays content on the master web page, retrieves this content promptly to ensure users experience minimal delay when viewing updated data.

427. Therefore, a POSITA would have understood “simultaneously” and “real time” as describing a system where content from multiple sources is updated and displayed as soon as it becomes available. The cloud-based architecture, as described in the patent, relies on minimized latency to fulfill its purpose of providing access to up-to-date content across various sources.

428. Darnell's Figure 1, alongside Darnell's description, depicts content items streamed to server system 104 (the master server) from their respective hosts or feed sources 120. Darnell, ¶32. The terms “server” and “host” are frequently used interchangeably and mean the same thing. A “host” of information “serves” that information to other computers; hence the similarity between the two terms. Darnell discloses that “server system 104 accesses content feeds from the feed sources 120” through server 108, which not only “retrieves the content feeds from the feed sources 120” but also provides “an interface between the server system 104 and the client systems 102.” *Id.*, ¶33. Accordingly, Darnell reveals that content from multiple feeds

reaches the server and is then delivered to the user, presented in the user's feed with timestamps organized chronologically. This is evident in Darnell's Figures 3A through 3D, which depict content feeds arriving and appearing sequentially to the user. *Id.*, Figs. 3A-3D, ¶¶51, 109.

429. Additionally, Darnell describes server 108's "feed tracker module 516," which retrieves content items "periodically and/or as content feeds are updated." *Id.*, ¶74. Given that the feed tracker fetches items when they are updated, Darnell's system receives content in "real time," as would have been understood by a POSITA. Darnell reinforces this concept by explaining that "feed items from a content feed source can be considered to be a stream of items," thus conveying the "streaming" nature of these content feeds. *Id.*, ¶31. Accordingly, Darnell's system streams content from multiple sources as updates become available, meeting the '068 patent's claim of content feeds streamed "simultaneously in real time."

430. As further explained above with regard to limitation 1[b1], Darnell enables the user to view content via a feed reader user interface 116. This user interface 116, implemented as a web-based display, includes "Frames" for partitioning content and utilizes "JavaScript" for rendering and organizing content items at the client side. *Id.*, ¶41. When a user scrolls to the end of the visible list of content items, the system triggers a "fetch" command to retrieve additional items from the server, allowing the new content items to be appended to the list as the user

scrolls. *Id.*, ¶112. The content items are thus displayed to the user in a continuous, updated stream, meaning that the server system 104 receives, retrieves, and then displays new content items to the user at the client system (remote computing device) in “real time” as they scroll through the feed. Darnell's Figure 3A exemplifies this, with a scroll bar shown on the user interface webpage for navigating through content. *Id.*, ¶54, Fig. 3A (annotated below).

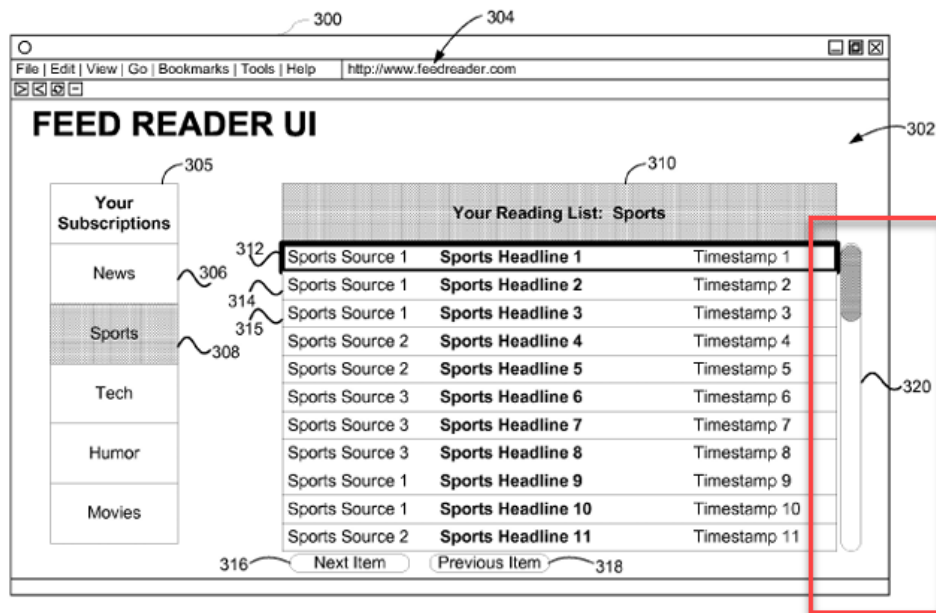


Figure 3A

Darnell, Fig. 3A.

431. A POSITA would have recognized that Darnell’s user interface is a “master web page,” meeting the ’068 patent’s description of a web-based interface that allows for scrolling or paging through “content data and/or content application

from multiple content data sources and/or service providers in real time” (’658 patent, 4:24-26). *Id.*, ¶41.

432. Therefore, Darnell’s cloud-based system discloses streaming content from multiple feed sources 120—wherein multiple content sources are “streamed... simultaneously... from their respective servers”—and sent to server system 104 (the master server). The content is rendered viewable through the feed reader user interface webpage, which features a scroll bar on a client system (remote computing device). Darnell’s feed reader user interface webpage is a “master web page” that presents content “simultaneously in real time” as updates occur.

- e. **Claim 1, limitation [b3], “wherein the master application programs a processor on the remote computer devices to operate and execute a slide tool program for scrolling or paging through the content data or the content applications from the multiple content sources or service providers simultaneously and through the master server in real time from the master web-page and from the remote computer by operating the slide tool program from the remote computer; and”**

433. It is my opinion that Darnell discloses or suggests this limitation. Darnell’s feed reader UI 116 is received from server system 104 executing modules and data structures (the master application) as a program (e.g., webpage) for execution on the processor (programs a processor) of the client system(s) (the remote computer devices). The ’068 patent describes “a slide tool [that] scrolls or pages through content data.” ’068 patent, 4:35-37. The claim does not limit the type of

slide tool that must be used; it only requires that the tool be suitable “for scrolling or paging through” content. A POSITA would have understood that “a slide tool program” is software implementing a slide tool, and a POSITA would have further recognized that this term includes a scrollbar. At the time of invention, scrollbars were widely known and commonly used in web interfaces as tools that allow users to navigate through displayed content.

434. Darnell’s disclosure supports this understanding, describing a feed reader user interface that includes a scrollbar 320 used to navigate an ordered list 310 of content items from various feeds. Darnell, ¶54. This scrollbar is embedded within the user interface webpage and operates as a slide tool that enables a user to scroll through the content in real time as the feed reader dynamically updates and displays content from multiple sources.

435. Darnell also explains that its user interface is constructed in a web-compatible language, such as HTML or XML, and that the webpage includes embedded scripts, such as JavaScript or ECMAScript, allowing the interface to interact with the server system. *Id.*, ¶41. These scripts, embedded in the webpage served from the server system 104, are executed when the webpage is rendered. Specifically, the feed reader module 420 script within the HTML enables a user to scroll through the content items served from the server. *Id.*, ¶67. This architecture is like the architecture described in the ’068 patent, where the master server

“programs” the client-side device to display the feed reader user interface, which includes a scroll bar allowing continuous, real-time access to updated content.

436. Moreover, as Darnell explains, the scrollbar’s function operates from the client system, enabling the user to interact with the feed interface as updated content is dynamically loaded and displayed. Darnell’s server architecture programs the remote client device to interact with the feed content on the master webpage and provides a scrolling tool for viewing multiple content sources through the master server in real time. Accordingly, Darnell’s feed reader user interface, delivered by the server to the client device, includes a scrolling tool that allows for real-time navigation of aggregated content that meets the claim limitation 1[b3].

- f. **Claim 1, limitation [c1], “a remote tool box feature application that allows code to be written, selected or deleted to enable and disable tool features that control how sub-sets of the content data or the content applications from the multiple content sources or service providers are received, displayed or organized,”**

437. Darnell discloses or suggests a “remote tool box feature application that allows code to be written, selected or deleted to enable and disable tool features that control how sub-sets of the content data or the content applications from the multiple content sources or service providers are received, displayed or organized.” The ’068 patent defines a toolbox as “a set of sub-features...that allow a web-page layout and/or operation to be customized by linking tools with content applications.” ’068

patent, 4:28-31. Darnell similarly discloses that the feed reader interface includes a JavaScript application, which manages the rendering of content items and “performs the client-side aspects of other tasks, such as marking items as read, adding and removing content feed subscriptions, labeling content feed subscriptions and/or individually selected content items, selecting or revising user preferences, and the like.” Darnell, ¶41. Darnell’s server also includes “a subscriptions database 518 for storing information about the content feed subscriptions of users of the system” and “a user preferences database 526 for storing user-specific information, including user preferences with regard to the user interface.” Darnell, ¶¶75, 78.

438. As shown in Figure 6C, Darnell provides a “data structure for user preferences,” which is stored in a table format or through sparse table storage mechanisms. Darnell, ¶¶84, 86. A POSITA would have understood that these data structures and databases in Darnell would be implemented in a database language such as SQL.

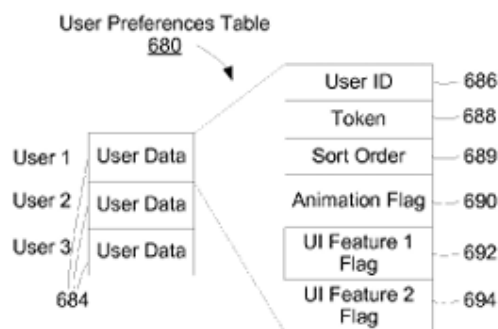


Figure 6C

Darnell, Fig. 6C

439. Accordingly, Darnell’s system discloses a subset of features in the user interface (a remote tool box feature) that allows users to add (write) or modify (delete) user preferences in a database (code), which enables the customization of user interface functionalities. These preferences include fields like Animation Flag 690, UI Feature 1 Flag 692, and UI Feature 2 Flag 694, which can control aspects such as “formatting and color of a user interface,” “sort order,” and “interface feature flags.” Darnell, ¶85. A POSITA would have understood that these user preferences interact with the server-side application to modify what content items are displayed and how they are displayed, thus enabling and disabling tool features that control the reception, display, and organization of content data or applications from multiple sources.

440. Darnell also discloses that users can “create filters or labels to be applied to content feeds and/or content items.” Darnell, ¶40. These filters and labels are transmitted to the server system and written in the database, which, as explained above, corresponds to writing code as claimed because a user creates (writes) the filters and labels. Darnell, ¶40. This interpretation is consistent with the plain and ordinary meaning of the claim term, which Patent Owner seems to agree with based on Patent Owner’s infringement contentions. EX1014. I do not take any position with respect to written description or indefiniteness issues related to this claim limitation.

441. For example, as shown in Figure 6D, Darnell discloses the ability for users to write labels and apply checkmarks to certain content feed items within a “user subscription table.” Darnell, ¶¶87-90, Fig. 6D. The labels created by the user are stored in the subscriptions database on the server, and the server application subsequently utilizes these labels to construct the webpage, which displays them in the feed reader interface. The feed subscriptions (e.g., Feed ID) and labels written by the user (writing code) and stored in the subscriptions database are used by the server application to construct the webpage using those labels. Darnell, ¶51.

User Subscription Table
625

640	Stream 1 630	Stream2 632	Stream3 634	Stream4 636	Stream5 638	
642	Feed ID: URL1	Feed ID: URL2	Feed ID: User1:tech	Feed ID: User2:friends	Feed ID: User25:tech	...
User 1	√ label:tech	√ label:tech				
User 2			√ label:friends		√ label:friends	
User 3	√			√		
644 646	⋮					

Figure 6D

Darnell, Fig. 6D.

442. Darnell further explains, for instance, that “Feed reader user interface 302 includes a list 305 of labels, also referred to as folders, associated with content feeds to which a user has subscribed.” Darnell, ¶51. As illustrated in Figure 3A, these

labels are displayed on the webpage and can be selected to adjust the display of the ordered list of content items. *Id.*, ¶51. I've annotated Figure 3A below to highlight the labels.

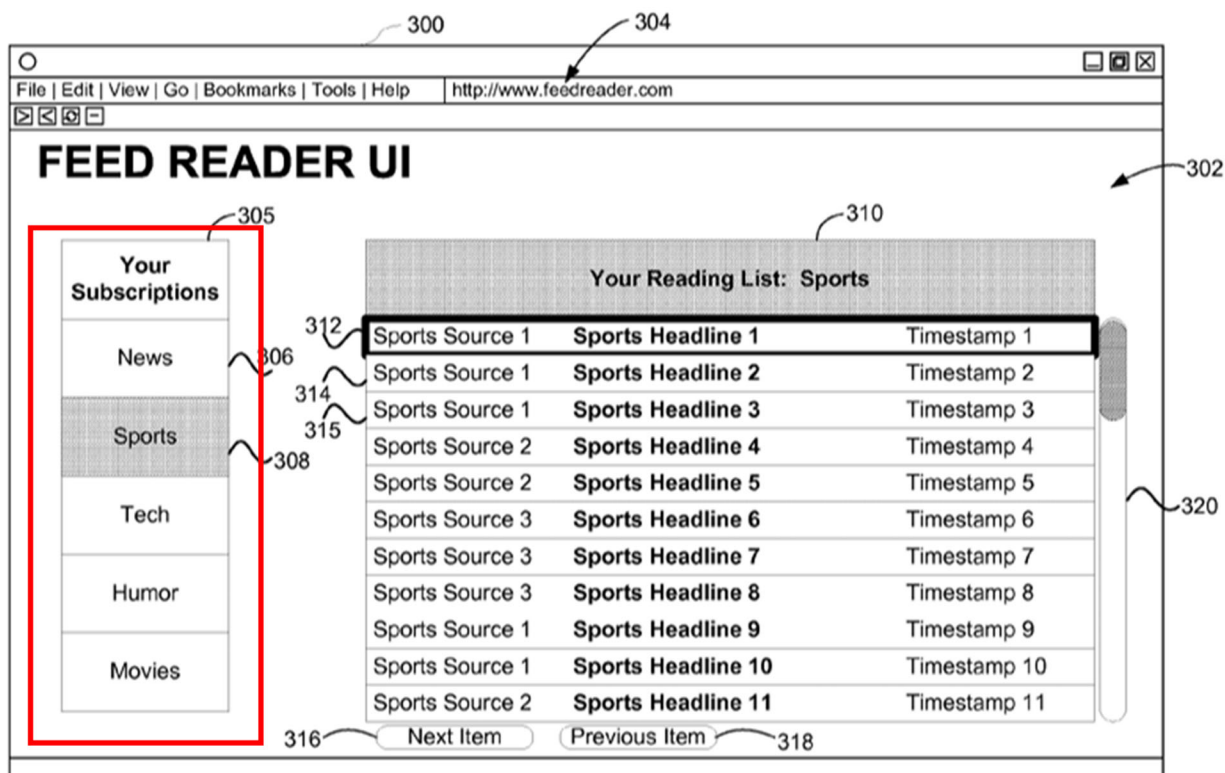


Figure 3A

Darnell, Fig. 3A.

443. Accordingly, Darnell discloses a user interface webpage with JavaScript applications (remote tool box feature application) allowing users to create and modify feed labels and user preferences (write, select, or delete code to enable or disable tool features) that alter the content and display of the user interface webpage (that control how sub-sets of the content data or the content applications

from the multiple content sources or service providers are received, displayed or organized).

- g. Claim 1, limitation [c2], “wherein the master server collects analytics based which tool features are enabled to target specific products, services or information to the user.”**

444. In my opinion, Darnell discloses or suggests the claimed “analytics” functionality that collects user data based on the enabled tool features to tailor the display of products, services, or information to the user. Specifically, Darnell’s system includes a user preferences database 526, stored on the server (the master server), which accumulates user-specific information for purposes of customization. Darnell, ¶84. The user preferences table 680 illustrated in Figure 6C contains fields like user data 684, user ID 686, and token 688, which authenticate and identify user-specific activities. *Id.* The preferences stored here also include a sort order flag 689, which defines the order in which content items are presented to the user—whether sorted chronologically or by a ranking based on quality and timeliness of content. *Id.*, ¶85 (“[T]he user data optionally may include a sort order flag 689, for specifying the order in which to display content items on a user interface, such as user interface 302 (FIG. 3A).”). These customization fields and flags collect data on the user’s interactions with the interface, which a POSITA would have recognized as analytics. I’ve annotated Figure 6C of Darnell below to show the User Data 684 that is part of the user preferences table 680.

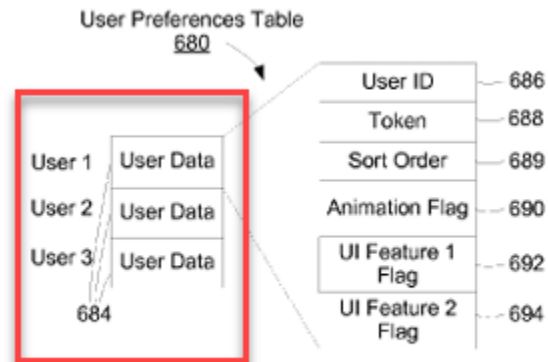


Figure 6C

Id., Fig. 6C.

445. Further, Darnell’s user preference database provides information about what specific content types the user accesses, allowing for targeted recommendations. By capturing data related to the “recency” or “quality” of content items, Darnell’s server gathers insights about the user’s preferences and interest patterns, which can guide the system’s targeted display. *Id.*, ¶51. User preferences and labels (enabled tool features), for example, determine the displayed content. *Id.*, ¶¶51-52. Thus, Darnell’s user preferences database 526 collects information about the user and content (analytics) and uses that information to display content items based on the user’s preferences (enabled to target specific...services or information to the user). A POSITA would have understood that these characteristics—such as recency and quality—constitute analytic data since they provide insights from the data about user interaction patterns and preferences. A POSITA would have

understood that keeping track of content quality and freshness of the content corresponds to “collecting analytics” as claimed.

446. Darnell also discloses the enabled tool features on content presentation. As discussed in ¶85, the user preferences that Darnell collects impact the presentation of content items, allowing for a personalized user experience based on preferences stored in the server database. These preferences include flags like Animation Flag 690, UI Feature 1 Flag 692, and UI Feature 2 Flag 694, which can be used to adjust aspects such as “formatting and color of a user interface,” sort order, and other interface settings. These flags act as data points regarding which features are enabled by the user and, thus, are “analytics” collected by the server to determine the user’s preferences. *Id.*, ¶85.

447. In Darnell, the word “regency” appears to be a typographical error and, instead, should be “recency.” “Recency” of data means how recent a user accessed that data. A POSITA would have understood that more recently accessed content is likely more relevant to a user than content that has not been recently accessed by a user.

448. In Darnell’s system, the notion of a “score” that “takes into account both the content of the items and the timestamps of the items” (Darnell, ¶51) could incorporate a type of regency where content is evaluated based on predefined criteria that might include the importance of the content in the overall context, its value to

the user, or its weight within the aggregation structure. This scoring system could prioritize or order content not only by how recent it is but also by how “regent” it is according to the system's prioritization metrics.

449. To the extent it might be argued that Darnell alone does not disclose all aspects of the analytics functionality, Kim provides a complementary teaching that would render this element obvious. Kim’s system collects and uses a user’s *usage history*, which tracks the user’s interaction with content feeds—such as how often each feed is accessed and the read-to-unread item ratio. Kim, ¶¶69-70. A POSITA would have recognized that tracking a user’s page interactions, such as visits and read/unread status, constitutes analytics since these actions reflect analytic data commonly collected to understand and optimize user engagement.

450. A POSITA would have understood “analytics” as a process involving the collection, examination, and interpretation of data to derive meaningful insights, particularly in enhancing and personalizing the user experience. In the context of the ’068 patent, this would extend to tracking a user’s interactions with specific features and tools to adapt content presentation dynamically based on user behavior.

451. For example, as the ’068 patent describes, users have options to configure their content experience by enabling or disabling features, which then influence the organization, display, and prioritization of content data on the interface. Tracking these configurations and user interactions provides a feedback

loop—analytics—on user preferences, which can then inform subsequent content organization and feature customization, targeting specific content items, services, or applications to the user based on inferred interests.

452. Thus, “analytics” here would have been understood to refer to the way data on user interactions is aggregated and assessed to provide actionable insights that improve or tailor the content delivery system. This is consistent with definitions of “analytics” as “information resulting from the systematic analysis of data or statistics” as found in the Google search definition of “analytics” (<https://www.google.com/search?q=definition+of+analytics>), which relies on definitions from Oxford Languages and other dictionaries and reflects how a POSITA would have recognized the importance of tracking user engagement to optimize system responses and interface functionality. Other dictionary definitions concur:

453. **Oxford English Dictionary (OED):** “[A]nalytics” refers to “the systematic computational analysis of data or statistics.” EX1047. A POSITA would have understood this definition to include the computational tracking of user content views and device usage on the server, which Darnell and Kim disclose as methods to tailor content delivery to user interactions.

454. **Cambridge Dictionary:** “[A]nalytics” is defined as “the process in which a computer examines information using mathematical methods in order to

find useful patterns.” EX1046. This definition suggests that analytics involves gathering historical data on content views and device usage patterns, as seen in Darnell’s user account tracking and Kim’s usage history feature, to tailor content presentation to user behaviors.

455. Furthermore, Kim organizes content based on *traits* that users can select, as detailed in ¶58 and depicted in Figure 10. Traits serve as indicators of content type (e.g., images, text, links), and the user’s engagement with these traits provides insight into preferred content categories. Kim, ¶58. A POSITA would have understood that these content traits qualify as tool features under the ’068 patent’s description, since traits indicate specific content attributes, allowing the system to tailor presented information based on user engagement. A POSITA would have understood a user’s selection and viewing of the content items is determined by the selected traits.

456. Kim’s Figure 13 presents a further example of user-driven content organization, where summaries of content feeds are ordered based on factors such as frequency of user engagement and other interaction-based metrics. Kim, ¶¶67-71. Here, Kim’s system presents headlines in a particular order based on a user’s history with the content—an example of content being targeted to the user based on collected analytics. *Id.*, ¶69. For example, the feed application “presents headlines for content items in chronological order for the five feeds with the strongest usage

history in region.” Thus, the order of the content feeds in Figure 13 are organized based on a user’s usage history (to target specific...information to the user). A POSITA would have applied Kim’s teachings to Darnell’s ordered list system to create a more refined sorting methodology based on traits and engagement history.

457. Given the analogous teachings in Darnell and Kim, a POSITA would have been motivated to apply Kim’s tracking of usage history and traits to Darnell’s user preference and display systems, expecting to achieve enhanced customization of content displays based on user-specific data. For example, a POSITA would have understood that Kim’s usage history (master server collects analytics) is based on which content and traits a user selects and views (based which tool features are enabled), which then determines, for example, the order of the summary content feeds (target specific...information to the user) and applied Kim’s teaching to Darnell. Darnell discloses the “items in ordered list 310 are ordered in accordance with a score that takes into account both the content of the items and the timestamps of the items.” Darnell, ¶51. Darnell, however, does not disclose how the score is calculated or what aspects of the content are “tak[en] into account.” But Kim discloses “analyz[ing] a feed's content for its traits... such as whether or not the feed content is primarily links, text, inline photos, enclosed media, and the like.” Kim, ¶¶25, 38, 58. For the reasons I explained in § XII.A.1, a POSITA would have been

motivated to apply Kim’s analysis of content traits to Darnell’s “score” system in order to better define how Darnell’s “score” is calculated.

458. A POSITA would have had a reasonable expectation of success in using Kim’s traits and usage history to control how Darnell displays feeds based on scores because Kim shows that using such a technique to organize displayed feeds was known by a POSITA. Furthermore, organizing and presenting content to a user based on previous viewing of the content would have been a straightforward application of known, conventional content aggregation methods used according to their known functions to yield predictable results. Applying this approach would have been a routine exercise in combining known techniques for user analytics, leading to predictable improvements in delivering targeted content to the user.

B. GROUND 2: DASGUPTA AND DARNELL RENDER CLAIM 1 OBVIOUS

459. In Ground 1, I applied Darnell as the primary reference. Here, in Ground 2, I will apply Dasgupta as the primary reference. In my opinion, Dasgupta in view of Darnell renders claim 1 obvious for the reasons that follow.

1. Motivation to Combine Dasgupta and Darnell

460. A POSITA would have been motivated to extend Dasgupta to incorporate Darnell’s teachings. A POSITA would have recognized that Darnell’s system for aggregating content—such as news and sports on a cloud-based server—would be compatible with Dasgupta’s cloud-based media content access system

because Darnell and Dasgupta each supports content aggregation and presentation. Darnell, Abstract; Dasgupta, Abstract. Moreover, Both Darnell and Dasgupta aggregate content at a server and send the aggregated content to a remote client. Darnell, ¶¶33-34, Fig. 1; Dasgupta, 6:52-61, 7:44-55, Fig. 2. A POSITA would have recognized the efficiency and functionality gained by combining these teachings because Darnell's framework centers on collecting and organizing different content types on a server before presenting them to users through a unified interface (Darnell, Abstract, ¶¶11-14, 30-32, Fig. 1), and Dasgupta similarly describes a system where diverse media content (e.g., digital video and audio) is aggregated and streamed from a server to the client, providing users with a unified graphical user interface (GUI) for interaction. Dasgupta, ¶¶7:3-67, Fig. 2. Darnell's known techniques would have improved Dasgupta because they share similar architectural models and combining their features would provide a more versatile and comprehensive media content aggregation system in similar ways. A POSITA would have also found these references' common reliance on Internet-based protocols, specifically for content streaming and organization, as a motivation to combine them. Darnell, for instance, uses RSS feeds and similar protocols to organize and stream digital content (Darnell, ¶¶8, 32, 63, 73, 96), while Dasgupta also relies on Internet protocols to deliver content over HTTP, facilitating dynamic user interaction with various content types through the GUI. Dasgupta, ¶¶7:3-29, 10:10-

24, 12:48-62. This technological overlap means that combining Darnell and Dasgupta's systems to streamline and enhance the management and display of content would have been an obvious and straightforward application of known media content delivery techniques over the Internet.

461. A POSITA would have recognized that integrating these systems provides considerable advantages. Specifically, Darnell's methods for managing feed subscriptions and authenticating users complements and enhances Dasgupta's existing media access capabilities. Darnell discloses a system that enables users to select specific content feeds, securely access them, and personalize their content viewing experience according to their preferences, as reflected in Darnell's subscription and feed reader structure. This structured selection of content allows users to focus on chosen sources while preventing unauthorized access to sensitive or private information. Darnell, ¶¶40, 84, 99-102. Integrating these capabilities into Dasgupta's system would provide users with a more organized, user-focused interface while leveraging Darnell's robust selection and security functionalities to ensure that only authenticated, authorized users can view selected content.

462. Dasgupta's media content system, in turn, is designed to provide extensive access to a wide range of media channels and sources, and explicitly supports customization for the presentation and control of media content access using "any technologies suitable for accessing media content." Dasgupta, 3:14-28.

By adopting Darnell's subscription techniques, Dasgupta's system would benefit from displaying content specifically tailored to each user's subscriptions and preferences, rather than presenting the entire breadth of available content without distinction. A POSITA would have recognized that Dasgupta's "access facility" could readily accommodate Darnell's selection and security features, adding value by offering a more personalized media experience that aligns with individual user profiles.

463. Furthermore, combining Darnell's feed selection and authorization methods with Dasgupta's system would have been well within the knowledge of a POSITA. Systems that organize and restrict access based on user authentication and predefined subscriptions were well-understood in the field at the time. Therefore, the combination would be a straightforward application of these known methods to yield predictable results, offering users a streamlined and customizable interface. Incorporating Darnell's methods would merely apply recognized techniques for enhancing control over content access and personalization to the content management features in Dasgupta, resulting in a unified and secure media access system without introducing undue complexity. A POSITA would have had a reasonable expectation of success in using Darnell's feed selection methods in Dasgupta's media content access system because Dasgupta acknowledges that any method may be used, and Darnell shows that using its technique controls which

content sources are displayed in the user interface. Dasgupta, 3:14-28. Furthermore, creating content lists with restricted access would have been a straightforward application of known, conventional content aggregation methods used according to their known functions to yield predictable results.

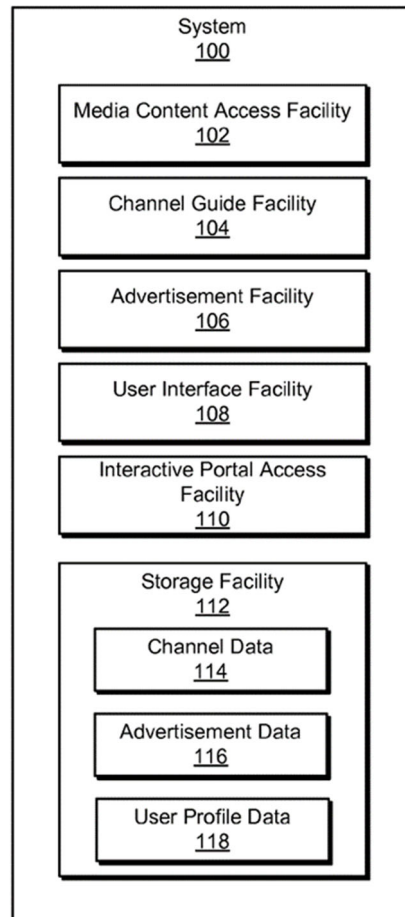
2. Independent Claim 1

a. Preamble of claim 1, “A cloud-based computer system comprising:”

464. To the extent it is limiting, the preamble of Claim 1, “a cloud-based computer system,” is disclosed or suggested by Dasgupta’s media content access system, which comprises various components for accessing, delivering, and presenting media content from multiple sources through network connections such as the internet or intranet. A POSITA would have recognized that Dasgupta’s media content access system is a cloud-based computer system, as required by the claim.

465. Dasgupta’s disclosure describes a comprehensive media content access system configured for retrieving and delivering media content, supported by both software and hardware elements. Dasgupta explains that system 100 includes components like media content access facility 102, channel guide facility 104, advertisement facility 106, user interface facility 108, interactive portal access facility 110, and storage facility 112. Dasgupta, 3:4-13. Each of these facilities contributes to the system’s overall function of providing and managing content accessible through a centralized networked environment, qualities intrinsic to a

cloud-based system. Figure 1 of Dasgupta effectively illustrates this architecture, emphasizing the interactive and content-rich nature of the system, similar to what is expected from cloud-based implementations.



Id., Fig. 1.

466. Moreover, the system's configuration allows media content access facility 102 to retrieve and aggregate content from various providers, making it accessible to end-users through Dasgupta's user interface facility 108. This architecture is facilitated by components like the media content delivery subsystem 202 and media content access subsystem 204, which communicate over network

206, enabling delivery and access across a variety of network types, including the internet or intranets. Dasgupta, 7:3-19. This design, involving a “network cloud” as depicted in Figure 2, is a cloud-based computer system, where content is managed, processed, and transmitted through cloud-network infrastructures.

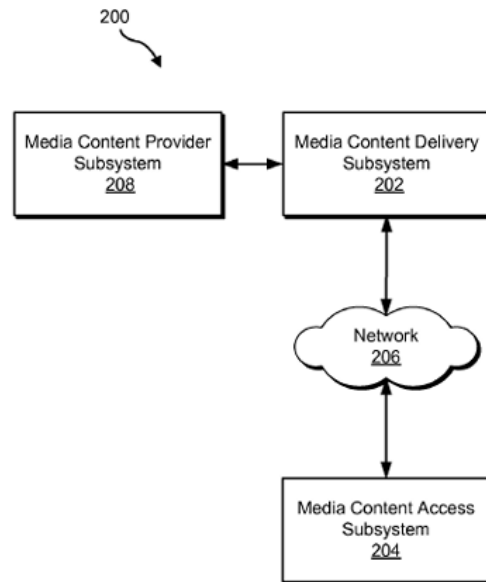


Fig. 2

Id., Fig. 2.

467. The access facility “process[es] and present[s] media content for experiencing by a user,” whereas the channel guide “maintain[s] channel data representative of one or more media content” accessible by the access facility and

stored as channel data in the storage facility. *Id.*, 3:14-51. The UI is “configured to generate and launch one or more GUIs for display to a user.” *Id.*, 4:33-44.

468. Dasgupta further emphasizes that the system’s software is responsible for generating and providing GUIs (graphical user interfaces) to users, including end-user devices, for interacting with media content. *Id.*, 2:64-3:3. The ability of Dasgupta’s media content access system to generate and distribute GUIs demonstrates its role as a cloud-based environment, as it supports centralized content delivery, user interaction, and control across remotely connected devices. This configuration discloses the requirements of a cloud-based computer system as specified in Claim 1.

469. A POSITA would therefore have understood that Dasgupta’s media content access system qualifies as a “cloud-based computer system” as described in the preamble of Claim 1.

b. Claim 1, limitation [a], “master server with a processor and memory, wherein the master server accessible over the internet/Intranet from remote computer devices by a user”

470. In my opinion, Dasgupta discloses or suggests this limitation 1[a]. As detailed in Dasgupta, the media content access system operates through one or more computing devices, which include “a processor (e.g., a microprocessor)...from a computer-readable medium (e.g., a memory, etc.).” Dasgupta, 6:27-51, 12:15-26.

This description corresponds to a master server containing both a processor and memory, as claimed.

471. Dasgupta's delivery subsystem 202 includes a processor 304 and storage device 306, such as memory or storage drives. Figure 3 of Dasgupta illustrates this delivery subsystem, which is designed to manage operations for the media content access system. Subsystem 202 performs functions, including data processing and storage, which, in my opinion, meet the claimed "master server with a processor and memory." Dasgupta, 7:56-8:67. I've outlined the relevant components of subsystem 202 below:

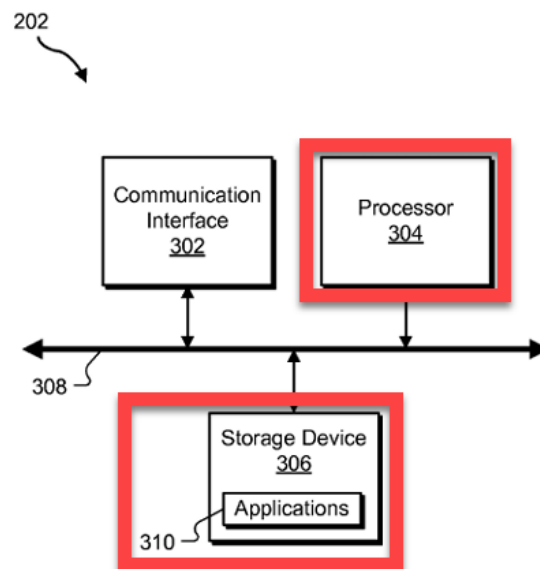


Fig. 3

Id., Fig. 3.

472. Further, Dasgupta explains that “any or all of the facilities described” may be implemented within delivery subsystem 202, which encompasses functionalities such as media content access, channel guide management, and advertisement handling. Dasgupta, 8:54-67.

473. Additionally, Dasgupta discloses a second subsystem, subsystem 204, which contains a processor 404 and storage device 406. This subsystem may be implemented on remote computing devices, including “a personal computer,” and facilitates user access to content. *Id.*, 9:1-18, Fig. 4. Dasgupta further explains that subsystem 204 receives content from subsystem 202 over network 206, which includes internet and intranet connectivity options. Dasgupta, 7:3-43. In my view, this arrangement establishes subsystem 202 as the “master server” that is accessible from remote devices (e.g., subsystem 204) via the internet or intranet, thereby fulfilling the requirements of Claim 1[a].

474. Based on the description provided in Dasgupta, a POSITA would have understood delivery subsystem 202, equipped with processing and memory capabilities and communicating over network 206, to perform as the claimed “master server with a processor and memory, accessible over the internet/intranet from remote computer devices by a user.”

- c. **Claim 1, limitation [b1], “a master application stored in the memory of the master server and that executes on the processor of the master server and that includes software coded for organizing content data**

**and content applications from multiple content
sources or service providers pre-selected”**

475. In my opinion, Dasgupta discloses or suggests this limitation 1[b1]. Dasgupta describes “one or more applications 310” stored in storage device 306 within subsystem 202, which contains processor 304. As shown in Dasgupta’s Figure 3, applications 310 are housed in memory on the master server and are “configured to direct processor 304 to perform any of the operations described herein,” including functions associated with the access, guide, and advertisement facilities. Dasgupta, 8:28-67. These applications direct processor 304 to perform tasks integral to the system, establishing that Dasgupta’s system includes a “master application” capable of organizing content in a manner consistent with the claim requirements. I annotated Figure 3 below to show Applications 310 (the master application):

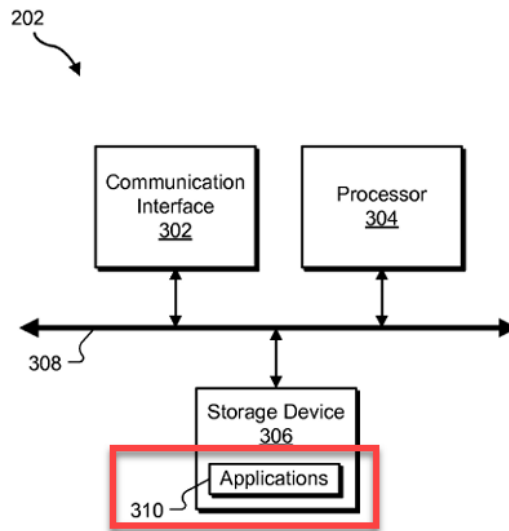


Fig. 3

Id., Fig. 3.

476. “[A]pplications 310 [are] configured to direct processor 304 to perform any of the operations described herein” and “may be configured to direct processor 304 to perform one or more processes or functions associated with” the facilities as described in 1[pre] (executes on the processor of the master server). *Id.*, 8:39-67. Dasgupta also explains that channel guide facility 104, advertisement facility 106, and user interface facility 108 interact to arrange and present content to users, particularly through a graphical user interface (GUI) displaying a channel guide with icons representing various media content channels. *Id.*, 1:64-2:24, 4:33-44, 13:9-23, 17:43-63. In addition to representing channel guide and advertisements in this UI,

Dasgupta describes how this interface serves as a centralized platform for accessing content. These icons are configured as launching points to external interactive portals, providing users access to diverse content sources directly within the system. *Id.*, 2:17-24, 4:24-32, 13:24-40, 13:24-30. The process of assembling channel icons and advertisements into a unified, organized structure meets the claim's requirement for "organizing content data and content applications from multiple content sources or service providers."

477. Dasgupta's storage facility 112 within subsystem 202 holds channel and advertisement data, which organizes and manages content within the GUI. Dasgupta, 3:63-4:8, 6:7-12, 8:54-67, 13:41-14:44. In my opinion, Dasgupta's channel icons, serving as representations of multiple media content sources and stored in subsystem 202, would have been understood by a POSITA to represent "content data and content applications," providing the building blocks for the organized, user-friendly UI display specified by the claim.

478. Dasgupta's Figures 7 and 16 provide visual examples of the organized display of a channel guide featuring "a plurality of channel icons associated with a plurality of media content channels" originating from multiple content providers, such as the inclusion of "HBO." Dasgupta, 17:43-63, Figs. 7 & 16.

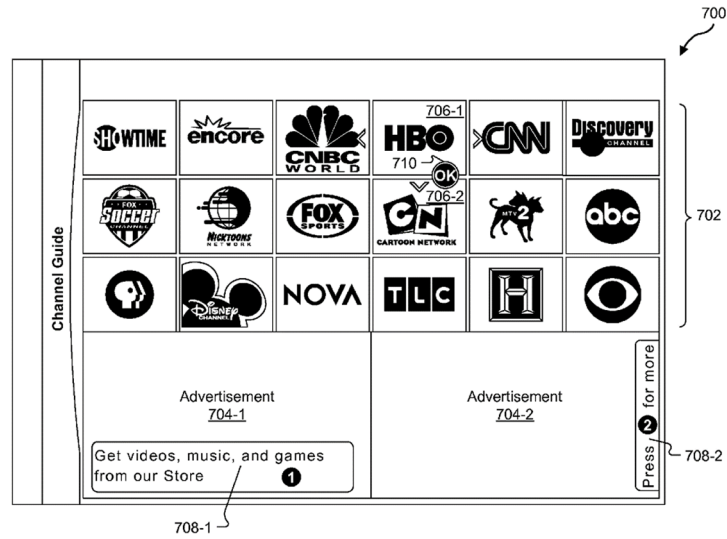


Fig. 7

Id., Fig. 7.



Fig. 16

Id., Fig. 16.

479. Regarding the phrase “pre-selected,” Dasgupta explains that channel guides contain content accessible only by “subscribers,” suggesting user selection.

In describing a user as a “subscriber to one or more services,” Dasgupta discloses that such services, including premium channels like HBO, require a user-initiated selection and purchase. Dasgupta, 2:64-3:3, 7:3-19, 14:3-30. A POSITA would have understood that a user must actively subscribe to certain channels, which are then displayed in the channel guide, meeting the requirement that these content sources are “pre-selected.” It was well-known that HBO is a subscription channel that can only be viewed if it is chosen and paid for by the user, such as in a package that contains the channel. A POSITA would have also understood that the content from a subscription channel, such as HBO, would be purchased by the user before it could be presented to the user in the UI.

480. Additionally, Dasgupta specifies that the UI system can “populate a GUI with content,” modify it by “adding new content,” and subsequently organize content within the GUI according to user selections and interactions. Dasgupta, 3:14-28, 4:33-5:5. Dasgupta discloses that the access facility “may be configured to access, process, and present media content for experiencing by a user,” using “any technologies suitable for accessing media content.” *Id.*, 3:14-28. Dasgupta’s UI may also be “configured to control the contents of a GUI.” *Id.*, 5:6-19. For example, it “may initially populate a GUI with content such as a graphical representation of a channel guide” and “[s]ubsequently...may modify the contents of the GUI such as by adding new content to the GUI, removing content from the GUI, modifying

content in the GUI, and replacing content in the GUI with new content.” *Id.* Furthermore, the UI accepts user input to control the GUI. *Id.*, 5:20-40. A POSITA would have understood that a user is operating the UI and selecting the content to be received.

481. Dasgupta further clarifies how the system allows users to navigate through the UI in both “navigation mode” and “launch mode,” where users select specific media content channels such as HBO. For example, as the user navigates and highlights a channel icon, content relevant to that icon is displayed and updated, allowing for preselection and engagement with the chosen content. *Id.*, 14:45-16:26, Figs. 7-9.

482. To the extent PO argues that “pre-selected” would not be obvious in view of Dasgupta alone, POSITAs would have been motivated to incorporate Darnell’s feed subscription methods into Dasgupta’s access facility to enhance user interaction and organization of media sources. In Darnell, the system presents subscribed feeds through the user interface, filtering content according to a list created by the user. By applying Darnell’s feed selection and management to Dasgupta’s system, the user could preselect and organize specific content sources within the UI, enhancing the user experience. *Id.*, 3:14-28; Darnell ¶¶40, 87-90.

483. For the reasons I explained in § XII.B.1, a POSITA would have been motivated combine Darnell with Dasgupta. In my opinion, incorporating Darnell’s

method would be a predictable application of known techniques to improve the organization and customization of content in Dasgupta's media content access system. This enhancement would provide the expected results of displaying only the preselected content sources based on user subscription, while maintaining a straightforward application of conventional aggregation methods.

484. In my opinion, Dasgupta alone discloses this limitation or Dasgupta in combination with Darnell discloses this limitation.

- d. Claim 1, limitation [b2], “and streamed to the master server simultaneously in real time from their respective servers on a master web-page and are viewable from the remote computer devices through the master server on the master web-page”**

485. In my opinion, Dasgupta discloses or suggests this limitation 1[b2]. Dasgupta describes how provider subsystem 208 transmits media content to delivery subsystem 202, which is then distributed to media content access subsystem 204 via network 206. *Id.*, 7:3-19, Fig. 2. Provider subsystem 208 includes sources operated by media providers, such as HBO, configured to send media data to subsystem 202, thereby establishing a direct streaming of content from these sources to the master server. *Id.*, 7:3-19.

486. In the context of cable systems, POSITAs would have understood that content transmitted over such systems relies on channels organized by distinct frequency bands. Wideband signals carry numerous channels simultaneously, each

differentiated by its assigned frequency. A receiver device “tunes” to a specified frequency to isolate and present the content associated with that frequency, such as specific channels on cable systems. Dasgupta describes a similar arrangement where media channels are defined by frequency bands, with subsystem 202 configured to access these “channels” based on frequency availability. *Id.*, 2:25-50, 3:14-28, 9:25-10:24, 12:40-13:4. Given that each frequency band corresponds to a separate content stream available concurrently across all channels, a POSITA would have understood that content could be delivered “simultaneously in real time” from these respective sources to the master server.

487. Dasgupta also discloses that subsystem 202, in conjunction with network 206, delivers content from multiple providers to subsystem 204, which operates as the remote client device. To facilitate this communication, Dasgupta explains the use of the Hypertext Transfer Protocol (HTTP) within the Transmission Control Protocol/Internet Protocol (TCP/IP) framework to manage data transmission. *Id.*, 7:20-29. HTTP is the foundational protocol for the World Wide Web, defining how data, including web pages, is transmitted from servers to client devices. Using HTTP, subsystem 202 can provide web-based content aggregation directly to subsystem 204 by sending structured data (such as HTML or other markup data) that is rendered on the client device. HTTP governs the interaction between web servers and web clients, such as browsers, allowing users to access and

interact with content over the internet, which is particularly relevant when web-based interfaces serve as content aggregation platforms. *See, e.g., Darnell*, ¶¶41, 96, 102. A POSITA would have thus recognized that subsystem 202, acting as the master server, organizes and distributes content as web pages viewable on subsystem 204 over HTTP, creating a “master web-page” accessible on remote devices through the master server. At the time of the alleged invention, it was well known that webpages included HTML and JavaScript programs to create user interfaces. A POSITA would have understood that subsystem 202 sends webpages to subsystem 204 using HTTP (i.e., “on a master web-page”).

488. Additionally, Dasgupta explicitly describes how media content becomes viewable on subsystem 204 (the remote computer device), which “present the media content for experiencing (e.g., viewing) by a user.” *Id.*, 7:36-43. Subsystem 202 serves as a central distribution point for content received from provider subsystem 208, thereby ensuring that the aggregated media content is accessible for viewing on subsystem 204. In my opinion, a POSITA would have thus understood Dasgupta’s system as streaming content in real time to the master server, accessible from a master web-page and ultimately viewable on remote devices through that server-based interface. This understanding is reinforced by Dasgupta’s disclosure of subsystem 202 as a central server facilitating the distribution and

display of media content on user-accessible interfaces, hosted and processed centrally and streamed to client devices for remote viewing.

- e. **Claim 1, limitation [b3], “wherein the master application programs a processor on the remote computer devices to operate and execute a slide tool program for scrolling or paging through the content data or the content applications from the multiple content sources or service providers simultaneously and through the master server in real time from the master web-page and from the remote computer by operating the slide tool program from the remote computer; and”**

489. In my opinion, Dasgupta discloses or suggests this limitation 1[b3]. As explained above, Dasgupta provides a method in which user interfaces, such as a channel guide or advertisement display, are transmitted from subsystem 202 to subsystem 204 using HTTP and displayed on a user’s client device. *Id.*, 7:20-29. The ’068 patent discloses that a “slide tool” scrolls or pages through content data and does not limit the form of the slide tool as long as it allows scrolling or paging. ’068 patent, 4:35-37. A POSITA would have understood that a “slide tool program” could include any software program that enables scrolling, whether a scroll bar, arrow key navigation, or similar visual tool in the user interface. This understanding is further supported by the use of JavaScript or HTML to execute and present user interfaces with scroll functionality on client devices as disclosed in Dasgupta. A POSITA would have thus found it obvious that Dasgupta’s user interface system would enable scrolling or paging through content data in a client

device interface using software-based tools like scroll bars or similar prompts. *Id.*, 8:39-67.

490. Dasgupta describes subsystem 202 as hosting the user interface in a way that provides client devices (subsystem 204) with a programmatic user interface. For example, Dasgupta discloses applications 310, stored in storage device 306 of subsystem 202, that configure processor 304 to execute various functionalities, including the graphical user interface (GUI) transmitted to the client computer. *Id.*, 8:27-67, Fig. 3. By delivering the user interface over HTTP, subsystem 202 “programs” the remote device, allowing the remote processor to display and interact with the interface locally, effectively executing the channel guide or advertisement display on subsystem 204. *Id.*, 7:20-29.

491. Dasgupta specifically enables scrolling functionality through its channel guide and advertisement interface. For instance, Dasgupta’s advertisement facility 106 enables users to scroll horizontally through advertisements within the interface. A visual prompt 708-2, as illustrated in Figure 7, allows users to “access more advertisements 704” by initiating scrolling. *Id.*, 14:3-30, Fig. 7. When activated, the prompt scrolls a strip of advertisements “such that one or more of the currently displayed advertisements 704 is moved off-screen... and one or more other advertisements 704 is moved on-screen.” *Id.* This disclosure meets the slide tool program requirement, as it allows a user to view advertisements sequentially by

scrolling within the user interface on the client device. A POSITA would have recognized this functionality as a form of scrolling or paging through content data or applications, since the advertisements are accessed and navigated in real time through the master server interface and viewed on the remote computer device. I annotated Figure 7 below to show the scrolling advertisements in Dasgupta's user interface:

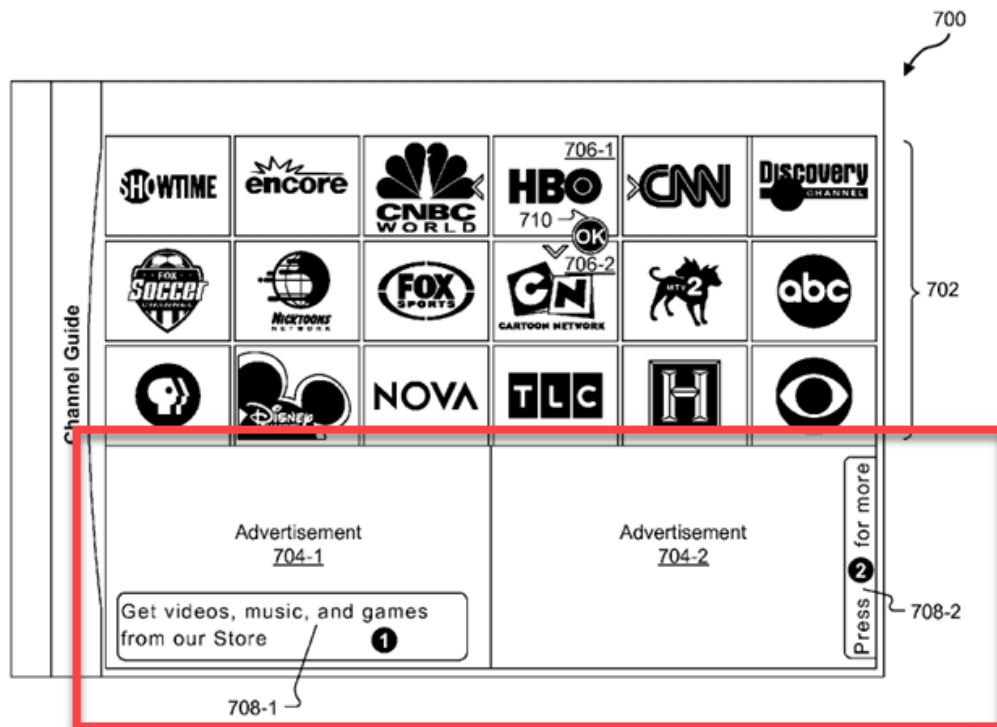


Fig. 7

Id., Fig. 7.

492. In another embodiment, Dasgupta discloses a channel guide 1602, shown in Figure 16. In this configuration, Dasgupta enables a three-dimensional grid of media channels that users can scroll through directionally. When a user initiates

a navigation event, Dasgupta describes how the system “respond[s] by directionally scrolling the three-dimensional grid matrix of channel icons relative to stage 1604.” *Id.*, 18:49-19:3. This channel guide thus provides a similar scrolling experience, allowing users to navigate channel options using the remote interface and view available media content channels from multiple providers in real time. This scrolling action on the remote device interface demonstrates a “slide tool” within the meaning of the ’068 patent, as it allows the user to navigate and select content from multiple providers directly from the master server interface.



Fig. 16

Id., Fig. 16.

493. As outlined above, Dasgupta teaches that such scrolling or paging features may occur on the remote device (subsystem 204) and are executed via programmatic instructions delivered from the master server (subsystem 202) over

HTTP. In my opinion, A POSITA would have understood that these features in Dasgupta disclose or suggest a “slide tool program” that enables scrolling through content sourced from multiple providers in real time via a server-managed interface, aligning directly with the requirements of limitation 1[b3].

- f. **Claim 1, limitation [c1], “a remote tool box feature application that allows code to be written, selected or deleted to enable and disable tool features that control how sub-sets of the content data or the content applications from the multiple content sources or service providers are received, displayed or organized,”**

494. In my opinion, Dasgupta, in combination with Darnell, discloses or suggests limitation 1[c1]. Dasgupta describes the ability of the user interface to enable navigation within a channel guide in “navigation mode,” where a user can browse through channel icons, selecting and highlighting various media content such as “HBO” or “CNN.” *Id.*, 14:45-15:44. The channel icons represent the media content channels in Dasgupta’s system (as discussed for limitation 1[b1]) and would have been understood by a POSITA as corresponding to the content data and applications provided by multiple content sources or service providers. A POSITA would have therefore recognized that this ability to navigate and highlight specific media content meets the “remote tool box feature application” as disclosed in the ’068 patent, which defines the tool box as an application that “allow[s] a web-

page layout and/or operation to be customized by linking tools with content applications.” ’068 patent, 4:28-31.

495. Further, a POSITA would have understood that when a user interacts with the channel guide and highlights a channel icon, additional content specific to the selected icon is displayed on-screen, thus modifying the display of content in real time. In “navigation mode,” Dasgupta discloses that when a channel icon is highlighted, channel content 902 is displayed, “inform[ing] a user of a specific input that may be provided to initiate access to information about one or more deals associated with the channel content.” Dasgupta, 15:45-16:26, Fig. 9. By highlighting a particular channel icon, the system triggers (or “selects”) additional code or program instructions that enable the display of specific content, thereby activating additional functionalities within the UI. A POSITA would have recognized that in selecting a particular content icon and displaying related content, Dasgupta’s system “allows code to be ... selected” to enable specific display features, in line with the requirement for selecting code “to enable and disable tool features that control how sub-sets of the content data or the content applications...are...displayed.”

496. Furthermore, Dasgupta’s system includes an “advertisement facility 106,” configured to personalize and control the display of advertisements within the UI. This facility “analyze advertisement metadata and information about the user” and “select[s] an advertisement that is likely of interest to the user based on the

analysis.” *Id.*, 4:9-23. A POSITA would have understood this feature to function as a “tool” that determines which specific advertisements are displayed based on the user’s data and preferences, dynamically adjusting the advertisement content available to each user. This advertisement facility thus embodies a tool feature that manages which “sub-sets of the content data or the content applications...are received, displayed or organized” on the user interface, as it controls both the selection and organization of advertisements delivered to the user.

497. Accordingly, it is my opinion that a POSITA would have recognized Dasgupta, in combination with Darnell, as disclosing a “remote tool box feature application” that allows user interaction to selectively enable or disable tool features affecting the receipt, display, and organization of various content sub-sets, consistent with the requirements of limitation 1[c1].

g. Claim 1, limitation [c2], “wherein the master server collects analytics based which tool features are enabled to target specific products, services or information to the user.”

498. In my opinion, Dasgupta discloses or suggests limitation 1[c2]. Dasgupta describes an “advertisement facility 106,” which dynamically selects and displays advertisements to users by considering user-specific data. Dasgupta explains that the advertisement facility analyzes both “advertisement metadata and information about the user” to “select an advertisement that is likely of interest to the user based on the analysis.” *Id.*, 4:9-23. This disclosure suggests that the

advertisement facility targets specific products, services, or information to users based on data about their individual preferences or behavior, aligning with the requirement in the claim to “target specific products, services or information to the user.” By using this approach, the system customizes the presentation of advertisements, selecting content tailored to the user’s preferences and data profile. A POSITA would thus have understood that the advertisement facility acts to target specific products and information by analyzing user-related data within the system.

499. Dasgupta also discloses “user profile data 118” stored in the system’s storage facility 112, which includes data relevant to customizing the user experience, such as “one or more preferences of a user, historical media content access data associated with the user (e.g., data indicating one or more media channels and/or media content programs accessed by the user), and any other information about the user that may be helpful for selecting a personalized advertisement for presentation to the user.” *Id.*, 4:9-23. This user profile data is maintained within the storage device 306 on subsystem 202, which a POSITA would have recognized as the “master server” described in claim limitation 1[c2]. By collecting and storing this user profile data, the master server gathers “analytics” regarding user preferences and historical interactions with media content, which allows the advertisement facility to make informed decisions on which products or services to target based on these analytics. *Id.*, 8:54-67. A POSITA would have understood that when a user selects a channel

in Dasgupta’s UI, particular code would be selected and enabled to, for example, “initiate access to information about one or more deals associated with the channel content.” *Id.*, 15:58-65. A POSITA would have further understood that when a user selects channels (i.e., “tool features are enabled”), the selections are stored in the user’s profile data. *Id.*, 4:9-23 (“historical media content access”). That profile data (e.g., showing the television channels to which the user subscribed) allows Dasgupta to provide targeted advertisement data to that user (target specific products, services or information to the user).

500. Furthermore, the ’068 patent defines a “tool feature” as a function that can initiate software sequences to display content or control the display of data on the UI. ’068 patent, 4:23-27. The analytics gathered within Dasgupta’s advertisement facility utilize user profile data—such as historical access data—to make decisions about advertisement selection and targeting, which a POSITA would have recognized as an enabled “tool feature” that directs specific content to the user based on the collected analytics. Consequently, the system in Dasgupta both gathers and utilizes analytics based on user-interaction data, stored and processed on the master server, to effectively target advertisements to users according to their preferences and historical actions within the platform.

501. Thus, it is my opinion that Dasgupta, in view of the disclosures of the ’068 patent, adequately suggests that the master server collects analytics based

on enabled tool features to target specific products, services, or information to users, as required by claim limitation 1[c2].

XIII. CONCLUSION

502. In my opinion, the Challenged Patents do not claim a novel invention. At least for all of the reasons explained above, combinations of Darnell Kim, and Dasgupta, disclose every feature of the Challenged Claims.

XIV. DECLARATION

I hereby 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: December 23, 2024.

By: _____

Paul D. Martin