UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

TIKTOK, INC.,

Petitioner,

v.

CELLSPIN SOFT, INC.,

Patent Owner

IPR2024-00757

U.S. Patent No. 8,756,336 B2

Before GREGG I. ANDERSON, CYNTHIA M. HARDMAN, and MICHAEL A. VALEK, *Administrative Patent Judges.*

PATENT OWNER RESPONSE

TABLE OF CONTENTS

I. INTRODUCTION 1

II. LEGAL FRAMEWORK 1

III. CLAIM CONSTRUCTION 5

IV. GENERAL RESPONSE 15

- A. Discussion of the OSI Layers 16
- B. Discussion of Aaltonen 24
- C. Discussion of Drescher 29
- D. Discussion of Takahashi 36
- E. Discussion of Na 38
- F. Discussion of Lind 45

V. SPECIFIC REBUTTAL TO ALLEGED OBVIOUSNESS COMBINATIONS 45

- A. Ground One Fails (Aaltonen and Drescher) 45
- B. Ground Two Fails (Takahashi and Na and Lind) 68

VI. CONCLUSION 90

TABLE OF AUTHORITIES

01 Communique Lab., Inc. v. Citrix Sys., No. 2017-1869,
2018 U.S. App. LEXIS 10614 (Fed. Cir. Apr. 26, 2018) 7

Amazon.com, Inc. v. Barnesandnoble.com, Inc.,
239 F.3d 1343 (Fed. Cir. 2001) 7

Amgen Inc. v. Hoechst Marion Roussel, Inc.,
314 F.3d 1313 (Fed. Cir. 2003) 7

Belden Inc. v. Berk-Tek LLC,
805 F.3d 1064 (Fed. Cir. 2015) 4

Current Lighting Solns v. Jiaxing Super Lighting,
2023 WL 5319763 (PTAB August 16, 2023).....5

Graham v. John Deere Co.,
383 U.S. 1 (1966).....3

In re Kahn,
441 F.3d 977 (Fed. Cir. 2006) 5

In re Stepan Co.,
868 F.3d 1342 (Fed. Cir. 2017) 2

Intercontinental Great Brands LLC v. Kellogg,
869 F.3d 1336 (Fed. Cir. 2017) 2

InTouch Techs., Inc. v. VGO Commc'ns, Inc.,
751 F.3d 1327 (Fed. Cir. 2014).....3

Merck Sharp & Dohme B.V. v. Warner Chilcott Co., LLC,
711 F. App'x 633 (Fed. Cir. 2017).....4

Metalcraft of Mayville, Inc. v. Toro Co.,
848 F.3d 1358 (Fed. Cir. 2017) 3

Orexo AB v. Actavis Elizabeth LLC,
903 F.3d 1265 (Fed. Cir. 2018) 4

Otsuka Pharm. Co. v. Sandoz, Inc.,
678 F.3d 1280 (Fed. Cir. 2012) 4

Hulu, LLC v Soundview Innovations LLC,
IPR2018-01039, Paper 34, 22 (August 5, 2019).....4

KSR Int'l Co. v. Teleflex Inc.,
550 U.S. 398 (2007).....3, 5

PAR Pharm., Inc. v. TWI Pharms., Inc.,
773 F.3d 1186 (Fed. Cir. 2014).....2

Personal Web Techs., LLC v. Apple, Inc.,
848 F.3d 987 (Fed. Cir. 2017) 5

Phillips v. AWH Corp.,
415 F.3d 1303 (Fed. Cir. 2005) 9

Polaris Indus., Inc. v. Arctic Cat, Inc.,
882 F.3d 1056 (Fed. Cir. 2018) 5

TQ Delta v. Cisco Sys., Inc.,
942 F.3d 1352 (Fed. Cir. 2019) 4

Trivascular, Inc. v. Samuels,
812 F.3d 1056 (Fed. Cir. 2016) 3

Unigene Labs., Inc. v. Apotex, Inc.,
655 F.3d 1352 (Fed. Cir. 2011).....3

TABLE OF EXHIBITS

Number	Description
2001	Declaration of Pavan M. Gajendragad
2002	Petitioner's District Court Disclosures
2003	Cohen
2004	Harrison
2005	Joint Claim Construction Statement
2006	2007 CCNA Cisco Certified Network Association Study Guide
2007	WAPARCH
2008	MMS 1.0 Specification Excerpts
2009	IrDA - OBEX Version 1.3
2010	Book Excerpts
2011	'030 File Wrapper
2012	Book Excerpts
2013	'766 File Wrapper
2014	'121 Patent Notice of Allowance
2015	'381 Patent Notice of Allowance
2016	2003 Bluetooth Specification, Version 1.1
2017	Website Excerpts
2018	MIT Article
2019	Wikipedia Page
2020	Oracle Document

2021	Wikipedia Page

I. INTRODUCTION

Petitioner TikTok fails to demonstrate that any of Challenged Claims 1-5 of United States Patent No. 8,756,336 B2 ("the '336 Patent") is unpatentable over the cited art. Indeed, there exist multiple independent grounds upon which the Board should hold the Challenged Claims as non-obvious and patentable over the cited art, inasmuch as the cited art collectively fails to teach every limitation of the invention as claimed.

More specifically, and among other deficiencies, the cited art collectively fails to disclose at least the following: (i) a user identifier; (ii) a segment identifier; and (iii) applying both those identifiers at an application layer level. Moreover, certain of the proposed combinations suggested by Petitioner are unworkable and incompatible, thus negating any likelihood that a POSITA would have made such combinations. Accordingly, Patent Owner requests that the Board hold that the Challenged Claims are patentable over the art cited by the Petitioner.

II. LEGAL FRAMEWORK

While Patent Owner recognizes the Board's knowledge and familiarity with the pertinent law as it relates to obviousness, certain principles are emphasized here.

A claimed invention is unpatentable for obviousness under 35 U.S.C. § 103 "if the differences between the claimed invention and the prior art are such

that the claimed invention as a whole would have been obvious" to one of ordinary skill in the relevant art. *Intercontinental Great Brands LLC v. Kellogg N. Am. Co.*, 869 F.3d 1336, 1343–44 (Fed. Cir. 2017) (quoting 35 U.S.C. § 103). Under that provision, the challenger must establish by clear and convincing evidence that "the differences between the subject matter sought to be patented 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 said subject matter pertains." *Id.*

When several prior art references are considered, a patent may not be found invalid as obvious unless the party challenging the validity of the patent has provided evidence showing that: (i) "a skilled artisan would have had reason to combine the teaching of prior art references to achieve the claimed invention," and (ii) "that the skilled artisan would have had a reasonable expectation of success from doing so." *PAR Pharm., Inc. v. TWI Pharms., Inc.*, 773 F.3d 1186, 1193 (Fed. Cir. 2014); *see also In re Stepan Co.*, 868 F.3d 1342, 1346 n.1 (Fed. Cir. 2017) ("Whether a rejection is based on combining disclosures from multiple references, combining multiple embodiments from a single reference, or selecting from large lists of elements in a single reference, there must be a motivation to make the combination and a reasonable expectation that such a combination would be successful").

When determining the question of obviousness, the Board must consider the underlying facts, including the scope and content of the prior art, differences between the claims and the prior art, the level of ordinary skill, and other secondary considerations, if any. *See Graham v. John Deere Co.,* 383 U.S. 1, 17–18 (1966). The question of obviousness must be in the context of the claimed subject matter, as a whole. *See KSR Int'l Co. v. Teleflex Inc.,* 550 U.S. 398, 406 (2007); *see also W.L. Gore & Assocs. v. Garlock, Inc.,* 721 F.2d 1540 (1983) (It is legal error to "consider claims in less than their entireties" and "consider the references in less than their entireties, *i.e.,* disregard disclosures in the references that diverge from and teach away from the invention at hand"); *InTouch Techs., Inc. v. VGO Commc'ns, Inc.,* 751 F.3d 1327, 1351 (Fed. Cir. 2014); *Metalcraft of Mayville, Inc. v. Toro Co.,* 848 F.3d 1358, 1367 (Fed. Cir. 2017) .

Of course, "[o]bviousness requires more than a mere showing that the prior art includes separate references covering each separate limitation in a claim." *Unigene Labs., Inc. v. Apotex, Inc.,* 655 F.3d 1352, 1360 (Fed. Cir. 2011). "Rather, obviousness requires the additional showing that a person of ordinary skill at the time of the invention would have selected and combined those prior art elements in the normal course of research and development to yield the claimed invention." *Id.*; *Trivascular, Inc. v. Samuels,* 812 F.3d 1056, 1066 (Fed. Cir. 2016) ("Although the KSR test is flexible, the Board 'must still be careful not to allow hindsight

reconstruction of references . . . without any explanation as to how or why the references would be combined to produce the claimed invention'"). Indeed, the Federal Circuit stated that "it is improper to combine references like separate pieces of a simple jigsaw puzzle without explaining what reason or motivation one of ordinary skill in the art at the time of the invention would have had to place these pieces together." *Merck Sharp & Dohme B.V. v. Warner Chilcott Co., LLC,* 711 F. App'x 633, 636 (Fed. Cir. 2017) (internal quotations and brackets omitted). "The inventor's own path itself never leads to a conclusion of obviousness; that is hindsight." *Otsuka Pharm. Co. v. Sandoz, Inc.,* 678 F.3d 1280, 1296 (Fed. Cir. 2012). Specifically, "[i]t is inappropriate to use the template provided by the inventor, to render the inventor's contribution obvious." *Orexo AB v. Actavis Elizabeth LLC,* 903 F.3d 1265, 1271 (Fed. Cir. 2018).

When a Petitioner does not articulate a reason to combine references, other than a generic statement of anticipated success in doing so, there is an insufficient showing that an ordinarily skilled artisan would have been motivated to do so. *Hulu, LLC v Soundview Innovations LLC,* IPR2018-01039, Paper 34, 22 (August 5, 2019) (informative) (discussing *Belden Inc. v. Berk-Tek LLC,* 805 F.3d 1064, 1073 (Fed. Cir. 2015)) ("[O]bviousness concerns whether a skilled artisan not only could have made but would have been motivated to make the combinations … of prior art to arrive at the claimed invention."); *see also TQ Delta v. Cisco Sys., Inc.,*

942 F.3d 1352, 1359 (Fed. Cir. 2019) ("Rejections on obviousness grounds, in particular, cannot be sustained by mere conclusory statements; instead, there must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness") (citing *KSR,* 550 U.S. at 418 and *In re Kahn,* 441 F.3d 977, 988 (Fed. Cir. 2006)); *Polaris Indus., Inc. v. Arctic Cat, Inc.,* 882 F.3d 1056, 1068 (Fed. Cir. 2018); *Personal Web Techs., LLC v. Apple, Inc.,* 848 F.3d 987, 993-94 (Fed. Cir. 2017) (vacating Board's motivation-to-combine finding because the Board's "reasoning seem[ed] to say no more than that a skilled artisan, once presented with the two references, would have understood that they could be combined. And that is not enough: it does not imply a motivation to pick out those two references and combine them to arrive at the claimed invention"). Likewise, failure to adequately support a contention that there are a limited number of finite solutions to choose from yields an unsupported petition. *Current Lighting Solutions v. Jiaxing Super Lighting Electric Appliance Co., Ltd.,* 2023 WL 5319763 at *11 (PTAB August 16, 2023).

III. CLAIM CONSTRUCTION

The District Court in this matter is in the process of carrying out its claim construction role, and both parties have provided specific constructions to many terms in the claims of the Asserted Patents. Still further, the parties have exchanged claim construction of key terms as required by the local rules of the district court.

In the pending seven IPRs, Petitioner has advanced broad interpretations for all terms of all Asserted Patents, and Petitioner has not requested the construction of any specific term. Below is an example text from the IPR2024-00768 Petition at page 7 (similar text is in all seven Petitions):

> **IV. CLAIM CONSTRUCTION**
>
> All claim terms should be construed according to the *Phillips* standard. *Phillips v. AWH Corp.*, 415 F.3d 1303 (Fed. Cir. 2005); 37 C.F.R. § 42.100. Based on the well-established principal that "claim terms need only be construed to the extent necessary to resolve the controversy," no express constructions are required to institute review and find the Challenged Claims unpatentable. *Wellman, Inc. v. Eastman Chem. Co.*, 642 F.3d 1355, 1361 (Fed. Cir. 2011).

Meanwhile, Petitioner is seeking to secure narrow constructions of ***one hundred plus*** **claim terms** in the District Court. Exhibit 2002 (Petitioner's Patent L.R. 4-1, 4-2 and 4-3 Disclosures). This Board should not allow Petitioner to capitalize on its inconsistent positions, and to potentially shield them from simultaneous scrutiny by the Federal Circuit. The Board should not allow the creation of unbalanced playing field and should adopt the same claim constructions as are being advanced by the Petitioner at the District Court (and as generally agreed by Patent Owner). Since Nov 13 2018, PTAB has since adopted the claim construction analysis that follow the principles articulated in *Phillips*, 415 F.3d. This the same as what is required at the US District Courts. In patent law, the principle that claim terms should be construed consistently for both invalidity and

infringement is well-established. Just recently Federal Circuit again emphasized that point in *01 Communique Lab., Inc. v. Citrix Sys.,* No. 2017-1869, 2018 U.S. App. LEXIS 10614 (Fed. Cir. Apr. 26, 2018)

"Claim terms must be "construed the same way for both invalidity and infringement." *Amgen Inc. v. Hoechst Marion Roussel, Inc*., 314 F.3d 1313, 1330 (Fed. Cir. 2003); see *Amazon.com, Inc. v. Barnesandnoble.com, Inc*., 239 F.3d 1343, 1351 (Fed. Cir. 2001) ("A patent may not, like a nose of wax, be twisted one way to avoid anticipation and another to find infringement." (citations and internal quotation marks omitted))."

The District Court trial is two months **before** the nominal date for issuance of a Final Written Decision here. Given that both the cases will almost surely be appealed, the Federal Circuit should be able to simultaneously scrutinize the claims for infringement and invalidity using the same claim construction and not the inconsistent positions taken by Petitioner.

These actions (inconsistent claim construction positions) are exactly what two recognized Intellectual Property, National Security, and National Competitiveness Policy experts have opined about; namely, the practice of so-called "lawfare," in which China's IP-related industrial policy favors the use of legal tools to advance national technology interests. Such interests have the goal of

creating an unbalanced playing field with foreigners. *See* Exhibit 2003 (Cohen) and Exhibit 2004 (Harrison).

If the Board applies claim construction inconsistently with the ones adopted by the District Court, the Federal Circuit would have to resolve these issues for many terms. Petitioner has taken a drastically different approach regarding claim construction for all Asserted Patents before this Board as compared with its positions in the District Court. Here, Petitioner said NO construction is required of any claim term in any of the seven patents.

On December 20, 2024, the parties filed required disclosures with the District Court concerning claim construction. Below are the five key claim terms that are at issue for the '336, '757, and '260 Patents (*see* Joint Claim Construction and Prehearing Statement, Exhibit 2005):

	Claim Term	**Proposed Constructions**
	"along with"	"in addition to"
	"user identifier"	"a text string that identifies uniquely a user of a web service" Alternatively: "a text string that identifies uniquely a user of a web service and contains sufficient information to authenticate a user of the web service"
	"segment identifier"	"a text string that identifies uniquely a data segment"
	"application layer"	"The highest layer of standards in Open Systems Interconnection (OSI) reference model. The application

		layer contains signals that perform useful work for the user, such as file transfer or remote access to a computer, as opposed to lower levels, which control the exchange of data between transmitter and receiver."
	"Web Service"	"A modular collection of Web protocol based applications that use standard Internet protocols such as HTTP, XML, and SOAP to provide connectivity"

The earlier '127 IPR Petition was filed prior to November 13, 2018, and the Board there construed terms of the challenged claims using the so-called "broadest reasonable interpretation" in light of the specification ("BRI"). Of course, since then, the Board has adopted a claim construction procedure which follows the principles articulated in *Phillips v. AWH Corp.,* 415 F.3d 1303 (Fed. Cir. 2005). Thus, earlier decisions based on the BRI standard have no bearing on the present case, at least insofar as claim construction is concerned.

The claim term **"along with"** is a disputed claim term at the District Court, and Patent Owner is asking both courts to construe "along with" as "in addition to." This proposed construction is fully supported by the opinions of Pavan M. Gajendragad. *See* Gajendragad Declaration [Exhibit 2001] at ¶¶ 38.

With reference to the '802 application specification, and specifically at [0047], a POSITA would appreciate that the patentee has clearly stated that "along with" should mean "in addition to." This evidence is clear (s*ee* Gajendragad [Exh. 2001] at ¶ 39):

Intrinsic Evidence from the patent specification

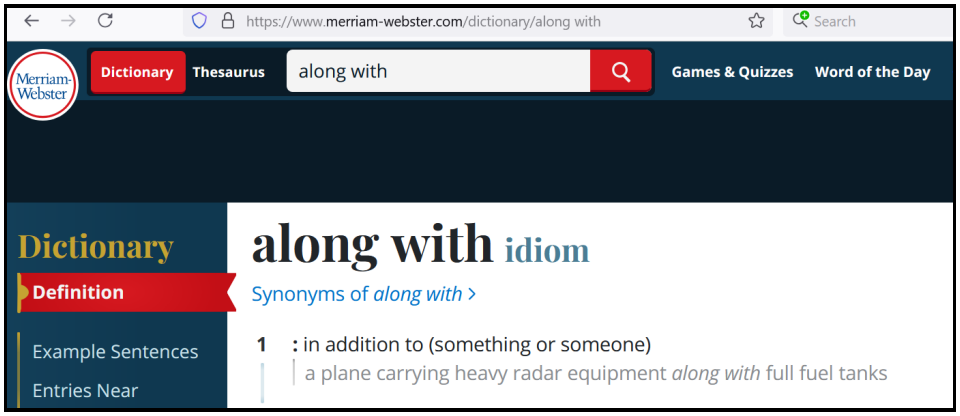
> **[0047]** If the user **201** is not satisfied **405** with the created multimedia object, the user **201** may review or delete **406** the multimedia object and start recording again. Otherwise, the user **201** transfers the multimedia object for publishing to a publication space selected by the user **201**. Additional information such as the list of the user preferences of the publication virtual spaces **205**, authentication identification, user preferences, default settings, manually entered geographical codes, and auction identifiers for eBay® and other shopping websites are selected **407** and transferred along with the user created multimedia content to the publishing service **204**. A remote method call **408** such as extensible markup language-remote procedure call (XML-RPC) may be initiated by the client application **202** to transfer the user created multimedia content along with the additional information to the publishing service **204**. The transfer to the publishing service **204** takes place using wireless technology, for example, global system for mobile communications, code division multiple access technology, wireless fidelity, or worldwide interoperability for microwave access.

Again, the '802 specification at]0047] states: "**Additional information** such as …., authentication identification, user preferences, ……and transferred *along with* the user created multimedia content to the publishing service." The "client application 202 to **transfer** the user created multimedia content *along with* the **additional information** to the publishing service."

A POSITA would thus appreciate that a variety of information was contemplated as being sent "along with" the multimedia file. Such information included: (i) user preferences; (ii) authentication identification; (iii) user preferences; (iv) default settings; (v) geographical codes; and (vi) auction identifiers and other identifiers. *See* Gajendragad [Exh. 2001] at ¶¶ 40-41.

Because such complex settings could not possibly have been incorporated into a media file's name, and because Petitioner makes no showing establishing this practice as of 2007, Petitioner's arguments fail.

It is further noted that dictionary evidence (https://www.merriam-webster.com/dictionary/along%20with) supports this understanding; namely, that "along with" means "in addition to," as illustrated below (*see also* Gajendragad [Exh. 2001] at ¶42):



Indeed, Dr. Gajendragad expressly disagrees with the suggestion that the plain and ordinary meaning of "along with" could include *"inside." See* Gajendragad [Exh. 2001] at ¶ 43. In sum, a POSITA would understand "along with" to mean "in addition to" or "together with," indicating something is alongside another thing. The concept of "inside," on the other hand, refers to something being physically contained within the boundaries of another object; essentially, "along with" describes a broader association, whereas "inside"

specifies a physical location within something else. *See* Gajendragad [Exh. 2001] at ¶ 43-44.

Still further, the term "segment identifier" is also disputed by the parties in the District Court case, as it relates to the '336, '757, and '260 Patents. The dispute here concerns a proposal from Petitioner (TikTok) to construe the term to mean "a text string that identifies uniquely a data segment," to which Cellspin agreed. A POSITA would understand that HTTP based identifiers must be text based because HTTP is a text-based protocol, meaning that the data that is transmits is typically in plain text format. This includes the request and response headers, which are human-readable text strings. *See* Gajendragad [Exh. 2001] at ¶ 46. Thus, a POSITA would have understood "segment identifier" in these claims as "a text string that identifies uniquely a data segment." *Id.*

The main reason to adopt the "segment identifier" in these claims as "a text string that identifies uniquely a data segment" is because for both invalidity and infringement each of Patent Owner and Petitioner have chosen HTTP as an application layer protocol for transfer of data. All prior art asserted in these related proceedings involve use of HTTP. Of course, HTTP is a (Hyper**Text** Transfer Protocol) application-layer protocol, based on **TEXT**. ALL the tags, header, are in *Text*. Therefore, the segment identifier has to be at least a "text string" that

identifies uniquely a data segment. TikTok itself proposed this construction in its District Court disclosures.

Similarly, the term "user identifier" is contested by the parties in the District Court case, as it relates to the '336, '757, and '260 Patents. The dispute here once again concerns a proposal from Petitioner to construe the term to mean "a text string that identifies uniquely a user of a web service and contains sufficient information to authenticate a user of the web service," to which Patent Owner agreed. As noted, a POSITA would understand that HTTP (Hyper**Text** Transfer Protocol) is an application-layer protocol, based on TEXT. All of the tags, header, etc. are in Text. Therefore, the "user identifier" would necessarily be understood by a POSITA as at least a "text string" that identifies uniquely a user of a website / web service. *See* Gajendragad [Exh. 2001] at ¶ 47.

Still further, the term "application layer" is contested by the parties in the District Court case, as it relates to the '336, '757, and '260 Patents. As to this term, Patent Owner has proposed a construction based on the June 2002, Microsoft Computer Dictionary Fifth Edition (page 32), which is set forth below. As confirmed by Dr. Gajendragad, a POSITA would have likewise understood "application layer" to have this defined meaning. *See* Gajendragad [Exh. 2001] at ¶ 48.

application layer *n.* The highest layer of standards in the Open Systems Interconnection (OSI) reference model. The application layer contains signals that perform useful work for the user, such as file transfer or remote access to a computer, as opposed to lower levels, which control the exchange of data between transmitter and receiver. See the illustration. *See also* ISO/OSI reference model.

ISO/OSI MODEL	
ISO/OSI Layer	***Focus***
Application *(highest level)*	Program-to-program transfer of information
Presentation	Text formatting and display, code conversion
Session	Establishing, maintaining, and coordinating communication
Transport	Accurate delivery, service quality
Network	Transport routes, message handling and transfer
Data-link	Coding, addressing, and transmitting information
Physical	Hardware connections

Application layer. *The highest layer in the ISO/OSI reference model.*

Finally, the term "web service" is contested by the parties in the District Court case, as it relates to the '336, '757, and '260 Patents. Once again, Patent Owner has proposed a construction based on the June 2002 Microsoft Computer Dictionary Fifth Edition (page 564), which is set forth below. Such construction is confirmed by Dr. Gajendragad. *See* Gajendragad [Exh. 2001] at ¶ 49.

> **Web services** *n.* A modular collection of Web protocol–based applications that can be mixed and matched to provide business functionality through an Internet connection. Web services can be used over the Internet or an intranet to create products, business processes, and B2B interactions. Web services use standard Internet protocols such as HTTP, XML, and SOAP to provide connectivity and interoperability between companies.

IV. GENERAL RESPONSE

None of the alleged prior art references would have been combined by a POSITA in the manner described by Dr. Patrick Traynor or by Petitioner. Moreover, the various alleged prior art references and their combinations fail to teach **many** of the elements of the Asserted Claims, and in certain cases the proposed combinations actually lead to "unworkable" solutions, when compared with the inventions as claimed in the Asserted Claims. As such, a POSITA would have had NO reasonable expectation of success in implementing the proposed combinations. *See* Gajendragad [Exh. 2001] at ¶ 50.

Petitioner's expert relies upon the same base "combinations" to support his ultimate conclusions that each Asserted Claim is purportedly obvious. However, not only do the specific references (alone or in combination) fail to teach many of the claim limitations, a POSITA would not have been motivated to combine them in the first place. *See* Gajendragad [Exh. 2001] at ¶ 51. Moreover, and as discussed in greater detail below, there are a number of instances where the alleged prior art actually leads to unworkable solutions and implementation, when

compared with the inventions as claimed. Again, a POSITA would have had NO reasonable expectation of success in implementing the combinations.

Patent Owner expressly disagrees with Petitioner's characterization of the alleged prior art, especially to the extent they erroneously state that the art discloses specific limitations of the Asserted Claims. In the discussion below, each reference is discussed individually, and various obvious errors in Dr. Patrick Traynor's evaluation of the art are emphasized. Again, the art cited by Dr. Patrick Traynor fails to teach all limitations of the Asserted Claims, and in any event there would have been no motivation for a POSITA to combine them in the manner relied upon. *See* Gajendragad [Exh. 2001] at ¶ 53.

A. Discussion of The OSI Layers

In order to understand the invention and the asserted patents and the prior art it is important to establish an industry standard and definition of certain terms in 2007. *See* Gajendragad [Exh. 2001] at ¶¶ 53-54.

A POSITA would have understood "Application Layer" as defined in Microsoft Computer Dictionary Fifth Edition on Page 32, as follows: "The highest layer of standards in Open Systems Interconnection (OSI) reference model. The application layer contains signals that perform useful work for the user, such as file transfer or remote access to a computer, as opposed to lower levels, which control

the exchange of data between transmitter and receiver." *See* Gajendragad [Exh. 2001] at ¶ 57.

application layer *n.* The highest layer of standards in the Open Systems Interconnection (OSI) reference model. The application layer contains signals that perform useful work for the user, such as file transfer or remote access to a computer, as opposed to lower levels, which control the exchange of data between transmitter and receiver. See the illustration. *See also* ISO/OSI reference model.

ISO/OSI MODEL	
ISO/OSI Layer	***Focus***
Application *(highest level)*	Program-to-program transfer of information
Presentation	Text formatting and display, code conversion
Session	Establishing, maintaining, and coordinating communication
Transport	Accurate delivery, service quality
Network	Transport routes, message handling and transfer
Data-link	Coding, addressing, and transmitting information
Physical	Hardware connections

Application layer. *The highest layer in the ISO/OSI reference model.*

The seven OSI Layers are clearly defined in the 2007 CCNA Cisco Certified Network Associate Study Guide Second Edition by Todd Lammle. (Exhibit 2006) (at pp. 5-21). *See* Gajendragad [Exh. 2001] at ¶ 58.

It is clear from the text that as of 2007, segmentation and reassembly of data was done by the *Transport Layer* and NOT by the Application Layer as described and claimed in the Asserted Patents. *See* Page 10 of the Exhibit 2006: ("Services located in the *Transport layer* both **segment and reassemble** data from upper-layer applications and unite it onto the same data stream"). *See* Gajendragad [Exh. 2001] at ¶ 59.

The Transport Layer

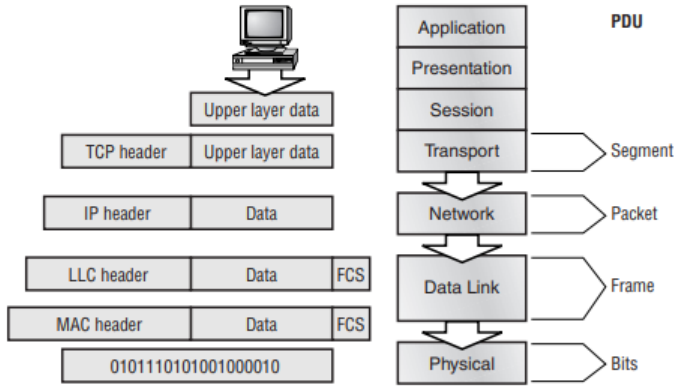
Services located in the *Transport layer* both segment and reassemble data from upper-layer applications and unite it onto the same data stream. They provide end-to-end data transport services and can establish a logical connection between the sending host and destination host on an internetwork.

Some of you might already be familiar with TCP and UDP (which you will learn about in Chapter 3) and know that TCP is a reliable service and UDP is not. Application developers have their choice of the two protocols when working with TCP/IP protocols.

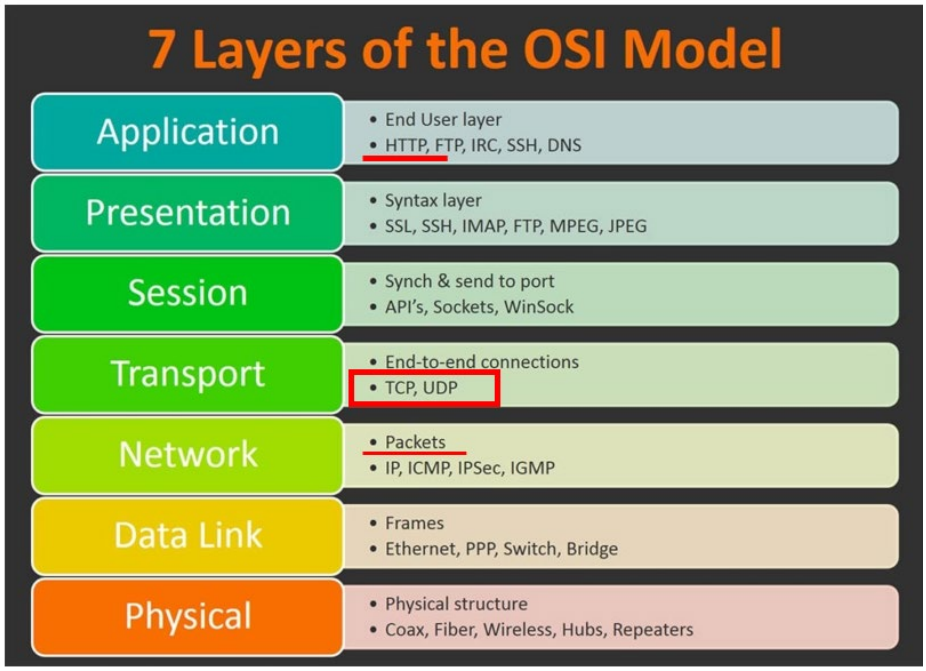
The Transport layer is responsible for providing mechanisms for multiplexing upper-layer application, session establishment, and teardown of virtual circuits. It also hides details of any network-dependent information from the higher layers by providing transparent data transfer.

is sequenced so the data stream can be put back together on the receiving side exactly as transmitted.

FIGURE 1.13 Data encapsulation



None of the Application Layer protocols did any segmentation of file/data at the application layer. Further, a POSITA would readily appreciate the fact that FTP and HTTP had no concept of segmentation of data at the Application Layer in 2007. *See* Gajendragad [Exh. 2001] at ¶ 60.



https://www.nwkings.com/osi-model-in-ccna-course

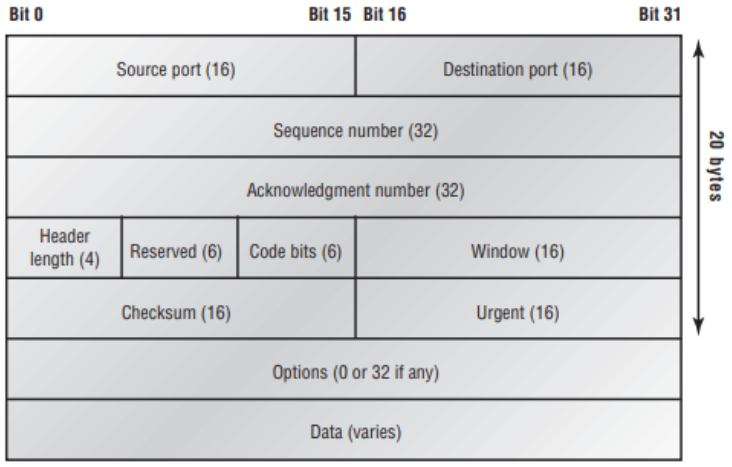
As of 2007, it was the Transport layer (TCP and UDP) that segmented the data, not the application layer. (Exhibit 2006) (Book page #108) ("Since the upper layers just send a data stream to the protocols in the Transport layers, we'll demonstrate how TCP segments a data stream and prepares it for the Network layer. The Network layer then routes the segments as packets through an internetwork"). *See* Gajendragad [Exh. 2001] at ¶ 61. The transport layer TCP Segment is illustrated below.

TCP Segment Format

Since the upper layers just send a data stream to the protocols in the Transport layers, we'll demonstrate how TCP segments a data stream and prepares it for the Network layer. The Network layer then routes the segments as packets through an internetwork. The packets are handed to the receiving host's Transport layer protocol, which rebuilds the data stream to hand to the upper-layer applications or protocols.

Figure 3.3 shows the TCP segment format. The figure shows the different fields within the TCP header.

FIGURE 3.3 TCP segment format

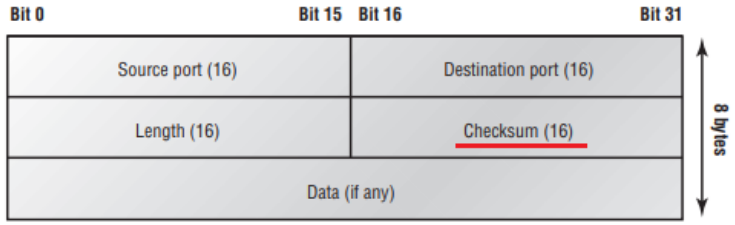


Further, below is illustrated the transport layer UDP Segment and structure of the UDP Packet and CRC (Checksum) inside the UDP packet. CRC stands for (*cyclic redundancy codes)* (*see* Gajendragad [Exh. 2001] at ¶ 62)*:*

UDP Segment Format

The very low overhead of UDP compared to TCP, which doesn't use windowing or acknowledgments, is shown in Figure 3.4.

FIGURE 3.4 UDP segment



You need to understand what each field in the UDP segment is. The UDP segment contains the following fields:

Source port Port number of the host sending the data

Destination port Port number of the application requested on the destination host

Length of the segment Length of UDP header and UDP data

CRC Checksum of both the UDP header and UDP data fields

Data Upper-layer data

UDP, like TCP, doesn't trust the lower layers and runs its own CRC. Remember that the Frame Check Sequence (FCS) is the field that houses the CRC, which is why you can see the FCS information.

The following shows a UDP segment caught on a network analyzer:

```
UDP - User Datagram Protocol
  Source Port:          1085
  Destination Port:     5136
  Length:               41
  Checksum:             0x7a3c
  UDP Data Area:
  ..Z............   00 01 5a 96 00 01 00 00 00 00 00 11
00 00 00
...C..2...._C._C   2e 03 00 43 02 1e 32 0a 00 0a 00 80 43
00 80
Frame Check Sequence:  0x00000000
```

Notice the low overhead! Try to find the sequence number, ack number, and window size. You will notice that these are absent from the UDP segment.

As of 2007, a POSITA would have understood that the industry definition of "packet" was as defined in the Microsoft Computer Dictionary Fifth Edition on Page 385, as follows: "packet n. 1. A unit of information transmitted as a whole from one device to another on a network. 2. In packet switching networks, a

transmission unit of fixed maximum size that consists of binary digits representing both data and a header containing an identification number, source and destination addresses, and sometimes error-control data." *See* Gajendragad [Exh. 2001] at ¶ 63.

> **packet** *n.* **1.** A unit of information transmitted as a whole from one device to another on a network. **2.** In packet-switching networks, a transmission unit of fixed maximum size that consists of binary digits representing both data and a header containing an identification number, source and destination addresses, and sometimes error-control data. *See also* packet switching.

As of 2007, a POSITA would have understood that the industry definition of "UDP" was as defined in Microsoft Computer Dictionary Fifth Edition on Page 536, as follows: "UDP n. Acronym for User Datagram Protocol. The connectionless protocol within TCP/IP that corresponds to the transport layer in the ISO/OSI reference model. UDP converts data messages generated by an application into packets to be sent via IP, but it is "unreliable" because it does not establish a path between sender and receiver before transmitting and does not verify that messages have been delivered correctly. UDP is more efficient than TCP, so it is used for various purposes, including SNMP; the reliability depends on the application that generates the message." *See* Gajendragad [Exh. 2001] at ¶ 64.

> **UDP** *n.* Acronym for **U**ser **D**atagram **P**rotocol. The connectionless protocol within TCP/IP that corresponds to the transport layer in the ISO/OSI reference model. UDP converts data messages generated by an application into packets to be sent via IP, but it is "unreliable" because it does not establish a path between sender and receiver before transmitting and does not verify that messages have been delivered correctly. UDP is more efficient than TCP, so it is used for various purposes, including SNMP; the reliability depends on the application that generates the message. *See also* communications protocol, ISO/OSI reference model, packet, SNMP, TCP/IP. *Compare* IP, TCP.

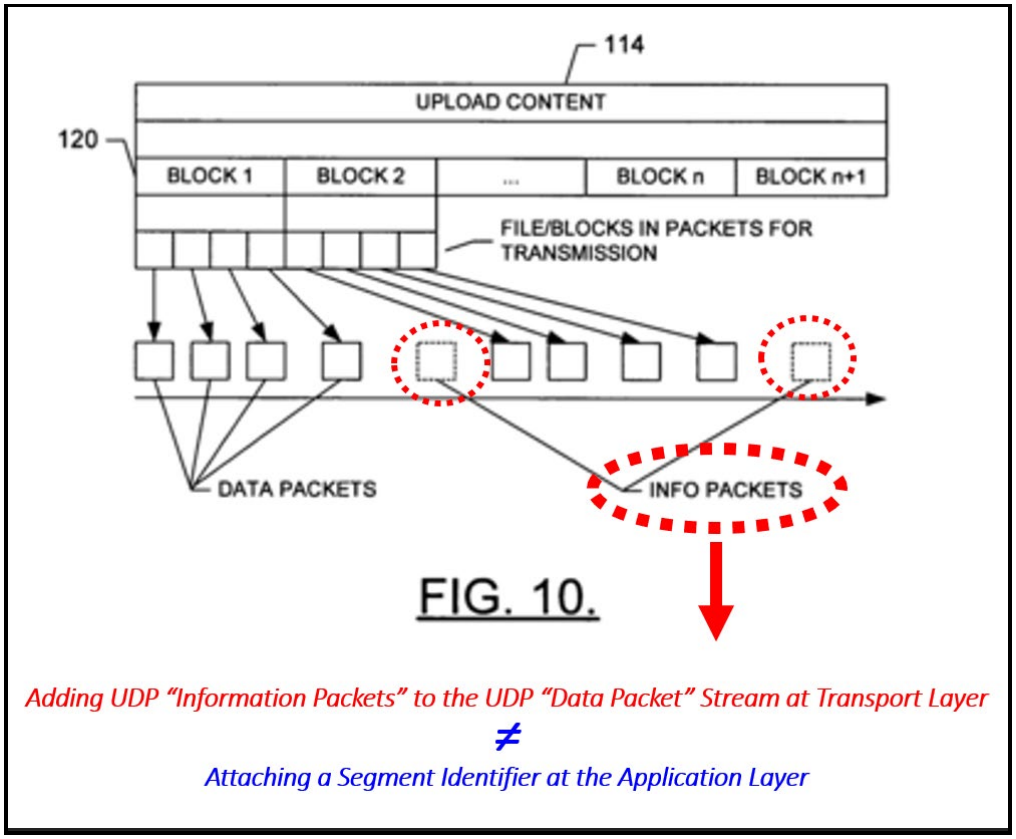
Still further, as of 2007, a POSITA would have understood that the industry definition of "HTTP" was as defined in the Microsoft Computer Dictionary Fifth Edition on Page 259, as follows: "HTTP n. Acronym for Hypertext Transfer Protocol. The protocol used to carry requests from a browser to a Web server and to transport pages from **Web servers back to the requesting browser**. Although HTTP is almost universally used on the Web, it is not an especially secure protocol." *See* Gajendragad [Exh. 2001] at ¶ 66.

> **HTTP** *n.* Acronym for **H**ypertext **T**ransfer **P**rotocol. The protocol used to carry requests from a browser to a Web server and to transport pages from Web servers back to the requesting browser. Although HTTP is almost universally used on the Web, it is not an especially secure protocol.

Finally, as of 2007, a POSITA would have understood that HTTP was only being used between web servers and requesting **Browsers** – not by standalone mobile applications. *See* Gajendragad [Exh. 2001] at ¶ 67.

B. Discussion of Aaltonen

As illustrated below, Aaltonen FIG 10 makes it very clear that INFO PACKETS are **not** attached to BLOCKs at the Application Layer. There is NO mention of Block Identifiers that are added or attached at the application layer in Aaltonen. *See* Gajendragad [Exh. 2001] at ¶ 68.



For **Element 1[j].** Petitioner relies on Aaltonen **Fig 11** (Head URI Request) for "sending, a synchronization request to the web service, **for said user identifier**"

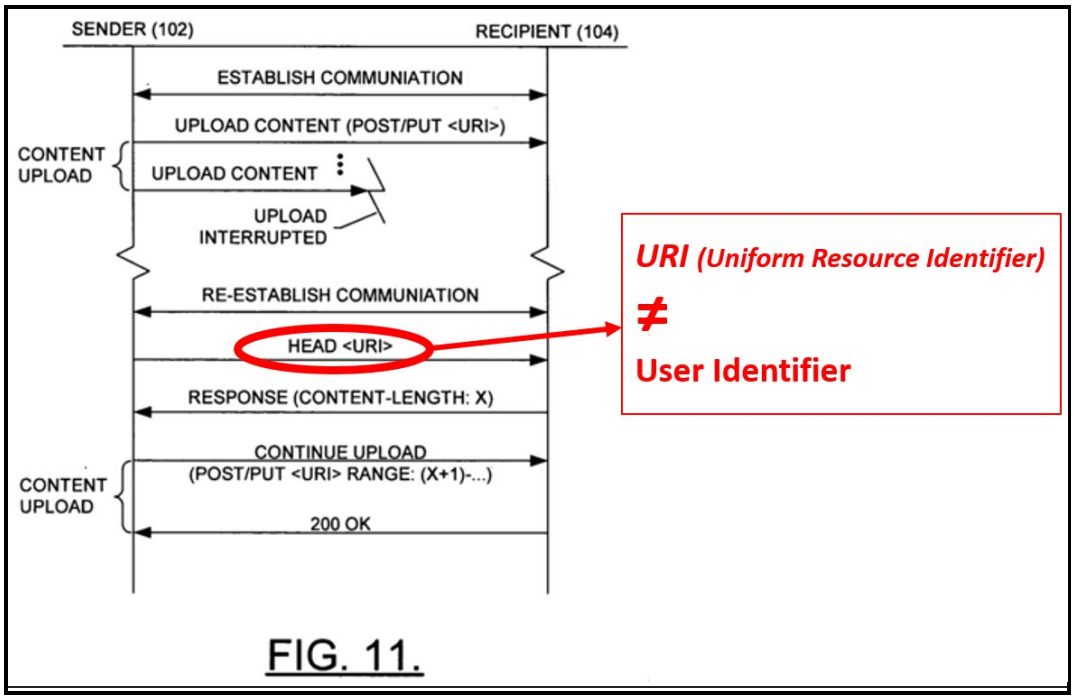


FIG. 11.

It is clear from Aaltonen [0111] and [0112] (discussing the control flow diagram of FIG. 11) that URI Stands for Uniform Resource Identifier (URI) and URI is not a User Identifier. Aaltonen **[0111]** teaches: "where messages in accordance with each include a header **identifying the <u>URI of a location</u>** within **content storage 112."** *See* Gajendragad [Exh. 2001] at ¶¶ 70-73.

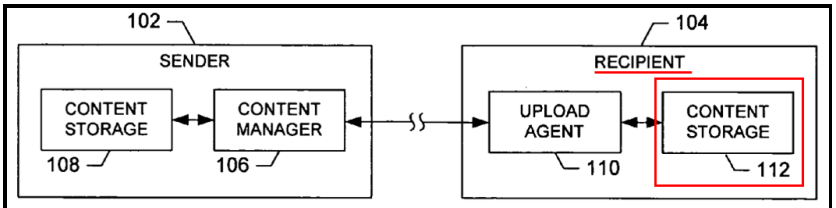


FIG. 6.

Aaltonen **[0112]** further teaches: "where the *<u>request includes</u>* the **URI of the location** within **content storage 112** including the received portion of the upload

content." A POSITA would appreciate that the Head Request in Aaltonen is NOT for a user identifier and the URI is of the location within the content storage. *See* Gajendragad [Exh. 2001] at ¶ 74.

Aaltonen **Fig 6** (as shown above) makes it clear that the **location of recipient content storage 112 (URI)** has *nothing* to do with the user ID. *See* Gajendragad [Exh. 2001] at ¶ 75. Indeed, the *only* time Aaltonen arguably mentions anything relating to any "user identifier" is in Aaltonen [102], as follows: "each <u>information packet can include</u> information (*e.g.,* Sequence of packet CRCs) uniquely describing the packets before or after the information packet, as well as <u>one or more identifiers identifying the Sender 102".</u> It is important to point out (and a POSITA would understand) that this **sender information is at the UDP information packet level**, and it is in reference to the IP address and destination port number of Sender 102. *See* Gajendragad [Exh. 2001] at ¶ 76.

UDP Checksum is *Optional* is also called CRC (*cyclic redundancy codes)*

UDP Packets

```
 0      7 8     15 16    23 24    31
+--------+--------+--------+--------+
|     Source      |   Destination   |
|      Port       |      Port       |
+--------+--------+--------+--------+
|                 |                 |
|     Length      |    Checksum CRC|
+--------+--------+--------+--------+
|
|          data octets ...
+---------------- ...

      User Datagram Header Format
```

That is why the text in Aaltonen says "one or more identifiers identifying the Sender 102" IP address and source and destination port inside the packet. It would have been readily understood by a POSITA that these IP address and Source and

Destination Port numbers are **NOT** unique because the internet and intranet use NAT (Network Address Translation) protocols at the network boundaries. *See* Gajendragad [Exh. 2001] at ¶ 77.

Further, the UDP INFO packets that include "one or more identifiers identifying the Sender 102" in Aaltonen are also encoded in Binary format. It is NOT a text string. *See* Gajendragad [Exh. 2001] at ¶ 78.

Thus, the "sender information" *inside the UDP INFO packet* of Aaltonen does not teach any "user identifier." It is *not* a "Text String" and it is *not* unique at the web service, and is *not* being applied at the Application Layer level. *See* Gajendragad [Exh. 2001] at ¶ 79.

In the District Court, "Web Service" is being defined as: "A modular collection of Web protocol based applications that use standard Internet protocols such as HTTP, XML, and SOAP to provide connectivity." This "Web Service" definition is based on the Microsoft Computer Dictionary Fifth Edition (page 564).

As such, the primary reason why Petitioner has to rely on Drescher for "user identifier" for Element **1[e]** is because Aaltonen does **not** have a user web site / webservice. For user web services with multiple users, Petitioner relies on a "recipient" that can be an "origin server 22," "digital broadcaster 32," and/or "digital broadcast receiver 36" ("one or more web services"). But Aaltonen **[0035]** clearly states that the origin server is **not** a user website or a web service. For

example, **origin servers** can provide content to the terminal, such *as in accordance with the Multimedia Broadcast* ***Multicast*** *Service (**MBMS**)." See* Gajendragad [Exh. 2001] at ¶ 81.

A POSITA would understand that HTTP itself is **not** inherently designed to work with multicast, as HTTP is a unicast protocol, meaning it is designed for one-to-one communication between a client and a server. Indeed, even up until 2020 it was not possible to use HTTP with Multicast. One approach, introduced **after 2020**, is using protocols like HTTP over Multicast QUIC. This method leverages the QUIC transport protocol to facilitate the transfer of HTTP resources over multicast IP, maintaining compatibility with **HTTP/3 that was introduced in 2021.** *See* https://www.ietf.org/archive/id/draft-pardue-quic-http-mcast-09.html. *See* Gajendragad [Exh. 2001] at ¶ 82.

Multicast QUIC is an extension of the QUIC *(Quick UDP Internet Connections)* protocol that allows for efficient one-to-many data transmission by utilizing multicast networking, enabling a single source to send data to multiple receivers simultaneously, similar to how live streaming or large software updates are distributed, while leveraging the security and performance benefits of the QUIC protocol itself; essentially combining the scalability of multicast with the reliability and security of QUIC. *See* Gajendragad [Exh. 2001] at ¶ 83.

Moreover, Aaltonen **[0039]** states that "digital broadcaster 32," and/or "digital broadcast receiver 36" is ALSO not a user website. Rather, Aaltonen **[0039]** teaches: "digital broadcast network, such as a **terrestrial** digital video broadcasting (*e.g.,* DVB-T, DVB-H, ISDB-T, ATSC, etc.) network". *See* Gajendragad [Exh. 2001] at ¶ 84.

The following servers are disclosed in Aaltonen: (i) "origin servers" that work in accordance with the Multimedia Broadcast *Multicast* Service (MBMS); and (ii) "digital broadcaster 32," and/or "digital broadcast receiver 36" (*e.g.,* DVB-T, DVB-H, ISDB-T, ATSC, etc.). However, NONE of these are based on HTTP, XML, and SOAP, and Petitioner has not shown otherwise (nor can it). *See* Gajendragad [Exh. 2001] at ¶ 85.

C. Discussion of Drescher

Drescher teaches a solution for server-to-server data transfer and a replacement for GIS and GridFTP protocol, together with using UNICORE job management and using Explicit Trust Delegation (ETD) variant. *See* Gajendragad [Exh. 2001] at ¶ 87.

The problems of GSI and GridFTP are discussed in FUJISTSU Sci. Tech. J. 40,2,p282-294 (December 2004). This document also describes the alternative UNICORE job management and security system for grid computing. UNICORE uses its own protocol for communication between the sites involved which runs over SSL. In the Explicit Trust Delegation (ETD) variant of UNICORE prposed in that document the security roles in the system include user, endorser and consigner, all of which have their identities established though the use of X.509 certificates issued by a certificate authority. Requests (known as Abstract Job Objects) for work to be executed on a server under the user's identity are transmitted to the server. These requests are signed either directly with the user's certificate or by an endorser, which is another entity in the in the UNICORE system, usually a server, that is permitted to make work requests on behalf of the user. (Such permissions are stored in the UNICORE system's authorisation

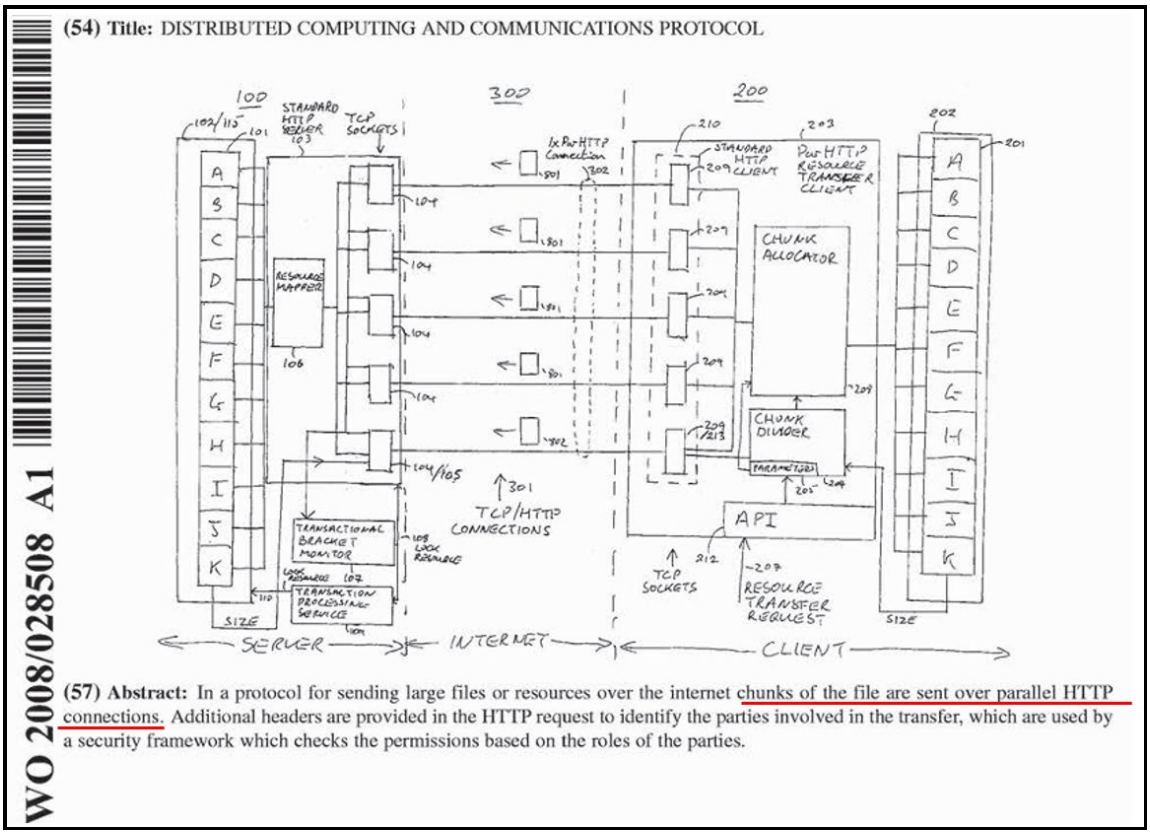
3

WO 2008/028508 PCT/EP2006/008710

3

database.) Where the transfer of a data file is required various implementations of UNICORE use different protocols but GridFTP is one of the options available.

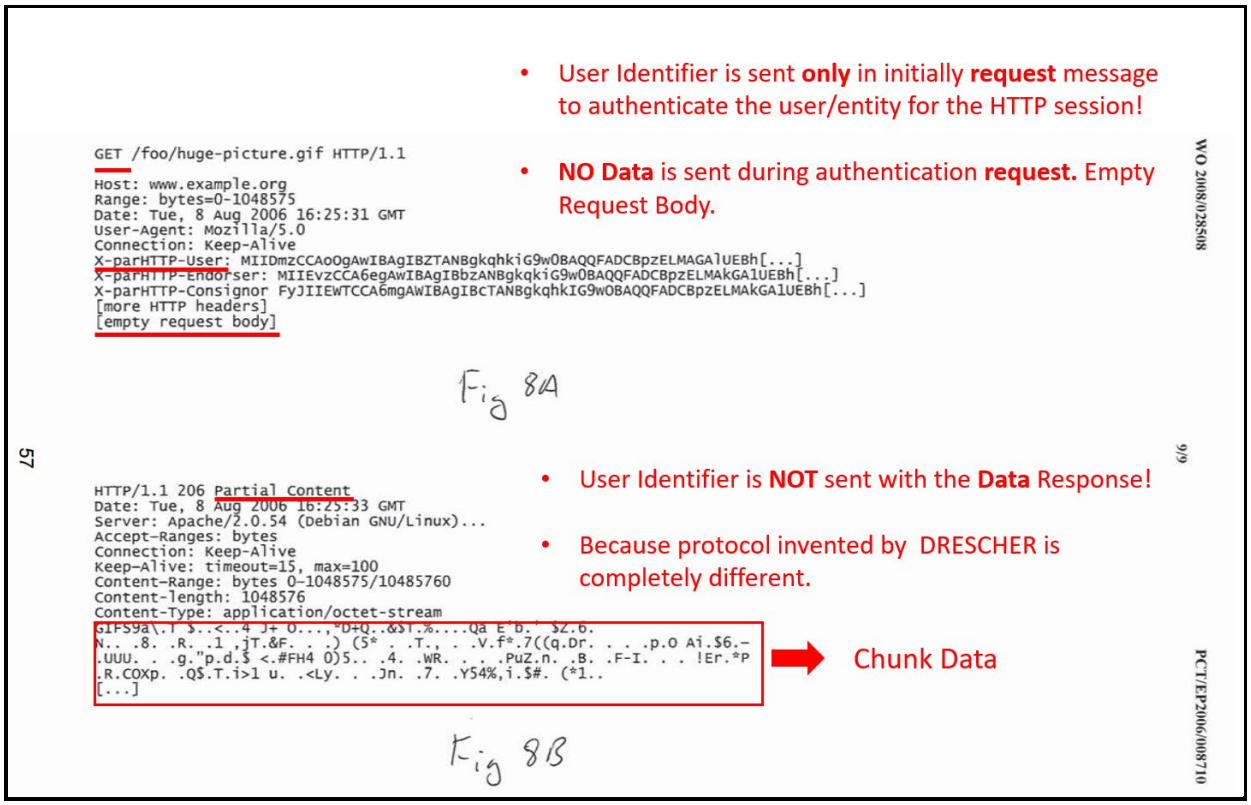
The Drescher solution is based on ParHTTP, which is a short form for Parallel HTTP. It is about sending large files in multiple chunks (the Grid Computing scenario) over multiple parallel HTTP Connections. In the main image shown below, Drescher uses five (5) parallel HTTP connections and five (5) separate parallel HTTP connections for each of the five (5) chunks. *See* Gajendragad [Exh. 2001] at ¶ 88.



As taught, the parHTTP method or parallel HTTP is a different protocol where "Chunks of file are sent over **Parallel HTTP** Connection**s**" where "the chunk allocator creates the number of parallel HTTP connection**s,**" and where "plural number of parallel HTTP stream**s** to provide transmission speed advantages." *See* Gajendragad [Exh. 2001] at ¶ 89.

Drescher **Fig 8A and Fig 8B** clearly show that the User ID is **ONLY** sent in the initial request message to authenticate the user for that HTTP Session. There is NO Data in the request body (*i.e.,* an empty request body). After the **Fig 8A** HTTP request for authentication is successful, in the Data response message **Fig 8B**, the

User ID is NOT sent along with the Chunk Data. This is because the parHTTP protocol is completely different. *See* Gajendragad [Exh. 2001] at ¶ 90.



Below is the text from Drescher describing the disclosures of **Fig 8A** and **Fig 8B**:

Figures 8A shows the text of the headers of a typical ParHTTP request, which shows both headers for transferring the certificates and the range header. As noted above in some embodiments of the invention either of those may be omitted. The text of the Figure has some annotations in square brackets; "[…]" indicates more similar data.

Figure 8B sows the response to the request. The data after "Content-Type: application/octet-stream" is binary and has been shown as random characters for the purpose of illustration.

The accompanying text (shown above) also makes it clear that transferring certificate for user ID / authentication is *only* done in the initial ParHTTP request and *not* in the response when data is being transferred. *See* Gajendragad [Exh. 2001] at ¶ 92.

Below is the pertinent paragraph from Drescher, which Petitioner relies on for the "Attaching user identifier" of Element **1[e]** ("applying a user identifier to each data segment for each destination web service at the application layer level"):

The secure version of ParHTTP provides, through the establishment of secure sockets as described above, a means of transferring one identity from the ParHTTP client to the

25

WO 2008/028508 PCT/EP2006/008710

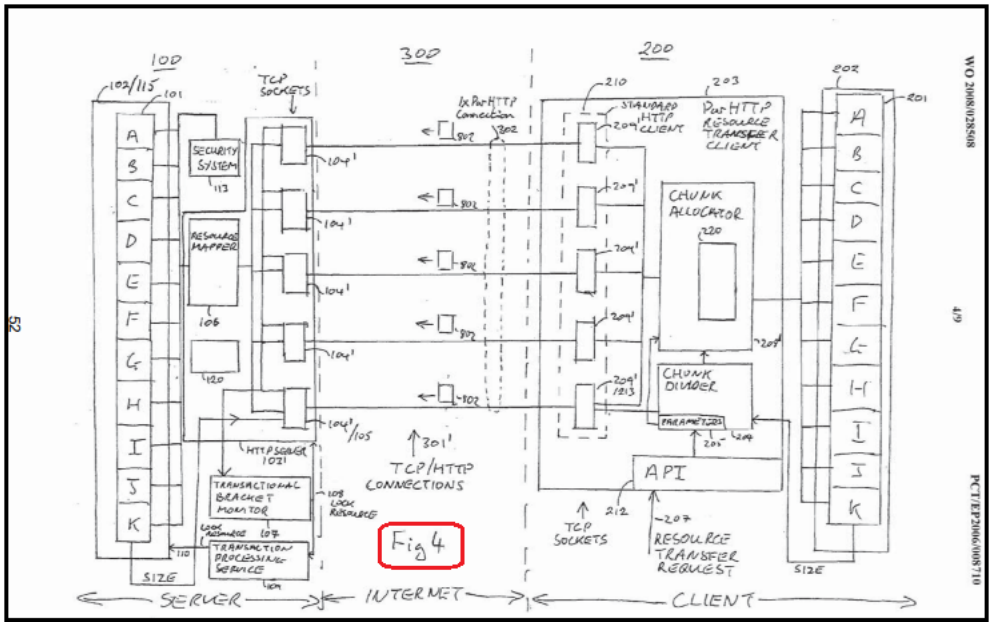
25

server. ParHTTP also provides a second method of communicating identities between the client 200 and the server 100. This is also implemented in the example of Figure 4 but may be used irrespective of whether the connection is secured. In this method the identity provider 220 adds to each HTTP request 802 sent to the server (i.e. for each chunk, and for PUT and GET) a header for each identity to be transferred. The header is called in the request "X-parhttp-<entity(i)>", where <entity(i)> is a name particularising the role of the ith entity, i.e. party, involved in the request. (In the later ETD example those roles are named "user", "endorser" etc., thus making the name of the header "X-parhttp-user" etc.. The information contained in the

Petitioner states that: "according to Drescher, HTTP PUT and POST messages have a user identifier disclosing "a name particularising the… [party] involved in the request" ("applying a user identifier" that "uniquely identifies a particular user of the web service")." However, Petitioner is incorrect, as Drescher does NOT do it for POST messages. Nowhere in Drescher is there any teaching that HTTP POST messages have user identifiers. The word "POST" itself does not

appear in Drescher. Again, it is clear that Drescher is limited to disclosing that it is ONLY done for **HTTP Request** (Fig 8A) and not response (Fig 8B) when data is being sent. *See* Gajendragad [Exh. 2001] at ¶¶ 94-95.

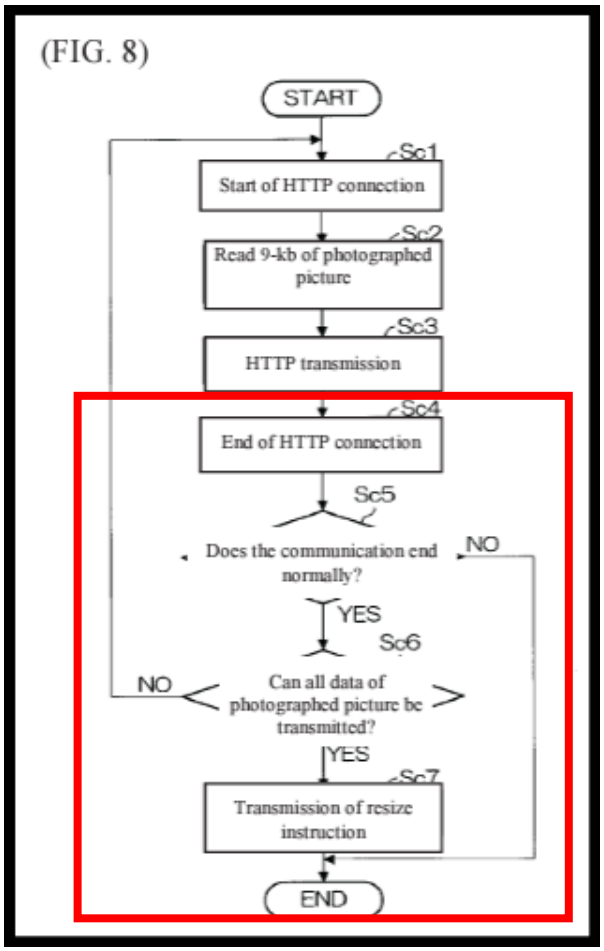
The reason the cited paragraph of Drescher (as relied upon by Petitioner) says for "each chunk" is because it (the cited text) **is talking about ParHTTP (Parallel HTTP)** and **Fig 4** (shown below) that has **five** (5) parallel HTTP connections sending different chunks and each of those five (5) HTTP connection have to be authenticated with a separate HTTP **request** with User ID (Fig 8A) before the chunk data can be sent in the HTTP **response (Fig 8B)**, and that HTTP response does **NOT** contain any user identifier (as clearly shown in Fig 8B) (*see* Gajendragad [Exh. 2001] at ¶ 96):



The User Identity of Drescher is sent for each parallel HTTP connection to be used by each chunk, BUT user identity is **NOT attached** or sent with any chunk **data**. User Identity is used **only initially** for each parallel HTTP connection authentication. *See* Gajendragad [Exh. 2001] at ¶ 97.

D. Discussion of Takahashi

As illustrated below, Takashi teaches a different and orthogonal data upload mechanism that does **full termination** of an upload if the connection is in any way interrupted. *See* Takahashi [Exh. 1016] at Fig. 8 (ending HTTP transmission in the event of abnormal termination of connection and recognizing necessity of an "all data" single transmission). As such, there is no "synchronization" of messages taught in Takahashi and, in fact, the opposite is expressly required. *See* Gajendragad [Exh. 2001] at ¶ 98.

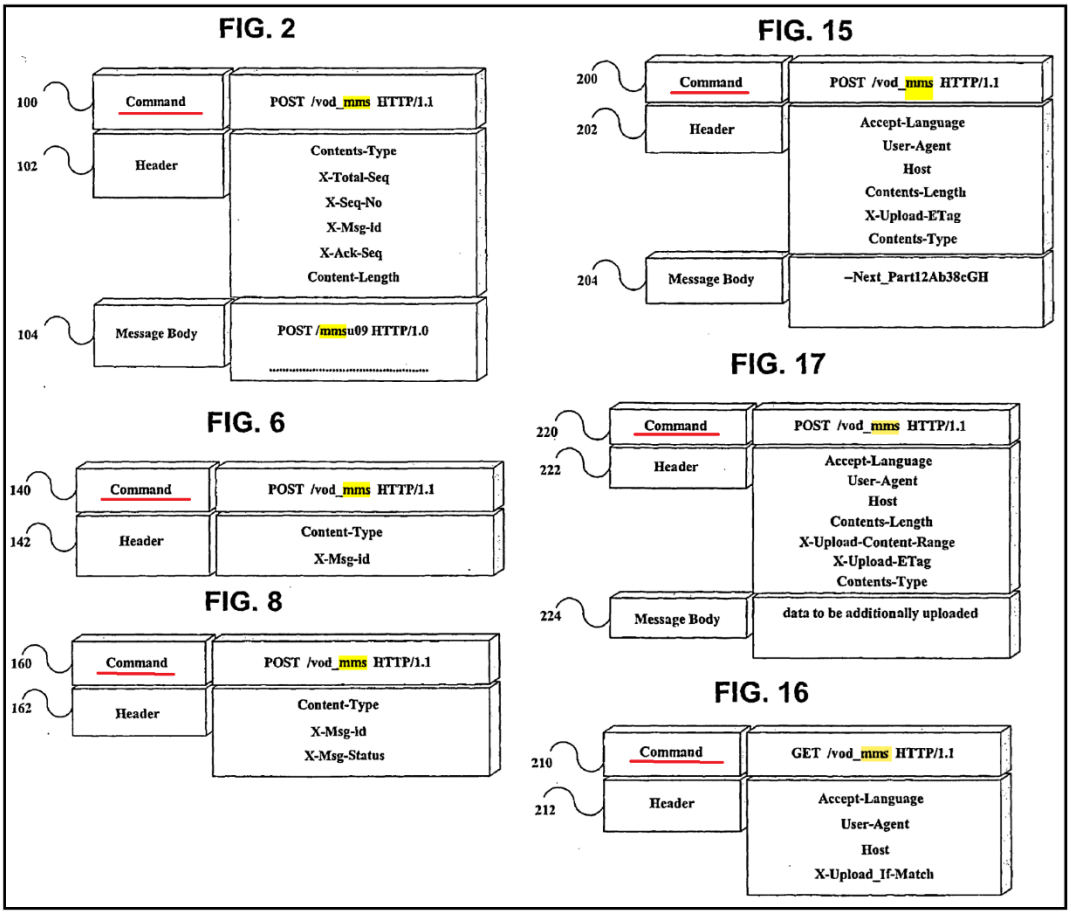


Further, Takashi fails to disclose any "segment identifier," given the fact that Takashi transmits full 9kb images sequentially and, if upload is interrupted, the entire process ENDs in Takashi. Further to this point, there is no disclosure of any "User ID" for each synchronization message. First, as explained above, there is no such synchronization message. Second, to the extent Takahashi discloses the use of a user identifier, such are integrated as part of the **upload file name and are not separately associated** with individual synchronized data messages. Takahashi does not teach sending any user identifier "along with" the data. Again, and as taught in Takahashi, the User ID (user identifier) is part of the <u>filename</u>, not an

identifier transmitted "in addition to" the segment identifier and the data. *See* Gajendragad [Exh. 2001] at ¶ 99.

E. Discussion of Na

The disclosures of Na are plainly directed to phone **MMS** *(Multimedia Messaging Service)* technology for the uploading of messages that are different from phone SMS messages or non-MMS messages. As shown below, all of the command messages contemplated in Na are subject to MMS 1.0 Specification requirements (*see* Gajendragad [Exh. 2001] at ¶ 100):



It would have been readily understood by a POSITA that **MMS** is a system application by which a Wireless Application Protocol (WAP) client is able to provide a messaging operation with a variety of media types. The WAP Forum defines a set of protocols in transport, security, transaction, session, and application layers. Still further, the Open Mobile Alliance (OMA) defines the WAP architecture, in: "Wireless Application Protocol Architecture Specification" (WAPARCH) (Exh. 2007 and Exh.2008) As a POSITA would have understood and appreciated, the phone MMS server application architecture and requirements are completely different from a regular web-based application architecture. *See* Gajendragad [Exh. 2001] at ¶ 101.

The MMS Application Server and MMS Open Mobile Alliance do not send filenames of the MMS media. Each segment message body in Na would have extra MMS 1.0 Specification data and header information, (as contemplated in Na at [0051]). *See* Gajendragad [Exh. 2001] at ¶ 102.

All segmenting and messaging of Na is in accordance with the MMS Application Server. Indeed, Na clearly states that every command includes MMS 1.0 specification information. *See, e.g.,* Na at [0062] ("The <u>command 100 includes MMS 1.0 specification information</u> (*e.g.,* POST/vod_mms HTTP/1.1)"); *see also id.* at [0065] ("The <u>message body 104 includes the first 512 bytes of the MMS 1.0 specification data</u>"); *see also id.* at [0069] ("The message body 122 includes a

message type field (X-**Mms**-Message-Type, e.g., M-send-conf) and an upload result report field (X-**Mms**-Response-Text, e.g., upload O.K.)"); *see also id.* at [0118] ("In FIG. 15, the message body 204 is defined within 64 bytes in accordance with the message body format of the MMS 1.0 specification, which is unique throughout the message"); *see also id.* at [0123] ("The message body 226 includes the data to be additionally uploaded in accordance with the message body format of the MMS 1.0 specification"). *See* Gajendragad [Exh. 2001] at ¶ 103.

MMS 1.0 specification information for MMS Protocol Data Units and MMS Header Fields are discussed in great detail in Exh. 2008 form Page 13 to 27. A brief recap of just the MMS **Header Information** for MMS Specification 1 is shown below. *See* Exh. 2008 form Page 42 to 43; *see also* Gajendragad [Exh. 2001] at ¶ 104:

Show = <Octet 129>

7.2.32 X-Mms-Status field

Status-value = Expired | Retrieved | Rejected | Deferred | Unrecognised |

Expired = <Octet 128>

Retrieved = <Octet 129>

Rejected = <Octet 130>

Deferred = <Octet 131>

Unrecognised = <Octet 132>

Indeterminate = <Octet 133>

Forwarded = <Octet 134>

The value Unrecognised is reserved for version management purpose only.

7.2.33 Subject field

Subject-value = Encoded-string-value

7.2.34 To field

To-value = Encoded-string-value

See Chapter 8 for addressing model.

7.2.35 X-Mms-Transaction-Id field

Transaction-id-value = Text-string

MMS Specifications 1.0

7.3 Header Field Names and Assigned Numbers

The Table 12 contains the field name assignments.

© 2004 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Name	Assigned Number
Bcc	0x01
Cc	0x02
X-Mms-Content-Location	0x03
Content-Type	0x04
Date	0x05
X-Mms-Delivery-Report	0x06
X-Mms-Delivery-Time	0x07
X-Mms-Expiry	0x08
From	0x09
X-Mms-Message-Class	0x0A
Message-ID	0x0B
X-Mms-Message-Type	0x0C
X-Mms-MMS-Version	0x0D
X-Mms-Message-Size	0x0E
X-Mms-Priority	0x0F
X-Mms-Read-Report	0x10
X-Mms-Report-Allowed	0x11
X-Mms-Response-Status	0x12
X-Mms-Response-Text	0x13
X-Mms-Sender-Visibility	0x14
X-Mms-Status	0x15
Subject	0x16
To	0x17
X-Mms-Transaction-Id	0x18
X-Mms-Retrieve-Status	0x19
X-Mms-Retrieve-Text	0x1A
X-Mms-Read-Status	0x1B
X-Mms-Reply-Charging	0x1C
X-Mms-Reply-Charging-Deadline	0x1D
X-Mms-Reply-Charging-ID	0x1E
X-Mms-Reply-Charging-Size	0x1F
X-Mms-Previously-Sent-By	0x20
X-Mms-Previously-Sent-Date	0x21

Table 12. Field Name Assignments

© 2004 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Likewise, a brief recap of just the MMS **<u>Message Format</u>** for MMS Specification 1 is shown below. *See* Exh. 2008 form Page 48 to 49:

MMS Specification version 1 Message Format

A.1.2 Sending of Multimedia Message

Item	Function	Reference	Status	Requirement
MMSE-C-016	X-Mms-Message-Type field	Table 1,2	M	
MMSE-C-017	X-Mms-Transaction-ID field	Table 1,2	M	
MMSE-C-018	X-Mms-MMS-Version field	Table 1,2	M	
MMSE-C-019	Date field	Table 1	O	
MMSE-C-020	From field	Table 1	M	
MMSE-C-021	To field	Table 1	O	
MMSE-C-022	Cc field	Table 1	O	
MMSE-C-023	Bcc field	Table 1	O	
MMSE-C-024	Support for at least one To, Cc or Bcc field	Table 1	M	MMSE-C-021 OR MMSE-C-022 OR MMSE-C-023
MMSE-C-025	Subject field	Table 1	O	
MMSE-C-026	X-Mms-Message-Class field	Table 1	O	
MMSE-C-027	X-Mms-Expiry field	Table 1	O	
MMSE-C-028	X-Mms-Delivery-Time field	Table 1	O	
MMSE-C-029	X-Mms-Priority field	Table 1	O	
MMSE-C-030	X-Mms-Sender-Visibility field	Table 1	O	
MMSE-C-031	X-Mms-Delivery-Report field	Table 1	O	

© 2004 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

MMSE-C-032	X-Mms-Read-Report field	Table 1	O	
MMSE-C-033	Content-Type field	Table 1	M	
MMSE-C-034	X-Mms-Response-Status field	Table 2	M	
MMSE-C-035	X-Mms-Response-Text field	Table 2	O	
MMSE-C-036	In the case of application/vnd.wap.multipart.related the presentation is the root part of the aggregate document.	5	M	
MMSE-C-037	Message-ID field in case of successful delivery to MMS Proxy-Relay	Table 2	M	
MMSE-C-038	X-Mms-Reply-Charging field	Table 1	O	
MMSE-C-039	X-Mms-Reply-Charging-Deadline field	Table 1	O	MMSE-C-038
MMSE-C-040	X-Mms-Reply-Charging-Size field	Table 1	O	MMSE-C-038
MMSE-C-041	X-Mms-Reply-Charging-ID field	Table 1	O	

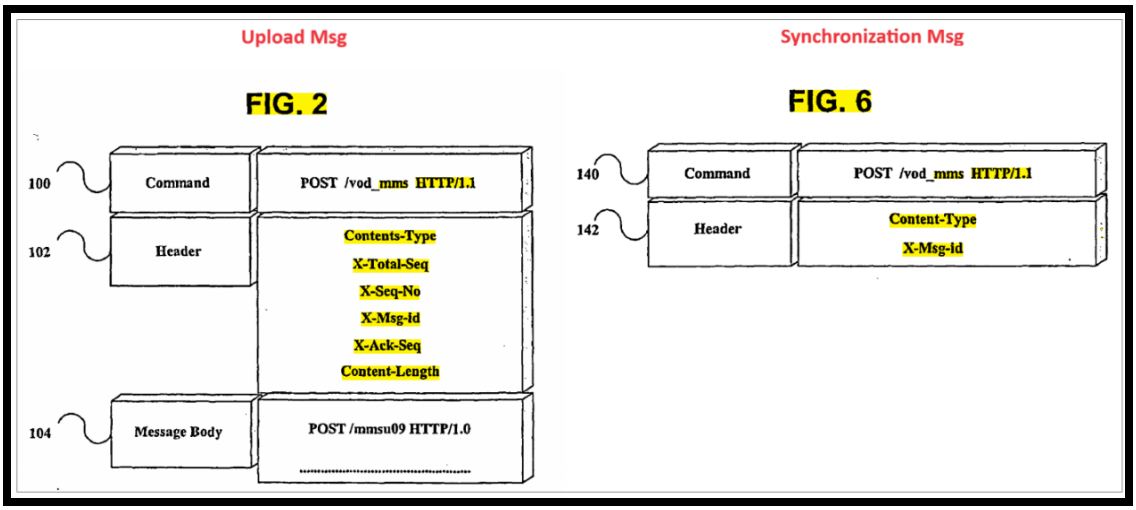
Table 16. Static Conformance Requirement for sent multimedia message and corresponding reply

A.1.3 MMS Notification

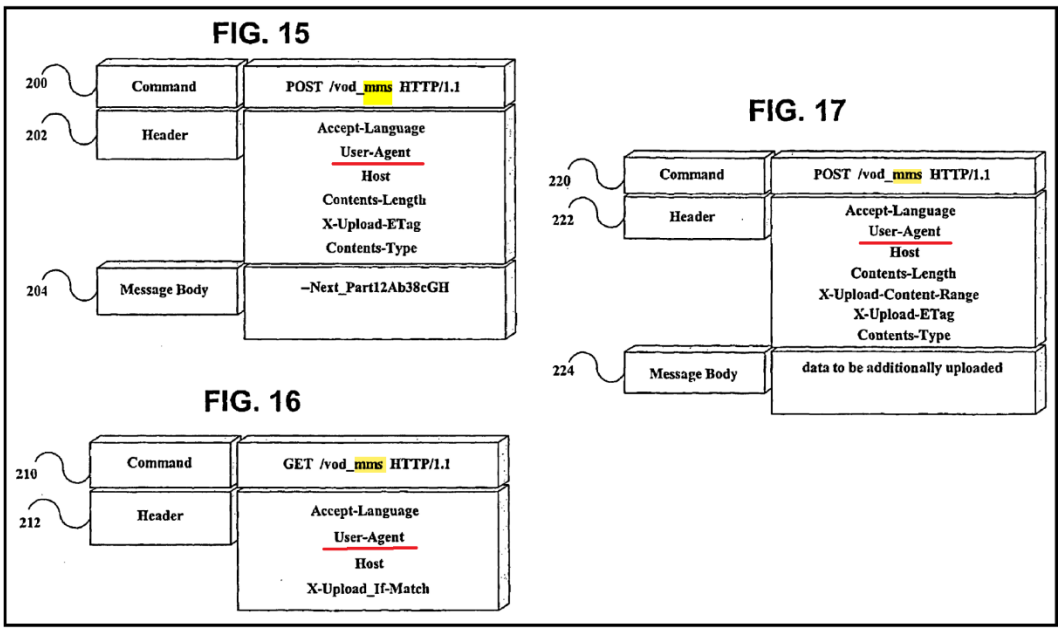
Item	Function	Reference	Status	Requirement
MMSE-C-042	X-Mms-Message-Type field	Table 3,4	M	
MMSE-C-043	X-Mms-Transaction-ID field	Table 3,4	M	
MMSE-C-044	X-Mms-MMS-Version field	Table 3,4	M	
MMSE-C-045	From field	Table 3	O	
MMSE-C-046	Subject field	Table 3	O	
MMSE-C-047	X-Mms-Message-Class field	Table 3	M	
MMSE-C-048	X-Mms-Message-Size field	Table 3	M	

© 2004 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Moreover, none of the MMS upload or synchronization messages disclosed in Na contain any "User ID" or even a "File name." *See* Na (Exh. 1006) at Figs. 2 and 6. More specifically, Na Figure 2 (reproduced below) illustrates the structure of an initial request message for a media data upload, while Figure 6 illustrates the structure of a request message querying the sequence number of the segment from which the upload should be resumed when the connection is closed. There is thus no "User ID" or even a "File name" in Na. *See* Gajendragad [Exh. 2001] at ¶ 106.



Na discloses use of HTTP based "User Agent" in Fig 15, Fig 16 and Fig 17. *See* Gajendragad [Exh. 2001] at ¶ 107.



Na teaches what a User Agent is at [117]: "The header 202 includes information on used languages (Accept-Language, *e.g.,* en, ko), user agent information (User-Agent, *e.g.,* SKT/XXX200)". It would have been clear to a POSITA that the HTTP "User Agent" in Na is NOT a unique user identifier. As of

the date of invention and even today, it was well-known in the art that the "User Agents" as defined by the HTTP protocol were/are not unique per user. Rather, due to the governing HTTP syntax, *literally thousands of individuals at any given point in time could have the exact same HTTP User Agent.* More specifically, as defined by the HTTP protocol, the "User Agent" simply identified the browser name and/or version and the PC or Mac version being utilized by the user (*see* image below). Such "User Agent" identifiers were not unique in any sense of the word, and would not have been understood by a POSITA as satisfying the "User Identifier" as recited in the Challenged Claims. *See* Gajendragad [Exh. 2001] at ¶ 108.

Syntax

HTTP

```
User-Agent: <product> / <product-version> <comment>
```

Common format for web browsers:

HTTP

```
User-Agent: Mozilla/5.0 (<system-information>) <platform> (<platform-details>)
<extensions>
```

Examples

```
Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:47.0) Gecko/20100101 Firefox/47.0
Mozilla/5.0 (Macintosh; Intel Mac OS X x.y; rv:42.0) Gecko/20100101 Firefox/42.0
```

```
Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/51.0.2704.103 Safari/537.36
```

See https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/User-Agent.

F. Discussion of Lind

Lind is merely cited in the Petition in support of Bluetooth capability of the mobile device. Lind has not been relied upon by Petitioner for any other claim language.

V. SPECIFIC REBUTTAL TO ALLEGED OBVIOUSNESS COMBINATIONS

Petitioner has not identified any alleged anticipatory references under 35 U.S.C. § 102. Petitioner has, however, identified a number of combinations which are identified as purportedly comprising invalidating disclosures under 35 U.S.C. § 103. Patent Owner disagrees with the conclusions drawn by Dr. Patrick Traynor, and Dr. Gajendragad expressly finds that none of the identified art, either alone or in combination, renders any Asserted Claim invalid. *See* Gajendragad [Exh. 2001] at ¶ 110.

A. Ground One Fails (Aaltonen and Drescher)

Petitioner and its expert suggest suggests that a POSITA would have been motivated to combine the teachings of Aaltonen and Drescher in order to arrive at the inventions of the Asserted Claims. For a variety of reasons, Patent Owner disagrees.

With respect to teachings of Aaltonen and Drescher, Dr. Patrick Traynor provides a claim-by-claim articulation of his thoughts regarding how the Aaltonen reference combined with Drescher reference render the asserted Cellspin patents

obvious. Again, patent Owner disagrees, and herein addresses the most glaring deficiencies in Petitioner's analysis. *See* Gajendragad [Exh. 2001] at ¶ 112.

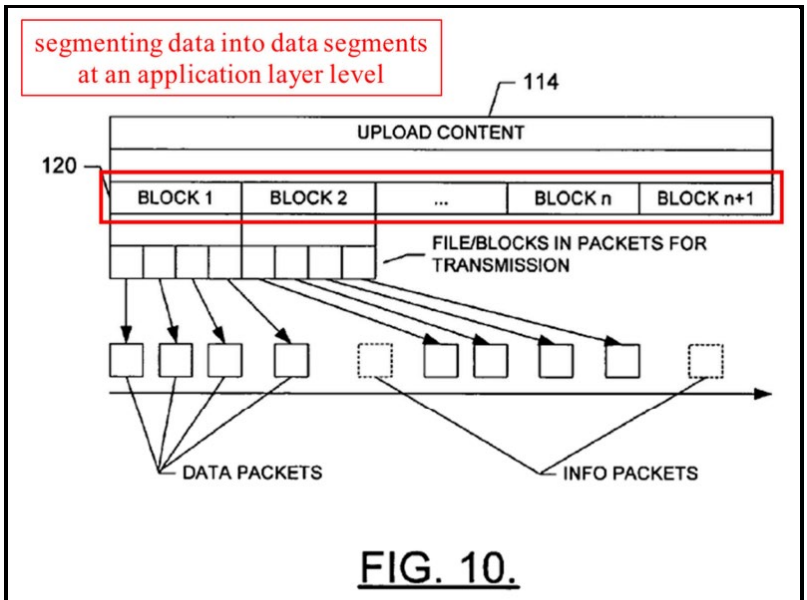
Both Aaltonen and Drescher and their combination teach a much different architecture than those of the Asserted Claims. *See* Gajendragad [Exh. 2001] at ¶ 113. Moreover, if the Board properly construes the two terms that both Petitioner and Cellspin have agreed to at the District Court in their 4-2 disclosures, namely: "user identifier" and "segment identifier" as shown below, both Aaltonen and Drescher and their combination do NOT teach or meet the required claim construction. *See* Gajendragad [Exh. 2001] at ¶ 114.

	Claim Term	**Proposed Constructions**
	"user identifier"	"a text string that identifies uniquely a user of a web service and contains sufficient information to authenticate a user of the web service"
	"segment identifier"	"a text string that identifies uniquely a data segment"

Even assuming a construction in which both identifiers have to be at least a unique *"text string"* without any authentication requirements, neither Aaltonen or Drescher disclose any such identifiers. *See* Gajendragad [Exh. 2001] at ¶ 115.

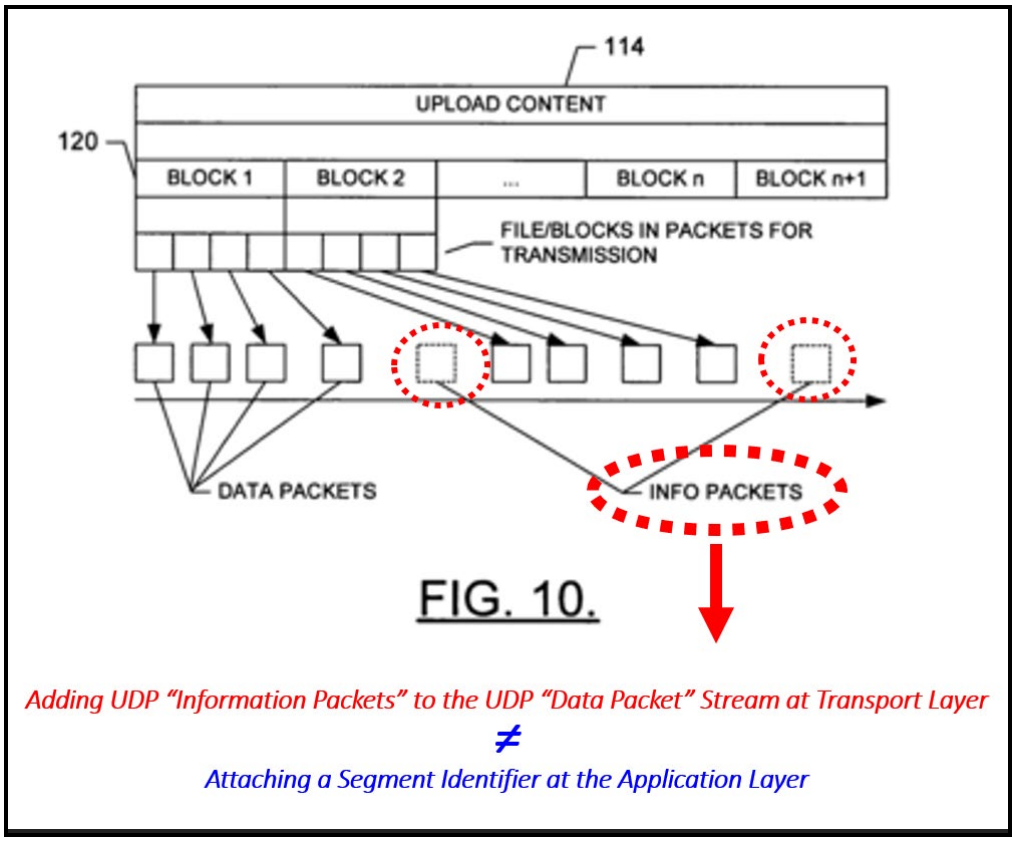
First, and with respect to Element 1[d]: *"applying a segment identifier to each data segment at the application layer level,"* Petitioner relies on Aaltonen "Information Packets" as segment identifiers. Dr. Patrick Traynor tries to explain (using Fig 10 of Aaltonen) that data is divided into BLOCKS at the application

layer and highlights the BLOCKs as segments. *See* Gajendragad [Exh. 2001] at ¶ 116.



TTI-1005, FIG. 10.

However, when identifying the purported segment identifiers in Aaltonen, Dr. Patrick Traynor could not find any such disclosure in Aaltonen of any kind of "BLOCK Identifiers" that were being attached to the BLOCKs at the application layer. *See* Gajendragad [Exh. 2001] at ¶ 117. Recognizing that there is no mention of BLOCK Identifiers in Aaltonen, Dr. Patrick Traynor then seeks to rely on "INFO PACKETS" as poor surrogates for segment identifiers, which they are not, and both of them are completely different and serve different purposes and operate at different layers of the OSI stack. *See* Gajendragad [Exh. 2001] at ¶ 118.

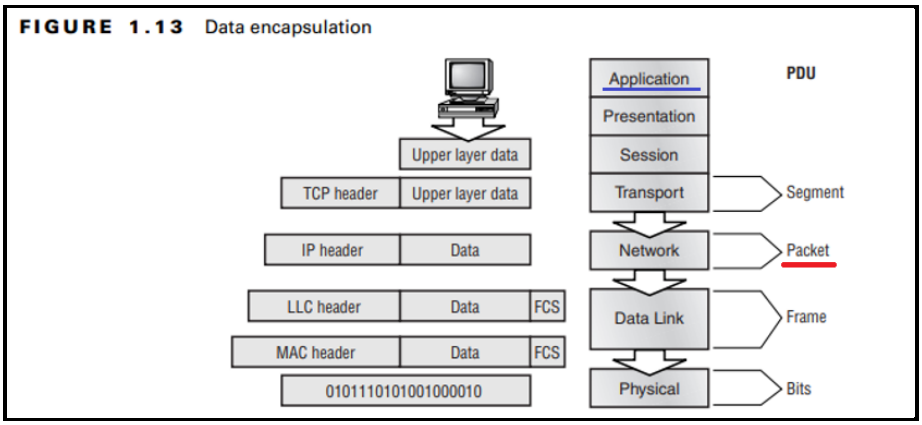


It is evident from the FIG 10 and the text in Aaltonen **[102],** Aaltonen **[103]** and Aaltonen **[104]** that INFO PACKETS are UDP packets at the *Transport layer* are **not** attached to BLOCKs at the Application Layer. Rather, Aaltonen that makes it crystal clear that INFO PACKETS are UDP packets. *See* Gajendragad [Exh. 2001] at ¶ 119.

> **[0102]** During the upload session, between groups or blocks **120** of one or more packets**, <u>an information packet can be added to the packet stream</u>**, as shown in **FIG. 10.**

> **[0103]** In accordance with various protocols, such as the **<u>User Datagram Protocol (UDP), packets</u>** may arrive at the upload agent 110 in an order different from the order the sender 102 uploaded the packets. Conventionally, however, **<u>packets in accordance with protocols such as UDP</u>** do not include information relating to the Sequence of the packets or the total number of data packets that comprise the upload content 114.

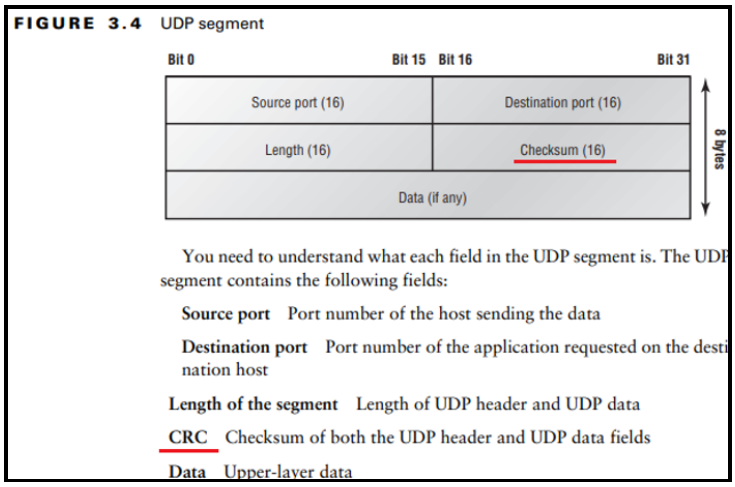
It should be noted that "packets" are at a different OSI layer. Adding "information packets" to the "packet stream" is a function of Network Layer and UDP at the Transport Layer. The claims, however, expressly require applying ***a segment identifier to each data segment at the application layer level.*** *See* Gajendragad [Exh. 2001] at ¶ 120.



Aaltonen **[0102]** teaches: "During the upload session, between groups or blocks **120** of one or more packets**, an information packet can be added to the packet stream**, as shown in **FIG. 10."** *See* Gajendragad [Exh. 2001] at ¶ 121. A POSITA would understand the fact that one can only add standalone UDP packets to the UDP packet stream. *See* Gajendragad [Exh. 2001] at ¶ 122.

Aaltonen **[0103]** teaches: "In accordance with various protocols, such as the **User Datagram Protocol (UDP), packets** may arrive at the upload agent 110 in an order different from the order the sender 102 uploaded the packets. Conventionally, however, **packets in accordance with protocols such as UDP** do not include information relating to the Sequence of the packets or the total number of data packets that comprise the upload content 114." *See* Gajendragad [Exh. 2001] at ¶ 123. A POSITA would understand that independent and standalone **transport layer** UDP based "Information packets" is completely different from Segment Identifiers Attached to a data segment at the application layer. *See* Gajendragad [Exh. 2001] at ¶ 124.

Another disclosure of Aaltonen shows that "Information Packets" are UDP at the transport layer. More specifically, **Aaltonen [0103]** states that *"each information packet **can** include a table including the CRCs"* just like UDP does. *See* Gajendragad [Exh. 2001] at ¶ 125.



Aaltonen further teaches that "Information packets" are completely different from Segment Identifiers. Aaltonen **[0104]** states that "However, the upload agent need **not** maintain all of the received information packets in storage." *See* Gajendragad [Exh. 2001] at ¶ 126.

It is evident to any POSITA that information packets are **not** similar to segment identifiers, because the disclosure is clear that "upload agent need not maintain all of the received information packets in Storage" This is not possible in the case of segment identifiers, because for segment identifiers attached to each data segment, the receiving server **has to** maintain all the segment identifiers and data segments to make sure the file can be reassembled. *See* Gajendragad [Exh. 2001] at ¶ 127.

Aaltonen **[0104]** further states that "For example, the upload agent can be configured to **store only one or more of the latest received information packets**." Again, information packets are **not** like segment identifiers, because in

segment identifiers the **file cannot** the reassembled with just storing only one or more of the latest segment identifiers. This approach is antithetical to the very essence of attaching segment identifiers for each segment. Once again it is quite evident that UDP transport level information packets are NOT the same as segment identifiers at the application level as required by the Asserted Claims. *See* Gajendragad [Exh. 2001] at ¶ 128.

Moreover, since Aaltonen **[0104]** teaches that "information packets" can be discarded or not maintained, that means they are two separate entities. Segment identifiers are not attached/applied to the Data Segments in Aaltonen. *See* Gajendragad [Exh. 2001] at ¶ 129.

Petitioner alleged that: "identifying information would ***preferentially*** be added for or to "each" block, as Aaltonen relies upon this information to "monitor the received… blocks of packet(s) or portion of the upload content," determine if an interruption has been experienced, and "facilitate recovery of the upload session" following an interruption." *See* Gajendragad [Exh. 2001] at ¶ 130. Petitioner itself recognizes the fact that Aaltonen does not disclose any segment identifiers or segment ID. That is the reason Petitioner uses the language "would **preferentially** be added." It is pure speculation to say "preferentially," and moreover it is incorrect. *See* Gajendragad [Exh. 2001] at ¶ 131.

Petitioner selectively extracts text from Aaltonen [0105] out of context. A POSITA, however, would appreciate the totality of the disclosure of Aaltonen [0105], which is excerpted and highlighted below (*see* Gajendragad [Exh. 2001] at ¶ 132)**:**

> [**0105**] During the upload session, then, the upload agent **110** can utilize the information packets to monitor the received data packets. More particularly, the upload agent can monitor the received data packets to track the number and accuracy of the received packets, blocks of packet(s) or portion of the upload content **114**. In this regard, the number of received packets, blocks **120** of packet(s) or portion of the upload content can be compared to an identified number of packets in or size of the upload content at any one or more points during the upload session. The accuracy of the received packets, blocks of packet(s) or portion of the requested content can be verified in any of a number of different manners, such as by utilizing CRC data in each of the packets. Similarly, a CRC or any other characteristic of accuracy can be included in each transmitted packet, block of packet(s) or portion of the upload content.

This disclosure is clearly talking about the transport layer UDP "packets" and behavior and the CRC (Checksum) of the "packets" and verifying using the accuracy of the CRC. None of this text has anything to do with Segment Identifiers or, in the language of Aaltonen, "Block Identifiers." Aaltonen does not teach or have any application layer segment identifiers or BLOCK identifiers attached to the data segment at the application layer. *See* Gajendragad [Exh. 2001] at ¶ 133.

Lastly, and as discussed above, claim construction must (at minimum) require a "segment identifier" to be a "a text string" that identifies uniquely a data segment. *See* Gajendragad [Exh. 2001] at ¶ 134.

Petitioner relies on the INFO PACKET of Aaltonen as a "segment identifier," but "INFO PACKET" does not satisfy the claim construction that requires "segment identifiers to be a text string." That is because INFO PACKETS are UDP Packets and it is well know that Data in UDP (User Datagram Protocol) packets **is in binary format**. It is well known that the data payload within a UDP packet is in binary form, which allows for efficient and compact transmission of information. *See* Gajendragad [Exh. 2001] at ¶ 135.

Below is an example of a UDP packet in binary format (*see* Gajendragad [Exh. 2001] at ¶ 136):

UDP Packet Structure:

Source Port (16 bits)
Destination Port (16 bits)
Length (16 bits)
Checksum (16 bits)
Data (variable length)

Example:

An exemplary UDP packet would have the following values:

Source Port: 12345 (0x3039)
Destination Port: 80 (0x0050)
Length: 8 bytes header + 4 bytes data = 12 (0x000C)
Checksum: 0x1A2B

Data: “Test” (ASCII values: 0x54 0x65 0x73 0x74)

The binary representation of this UDP INFO packet would be:

```
Source Port:      0011 0000 0011 1001 (0x3039)
Destination Port: 0000 0000 0101 0000 (0x0050)
Length:           0000 0000 0000 1100 (0x000C)
Checksum:         0001 1010 0010 1011 (0x1A2B)
Data:             0101 0100 0110 0101 0111 0011 0111 0100 (0x54657374)
```

Complete Binary UDP Packet would be:

0011 0000 0011 1001 0000 0000 0101 0000 0000 0000 0000 1100 0001

1010 0010 1011 0101 0100 0110 0101 0111 0011 0111 0100

This binary sequence represents the entire UDP packet, including the header and data.

Binary data in UDP INFO Packets do NOT meet the claim construction required for the “segment identifier.” *See* Gajendragad [Exh. 2001] at ¶ 136.

Petitioner has NOT shown that the INFO Packet is a text string that identifies uniquely a data segment. *See* Gajendragad [Exh. 2001] at ¶ 137.

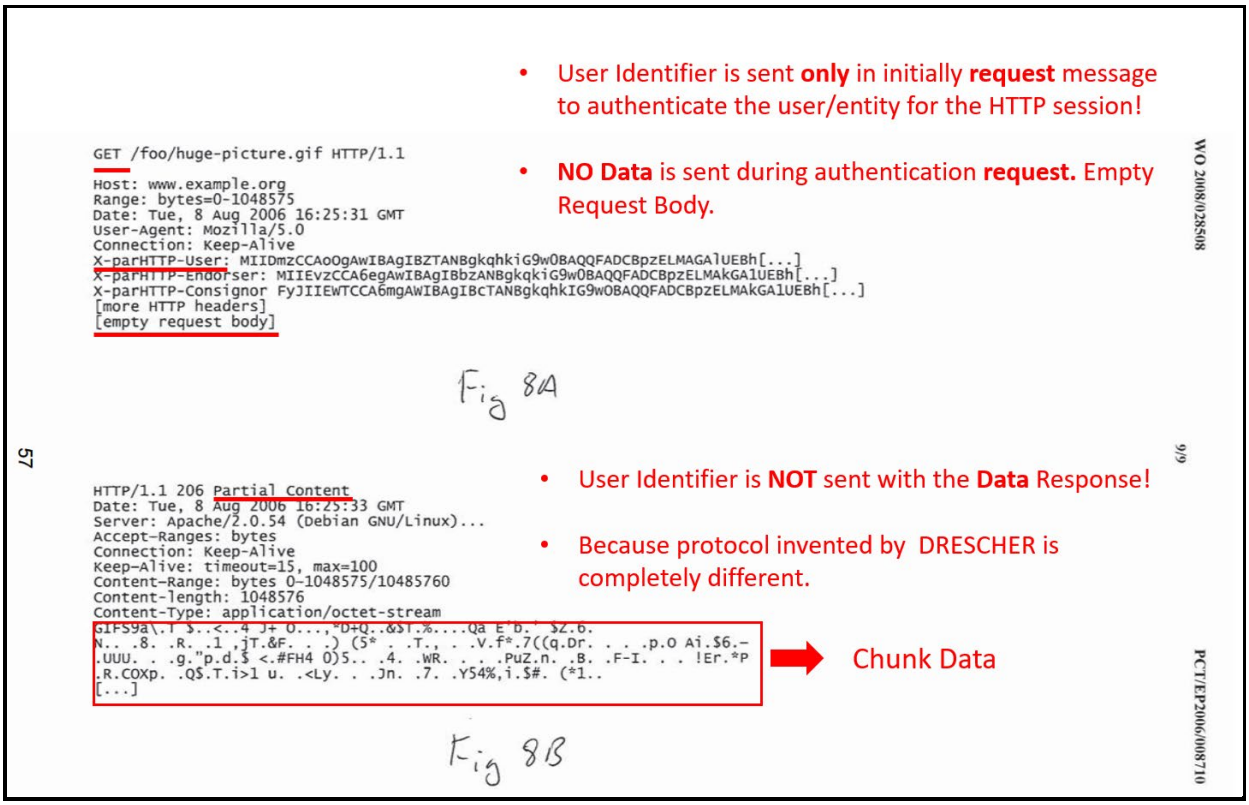
Second, and with respect to Element **1[e]:** “applying a user identifier to each data segment for each destination web service at the application layer level, wherein each user identifier uniquely identifies a particular user of the web service,” Petitioner makes passing reference to Aaltonen but then directly relies upon Drescher to apply a user identifier for each data segment. *See* ‘336 Petition at page 17 (also ‘757 Petition at page 19, and ‘260 Petition at page 18); *see also* Gajendragad [Exh. 2001] at ¶ 138:

identifying the sender 102." TTI-1005, ¶¶[0101]-[0102], [0104]. Moreover, Aaltonen uploads content "in a number of content data packets" "to the recipient in accordance with an HTTP POST or HTTP PUT technique," and according to Drescher, HTTP PUT and POST messages have a user identifier disclosing "a name particularising the… [party] involved in the request" ("***applying a user identifier***" that "***uniquely identifies a particular user of the web service***"). TTI-1005, [0101], [0111]; TTI-1010, 25:32-26:21, 34:13-35:4, 35:26-36:21. Drescher discloses that this information is preferentially added "for each chunk, and… PUT" (each of the data packets of the Aaltonen-Drescher combination). *Id*. Also, at least because Drescher teaches that user identity is used to determine "permissions" to allow "access to resources at the server" (the recipient in the Aaltonen-Drescher combination), this information is "***for the destination web service***." TTI-1010, 25:4-25, 34:13-35:4, 35:26-36:21. Because each data packet within a data block includes this information, "***each***" data block ("***each data segment***") would have included this information. *See supra* 1[c]-1[d]; TTI-1003, ¶55.

Petitioner's assertion is incorrect: Drescher does NOT send user identifier with data chunks. *See* Gajendragad [Exh. 2001] at ¶ 139..

Drescher Fig 8A and Fig 8B clearly show that the User ID is **ONLY** sent in the initial request message to authenticate the user for that HTTP Session. There is NO Data in the request body (*i.e.,* Empty request body). After the **Fig 8A** HTTP request for authentication is successful, in the Data response message (**Fig 8B)** the

User ID is NOT sent along with the Chuck Data. This is because parHTTP protocol is completely different. *See* Gajendragad [Exh. 2001] at ¶ 140.



Below is the text from Drescher describing **Fig 8A** and **Fig 8B**:

> Figures 8A shows the text of the headers of a typical ParHTTP request, which shows both headers for transferring the certificates and the range header. As noted above in some embodiments of the invention either of those may be omitted. The text of the Figure has some annotations in square brackets; "[…]" indicates more similar data.
>
> Figure 8B sows the response to the request. The data after "Content-Type: application/octet-stream" is binary and has been shown as random characters for the purpose of illustration.

The accompanying text shown above also makes it clear that transferring certificate for user ID / authentication is *only done in the initial ParHTTP request* and not in the response when data is being transferred. *See* Gajendragad [Exh. 2001] at ¶ 142.

Below is the whole paragraph from Drescher, which Petitioner relies upon for the "Attaching user identifier" of Element **1[e]** ("applying a user identifier to each data segment for each destination web service at the application layer level") (*see* Gajendragad [Exh. 2001] at ¶ 143):

The secure version of ParHTTP provides, through the establishment of secure sockets as described above, a means of transferring one identity from the ParHTTP client to the

25

WO 2008/028508 PCT/EP2006/008710

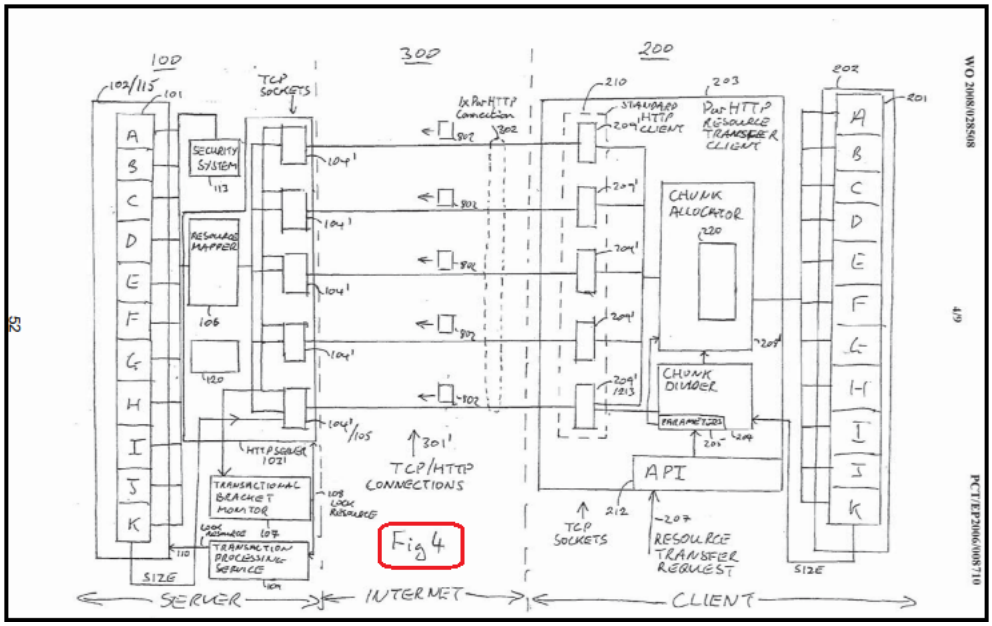
25

server. ParHTTP also provides a second method of communicating identities between the client 200 and the server 100. This is also implemented in the example of Figure 4 but may be used irrespective of whether the connection is secured. In this method the identity provider 220 adds to each HTTP request 802 sent to the server (i.e. for each chunk, and for PUT and GET) a header for each identity to be transferred. The header is called in the request "X-parhttp-<entity(i)>", where <entity(i)> is a name particularising the role of the ith entity, i.e. party, involved in the request. (In the later ETD example those roles are named "user", "endorser" etc., thus making the name of the header "X-parhttp-user" etc.. The information contained in the

Petitioner argues that "according to Drescher, HTTP PUT and POST messages have a user identifier disclosing "a name particularising the… [party] involved in the request" ("applying a user identifier" that "uniquely identifies a particular user of the web service")." However, Petitioner is incorrect, because Drescher does NOT do it for POST messages. Nowhere in Drescher is there any teaching that HTTP POST messages have user identifiers. The word "POST" itself

does not appear in Drescher. Again, it is clear from the paragraph shown above that it is ONLY done for **HTTP Request** (Fig 8A) and **not** response (Fig 8B) when chunk data is being sent. *See* Gajendragad [Exh. 2001] at ¶ 144.

The reason Drescher says for "each chunk" is because it **is talking about ParHTTP (Parallel HTTP)**, and **Fig 4** (shown below) that has **five** (5) parallel HTTP connections sending different chunks and each of those five (5) HTTP connections, which have to be authenticated with a separate HTTP **request** with User ID (Fig 8A) before the chunk data can be sent in the HTTP **response (Fig 8B)**, and that HTTP response does **NOT** contain any user identifier (as clearly shown in Fig 8B) (*see* Gajendragad [Exh. 2001] at ¶ 145):



In Drescher, User Identity is sent for each parallel HTTP *connection request* to be used by each chunk, BUT user identity is **NOT attached** or sent with any

chunk **data**. Rather, User Identity is used **only initially** for each parallel HTTP connection authentication request **Fig 8A** and **Fig 8B.** *See* Gajendragad [Exh. 2001] at ¶ 146.

The passing mention of Aaltonen as support for "user identifier" (*see* '336 Petition page 16-17 (also '757 Petition at page 18, and '260 Petition at page 16-17)) is insufficient (*see* Gajendragad [Exh. 2001] at ¶ 147):

Element 1[e] [4]**.**

Aaltonen discloses that "information packets… added to the packet stream" "between groups or blocks 120 of one or more packets" or alternatively "data packets" within its HTTP data "blocks" ("***data segments***") include "identifiers

16

Attorney Docket No. 50048-0057IP1
IPR of U.S. Patent No. 8,756,336

identifying the sender 102." TTI-1005, ¶¶[0101]-[0102], [0104]. Moreover,

With respect to Element **1[e]**, Aaltonen does not help the Petitioner because the **only** time Aaltonen even remotely mentions anything to do with "user identifier" is in Aaltonen [102], as follows: "each *information packet* can include information (*e.g.,* Sequence of packet CRCs) uniquely describing the packets before or after the information packet, as well as one or more identifiers identifying the Sender 102." Important point to NOTE is that this **sender information is at the *UDP information packet level***, and it is talking about IP

address and destination port number of Sender 102. *See* Gajendragad [Exh. 2001] at ¶ 148.

UDP Checksum is *Optional* is also called CRC (*cyclic redundancy codes)*

UDP Packets

```
 0      7 8     15 16    23 24    31
+--------+--------+--------+--------+
|     Source      |   Destination   |
|      Port       |      Port       |
+--------+--------+--------+--------+
|                 |                 |
|     Length      |    Checksum CRC|
+--------+--------+--------+--------+
|
|          data octets ...
+---------------- ...

      User Datagram Header Format
```

That is why Aaltonen says "*one or more identifiers* identifying the Sender 102." These are IP addresses and source and destination ports inside the UDP packet. It would have been readily understood by a POSITA that these IP address and Source and Destination Port numbers inside the UDP packets are **NOT** unique to the web service because the internet and intranet uses NAT (Network Address Translation) protocols at the network boundaries and UDP packet IP addresses get swapped at the network edges. *See* Gajendragad [Exh. 2001] at ¶ 149.

Further, a POSITA would understand that Aaltonen UDP INFO packets which include "one or more identifiers identifying the Sender 102" are also encoded in *Binary format*. It is NOT a text string. *See* Gajendragad [Exh. 2001] at ¶ 150.

Thus, the sender information inside the UDP INFO packet does **not** satisfy even the basic claim construction required for "user identifier." That is, it being a

"Text String" and it being Unique at the web service, and it is *not* being applied at the Application layer level. *See* Gajendragad [Exh. 2001] at ¶ 151.

Third, and with respect to **Element 1[j],** Petitioner relies on Aaltonen **Fig 11** (Head URI Request) for "sending, a synchronization request to the web service, for **said user identifier**" (*see* '336 Petition at page 24 (also '757 Petition at page 27, and '260 Petition at page 25)) (*see also* Gajendragad [Exh. 2001] at ¶ 152):

Attorney Docket No. 50048-0057IP1
IPR of U.S. Patent No. 8,756,336

recipient following an interruption. TTI-1003, ¶64.

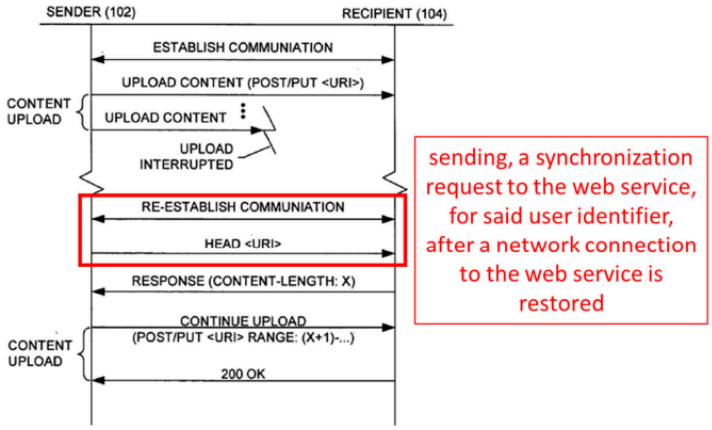


FIG. 11.

TTI-1005, FIG. 11.

Petitioner argues "Aaltonen discloses that" (Element 1[j]) and then incorrectly asserts that this HEAD <URI> "request from the sender **<u>would have been</u>** "for said user identifier," for example, because the interrupted upload is associated with said user identifier." This statement is incorrect, and is unsupported by any teaching of Aaltonen. The reason Petitioner cannot back this assertion with

any “user identifier” data in Aaltonen, is because Aaltonen does NOT have a “user identifier” “wherein each user identifier uniquely identifies a particular user of the web service.” *See* Gajendragad [Exh. 2001] at ¶ 153.

As such, Petitioner resorts to Drescher for Element **1[e]** “applying a user identifier to each data segment.” Drescher has the User identifier that it uses only initially for the authentication of the HTTP session. Having conceded that Aaltonen does not have user identifier but Drescher does, Petitioner cannot then turn around and assert that the Aaltonen **Fig 11** HEAD <URI> request is user identifier which uniquely identifies a particular user of the web service. *See* Gajendragad [Exh. 2001] at ¶ 154.

Moreover, it is clear from Aaltonen [0111] and [0112] (discussing the control flow diagram of FIG. 11), that URI Stands for Uniform Resource Identifier (URI) and URI is not a User Identifier. Aaltonen **[0111]** states: “where messages in accordance with each include a header **identifying the URI of a location** within **content storage 112.”** *See* Gajendragad [Exh. 2001] at ¶ 155.

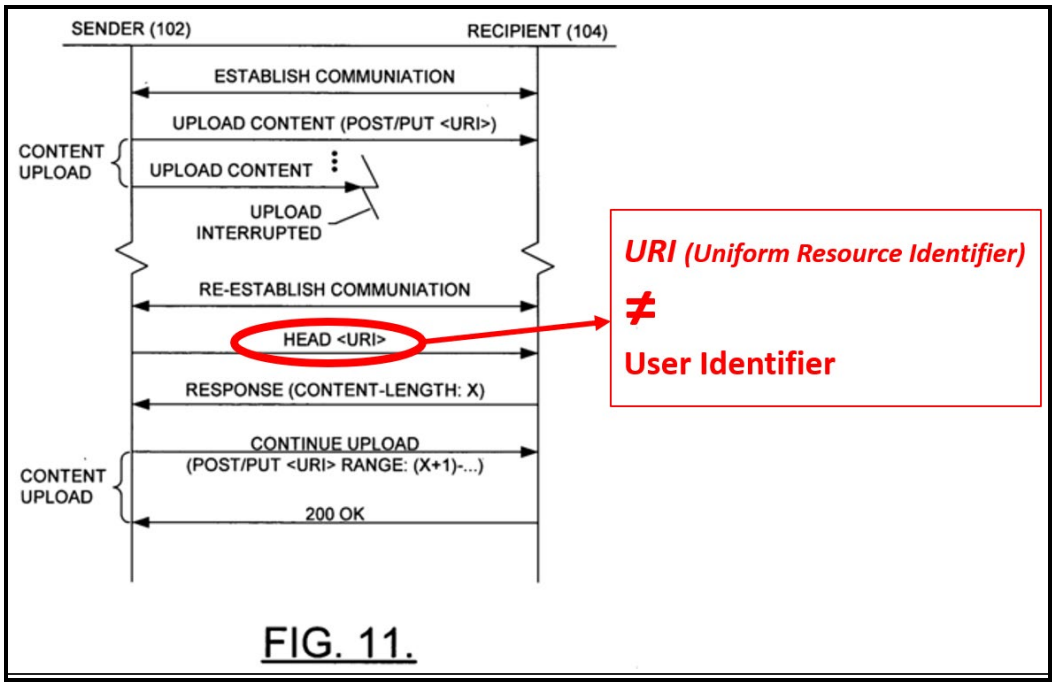


FIG. 11.

Aaltonen **[0112]** teaches: **"**where the *request includes* the **URI of the location** within **content storage 112** including the received portion of the upload content." A POSITA would understand that the Head request in Aaltonen is NOT for a user identifier and the URI is of the Location within the content storage. *See* Gajendragad [Exh. 2001] at ¶ 156.

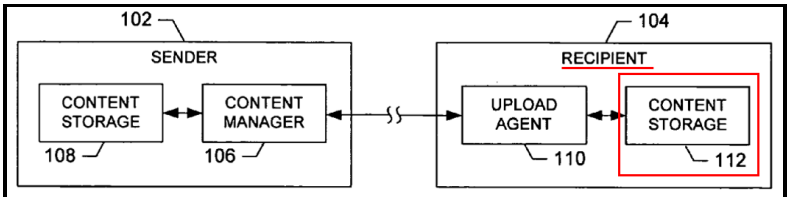


FIG. 6.

Aaltonen **Fig 6** as shown above makes it clear that the **location of recipient content storage 112 (URI)** has nothing to do with the user ID. *See* Gajendragad [Exh. 2001] at ¶ 157.

Also, for Element **1[e]** ("user identifier"), Petitioner relies on Aaltonen [0102] "information packets" containing "identifiers containing the sender" as follows:

Element 1[e] [4].

Aaltonen discloses that "information packets… added to the packet stream" "between groups or blocks 120 of one or more packets" or alternatively "data packets" within its HTTP data "blocks" ("***data segments***") include "identifiers

16

Attorney Docket No. 50048-0057IP1
IPR of U.S. Patent No. 8,756,336

identifying the sender 102." TTI-1005, ¶¶[0101]-[0102], [0104]. Moreover,

However, in its Petition for **Element 1[j],** Petitioner relies on Aaltonen **Fig 11** (Head URI Request) for "sending, a synchronization request to the web service, for **said user identifier**" (*see* '336 Petition page 23-24 (also '757 Petition at page 27, and '260 Petition at page 25)). *See* Gajendragad [Exh. 2001] at ¶¶ 158-159.

The Petition fails on the related basis of Petitioner's ***inconsistent*** purported disclosures of the recited "user identifier" throughout the Petition. Especially "user identifier" with respect to Element **1[e]** and Element **1[j],** Petitioner relies on

Aaltonen "Head URI" for Element 1[j] and "information packet" "identifier identifying the sender 102" for Element **1[e]** for the purported "user identifier." *See* Gajendragad [Exh. 2001] at ¶ 160.

Petitioner relies on Drescher for "user identifier" in Element **1[e]** because Aaltonen does **not** have a user web site / webservice. For user web services with multiple users, Petitioner relies on "recipient" that can be an "origin server 22," "digital broadcaster 32," and/or "digital broadcast receiver 36" ("one or more web services"). But Aaltonen **[0035]** clearly states that origin server is **not** a user website: "For example, **origin servers** can provide content to the terminal, such *as in accordance with the Multimedia Broadcast* ***Multicast*** *Service (**MBMS**).".* *See* Gajendragad [Exh. 2001] at ¶ 161.

A POSITA would understand that HTTP itself is **not** inherently designed to work with multicast, as HTTP is a unicast protocol, meaning it is designed for one-to-one communication between a client and a server. Even up until 2020 there were no means of using HTTP with Multicast. One approach (introduced after 2020) is using protocols like HTTP over multicast QUIC. This method leverages the QUIC transport protocol to facilitate the transfer of HTTP resources over multicast IP, maintaining compatibility with **HTTP/3 that was introduced in 2021.** *See* https://www.ietf.org/archive/id/draft-pardue-quic-http-mcast-09.html. *See* Gajendragad [Exh. 2001] at ¶ 162.

Moreover, Aaltonen **[0039]** clearly states that "digital broadcaster 32," and/or "digital broadcast receiver 36" is ALSO not a user website. Aaltonen **[0039]** discloses: "digital broadcast network, such as a **terrestrial** digital video broadcasting (*e.g.,* DVB-T, DVB-H, ISDB-T, ATSC, etc.) network." None of the **terrestrial** digital video broadcasting technologies mentioned used HTTP for broadcasting, meaning it is not a website or a web service. *See* Gajendragad [Exh. 2001] at ¶ 163.

B. Ground Two Fails (Takahashi and Na and Lind)

Petitioner and its expert suggest that a POSITA would have been motivated to combine the teachings of Takahashi and Na and Lind in order to arrive at the inventions of the Asserted Claims. For a variety of reasons, Patent Owner disagrees.

Both **Takahashi** *(Web based termination where filename contains a username*) and **Na** *(MMS 1.0 Specification with NO User ID or username or filename)* and their combination teaches a much different architecture than that of the Asserted Claims. *See* Gajendragad [Exh. 2001] at ¶ 166.

As discussed above, a POSITA would understand "along with" as "in addition to," and Takahashi does not teach sending any user identifier "along with" the data. Again, and as taught in Takahashi, the User ID (user identifier) is part of the filename, not an identifier transmitted "in addition to" the segment identifier

and the data. As such, any purported obviousness combination including Takahashi fails. *See* Gajendragad [Exh. 2001] at ¶ 167.

In the combination to Takahashi and Na, it is inconsistent to suggest that an element inside the file name is a "user identifier," while a "segment identifier" is outside the filename and sent with each segmented data. *See* Gajendragad [Exh. 2001] at ¶ 168.

The Board should construe the two terms as both Petitioner and Cellspin have agreed to at the District Court (namely, "user identifier" and "segment identifier"). As shown below, both Takahashi and Na and Lind and their combinations do NOT teach or meet the required claim construction in their respective specifications. *See* Gajendragad [Exh. 2001] at ¶ 169.

	Claim Term	**Proposed Constructions**
	"user identifier"	"a text string that identifies uniquely a user of a web service and contains sufficient information to authenticate a user of the web service"
	"segment identifier"	"a text string that identifies uniquely a data segment"

With respect to teachings of Takahashi and Na and Lind, Dr. Patrick Traynor provides a claim-by-claim articulation of his thoughts regarding how the Takahashi reference combined with Na and with Lind (for Bluetooth) render the Asserted Claims obvious. Patent Owner disagrees with Dr. Patrick Traynor's conclusions. *See* Gajendragad [Exh. 2001] at ¶ 170.

First, the combination of Takahashi and Na **are *incompatible* and the combined solution *is not workable*.** *See* Gajendragad [Exh. 2001] at ¶ 171.

Petitioner admits that "Takahashi does **not** appear to disclose details regarding the HTTP protocol and HTTP requests for transmitting divided media data," yet Petitioner goes on to assert that "A POSITA would have been motivated and would have found it obvious and straightforward to *incorporate Na's teachings **in** Takahashi's system.*" *See* '336 Petition page 33 (*see also* '757 Petition at page 43, and '260 Petition at page 43). Petitioner proposes using Takahashi as a base and incorporate the teachings of Na into it. *See id.; see also* Gajendragad [Exh. 2001] at ¶ 172.

Petitioner goes on to confirm that it considers Takahashi **as a base** prior art and Takahashi website server to receive all the commands and messages. *See* '336 Petition page 40: ("A POSITA would have recognized and found obvious that Takahashi's server would have hosted a website"). *See* Gajendragad [Exh. 2001] at ¶ 173.

Petitioner further contends that: "a POSITA also would have been motivated and would have found it obvious and straightforward, based on Na, to implement a mobile phone [of Takahashi] to send to the server a synchronization request querying 'the sequence number of the segment'… [from Na]." *See* '336 Petition at 62-63 (*see also* '757 Petition at page 75, and '260 Petition at page 72). Later,

Petitioner contends that: "A POSITA would have been motivated and would have found it obvious and straightforward to implement Takahashi's mobile phone to receive from the server a synchronization reply corresponding to 'the sequence number of the segment from which the upload should be resumed' or 'the range of the message body that has been successfully uploaded' … **as taught by Na**." *See* '336 Petition at 67 (*see also* '757 Petition at page 75, and '260 Petition at page 72); *see also* Gajendragad [Exh. 2001] at ¶ 174.

The major problem for Petitioner is that the Takahashi teaching is based on normal website upload commands and messages that terminate on connection failure, whereas the teaching of **Na is based on MMS** *(Multimedia Messaging Service)* message Specifications and **every command in Na is sent to the server that includes MMS 1.0 specification information**. For example: Na clearly states that every command includes MMS 1.0 specification information. *See* Na at [0062] ("The <u>command 100 includes MMS 1.0 specification information</u> (*e.g.,* POST/vod_mms HTTP/1.1))"; *see also id.* at [0065] ("The <u>message body 104 includes the first 512 bytes of the MMS 1.0 specification data</u>"); *see id. at* [0069] ("The message body 122 includes a message type field (X-**Mms**-Message-Type, *e.g.,* M-send-conf) and an upload result report field (X-**Mms**-Response-Text, e.g., upload O.K.)"); *see id.* at [0118] ("In FIG. 15, the message body 204 is defined within 64 bytes <u>in accordance with the message body format of the MMS 1.0</u>

specification, which is unique throughout the message"); *see id.* at [0123] ("The message body 226 includes the data to be additionally uploaded in accordance with the message body format of the MMS 1.0 specification"). All disclosed segmenting and messaging is in accordance with the MMS Application Server. *See* Gajendragad [Exh. 2001] at ¶ 175.

As shown below, MMS 1.0 specification information for MMS Protocol Data Units and MMS Header Fields are discussed in great detail in Exh. 2008 form Page 13 to 27. A brief recap of just the MMS **Header Information** for MMS Specification 1 is shown below. Exh. 2008 form Page 42 to 43:

OMA-WAP-MMS-ENC-V1_1-20040715-A Page 42 (68)

Show = <Octet 129>

7.2.32 X-Mms-Status field

Status-value = Expired | Retrieved | Rejected | Deferred | Unrecognised |

Expired = <Octet 128>
Retrieved = <Octet 129>
Rejected = <Octet 130>
Deferred = <Octet 131>
Unrecognised = <Octet 132>
Indeterminate = <Octet 133>
Forwarded = <Octet 134>

The value Unrecognised is reserved for version management purpose only.

7.2.33 Subject field

Subject-value = Encoded-string-value

7.2.34 To field

To-value = Encoded-string-value

See Chapter 8 for addressing model.

7.2.35 X-Mms-Transaction-Id field

Transaction-id-value = Text-string

MMS Specifications 1.0

7.3 Header Field Names and Assigned Numbers

The Table 12 contains the field name assignments.

© 2004 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

OMA-WAP-MMS-ENC-V1_1-20040715-A Page 43 (68)

Name	Assigned Number
Bcc	0x01
Cc	0x02
X-Mms-Content-Location	0x03
Content-Type	0x04
Date	0x05
X-Mms-Delivery-Report	0x06
X-Mms-Delivery-Time	0x07
X-Mms-Expiry	0x08
From	0x09
X-Mms-Message-Class	0x0A
Message-ID	0x0B
X-Mms-Message-Type	0x0C
X-Mms-MMS-Version	0x0D
X-Mms-Message-Size	0x0E
X-Mms-Priority	0x0F
X-Mms-Read-Report	0x10
X-Mms-Report-Allowed	0x11
X-Mms-Response-Status	0x12
X-Mms-Response-Text	0x13
X-Mms-Sender-Visibility	0x14
X-Mms-Status	0x15
Subject	0x16
To	0x17
X-Mms-Transaction-Id	0x18
X-Mms-Retrieve-Status	0x19
X-Mms-Retrieve-Text	0x1A
X-Mms-Read-Status	0x1B
X-Mms-Reply-Charging	0x1C
X-Mms-Reply-Charging-Deadline	0x1D
X-Mms-Reply-Charging-ID	0x1E
X-Mms-Reply-Charging-Size	0x1F
X-Mms-Previously-Sent-By	0x20
X-Mms-Previously-Sent-Date	0x21

Table 12. Field Name Assignments

© 2004 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Further, a brief recap of just the MMS **Message Format** for MMS Specification 1 is shown below. *See* Exh. 2008 form Page 48 to 49:

OMA-WAP-MMS-ENC-V1_1-20040715-A Page 48 (68)

MMS Specification version 1 Message Format

A.1.2 Sending of Multimedia Message

Item	Function	Reference	Status	Requirement
MMSE-C-016	X-Mms-Message-Type field	Table 1,2	M	
MMSE-C-017	X-Mms-Transaction-ID field	Table 1,2	M	
MMSE-C-018	X-Mms-MMS-Version field	Table 1,2	M	
MMSE-C-019	Date field	Table 1	O	
MMSE-C-020	From field	Table 1	M	
MMSE-C-021	To field	Table 1	O	
MMSE-C-022	Cc field	Table 1	O	
MMSE-C-023	Bcc field	Table 1	O	
MMSE-C-024	Support for at least one To, Cc or Bcc field	Table 1	M	MMSE-C-021 OR MMSE-C-022 OR MMSE-C-023
MMSE-C-025	Subject field	Table 1	O	
MMSE-C-026	X-Mms-Message-Class field	Table 1	O	
MMSE-C-027	X-Mms-Expiry field	Table 1	O	
MMSE-C-028	X-Mms-Delivery-Time field	Table 1	O	
MMSE-C-029	X-Mms-Priority field	Table 1	O	
MMSE-C-030	X-Mms-Sender-Visibility field	Table 1	O	
MMSE-C-031	X-Mms-Delivery-Report field	Table 1	O	

© 2004 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

OMA-WAP-MMS-ENC-V1_1-20040715-A Page 49 (68)

MMSE-C-032	X-Mms-Read-Report field	Table 1	O	
MMSE-C-033	Content-Type field	Table 1	M	
MMSE-C-034	X-Mms-Response-Status field	Table 2	M	
MMSE-C-035	X-Mms-Response-Text field	Table 2	O	
MMSE-C-036	In the case of application/vnd.wap.multipart.related the presentation is the root part of the aggregate document.	5	M	
MMSE-C-037	Message-ID field in case of successful delivery to MMS Proxy-Relay	Table 2	M	
MMSE-C-038	X-Mms-Reply-Charging field	Table 1	O	
MMSE-C-039	X-Mms-Reply-Charging-Deadline field	Table 1	O	MMSE-C-038
MMSE-C-040	X-Mms-Reply-Charging-Size field	Table 1	O	MMSE-C-038
MMSE-C-041	X-Mms-Reply-Charging-ID field	Table 1	O	

Table 16. Static Conformance Requirement for sent multimedia message and corresponding reply

A.1.3 MMS Notification

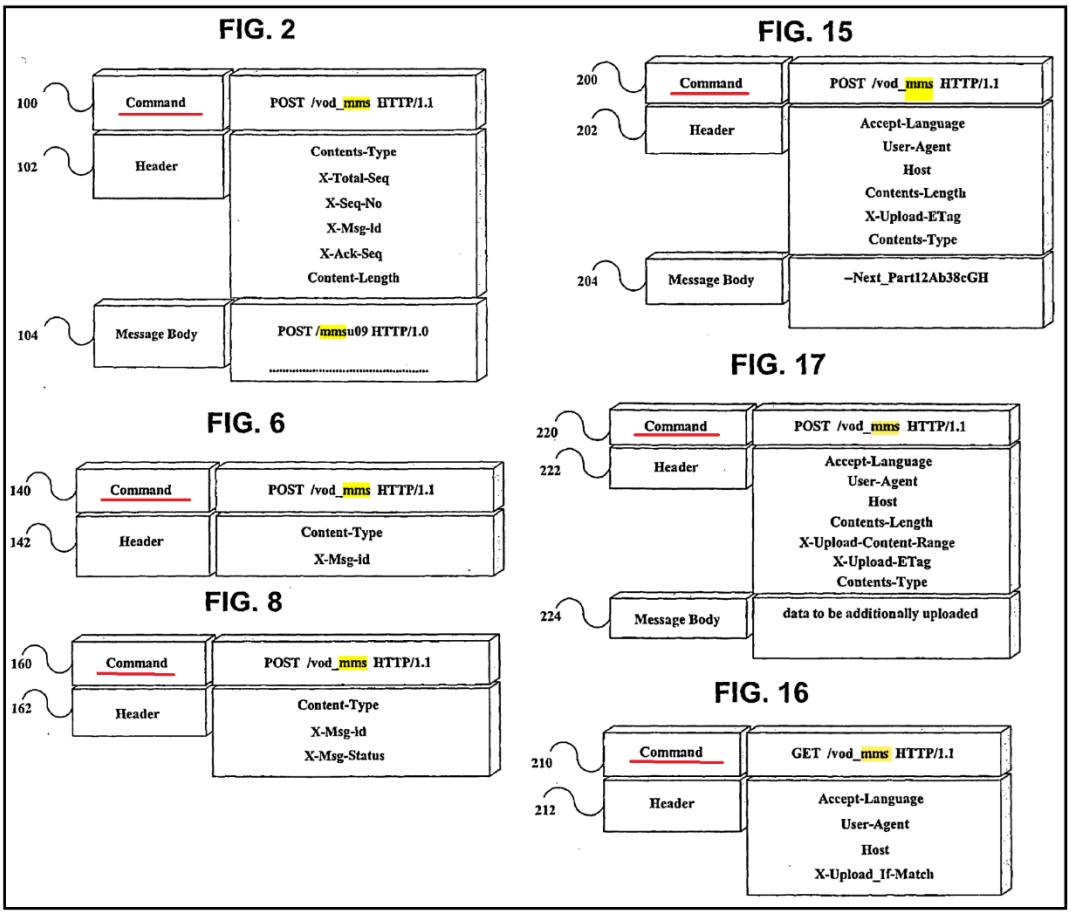
Item	Function	Reference	Status	Requirement
MMSE-C-042	X-Mms-Message-Type field	Table 3,4	M	
MMSE-C-043	X-Mms-Transaction-ID field	Table 3,4	M	
MMSE-C-044	X-Mms-MMS-Version field	Table 3,4	M	
MMSE-C-045	From field	Table 3	O	
MMSE-C-046	Subject field	Table 3	O	
MMSE-C-047	X-Mms-Message-Class field	Table 3	M	
MMSE-C-048	X-Mms-Message-Size field	Table 3	M	

© 2004 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Na [0085] teaches that the MMS command and message of FIG 2 is used to send the segment identifiers "the header 102 including the number of total segments of the media data (**X-Total-Seq**), the sequence number of the segment to be uploaded (**X-Seq-No**, *e.g.,* 1), and a message ID, and the message body 104 as shown in **FIG. 2** (S20)." The defect in this analysis is that the MMS 1.0 specification information included in the FIG. 2 are NOT understood by non MMS

servers like the web (website) servers of Takahashi. *See* Gajendragad [Exh. 2001] at ¶ 178.

Still further, FIG. 2 and all of the command messages contemplated in Na are subject to MMS 1.0 Specification requirements (*see* Gajendragad [Exh. 2001] at ¶ 179):



It is well understood in the art that **MMS** is a system application by which a Wireless Application Protocol (WAP) client is able to provide a messaging operation with a variety of media types. The WAP Forum defines a set of protocols in transport, security, transaction, session, and application layers. Still further, the

Open Mobile Alliance (OMA) defines the WAP architecture, in: "Wireless Application Protocol Architecture Specification" (WAPARCH). *See* Exh. 2007 and Exh.2008; *see also* Gajendragad [Exh. 2001] at ¶ 180.

Petitioner acknowledges that Takahashi does NOT disclose "Segment Identifiers," and thus seeks to expand upon the teachings of Takahashi by incorporating Na and its inclusion of segment identifiers**.** The Petition incorrectly argues that: "POSITA also would have been motivated and found it obvious and straightforward to implement a mobile phone similar to the one disclosed by Takahashi, <u>based on Na, to include a division/segment identifier</u> with each divided media data to be uploaded." *See* Gajendragad [Exh. 2001] at ¶ 181.

The problem here is that the proposed combination of Takahashi with Na **would not work** because every message in Na for both Header Information and Message Body contains MMS 1.0 specification information and MMS 1.0 specification data. *See* Na (Exh. 1006) at [0062], [0065], [0116], [0118], [0121], and [0123]; *see also* Gajendragad [Exh. 2001] at ¶ 182.

Similarly, Petitioner acknowledges that Takahashi terminates the connection and does NOT disclose synchronization and resuming the data upload, yet Petitioner asserts to expand Takahashi with Na for synchronization and restart. The Petition argues: "In this way, in the combination, the mobile phone need not restart upload of the whole media data." The problem here is that in Na,

commands and messages that contain details about the Segment Identifiers, synchronization and restart, all such MMS messages contain MMS 1.0 specification information. Of course, if Takahashi is the base teaching, then the messages of Na must be able to be understood by the Takahashi web (website) server. Further, if the server of Na is on the other side, then the Takahashi initial message with Filename and User ID embedded inside the filename must be understood by the Na MMS Servers. *See* Gajendragad [Exh. 2001] at ¶¶ 183-184.

In BOTH those scenarios the working solution is NOT possible as both messages are incompatible with each other and both messages (Takahashi/Na) cannot be understood by each other's servers (namely: MMS messages by web server and Web messages by MMS Server). *See* Gajendragad [Exh. 2001] at ¶ 185.

Moreover, and by way of example, as taught in **Na, there is no available field for "Filename,"** in any of the Na or MMS messages, and Petitioner cannot explain and fails to explain how such would be incorporated into the hypothetical header. Further, a POSITA would readily appreciate the fact that the MMS Application Server and MMS Open Mobile Alliance *do not* send filenames of the MMS media. *See generally* Exh. 2007 and 2008. *See* Gajendragad [Exh. 2001] at ¶ 186.

By way of confirmation of the fact that neither "filename" or "file name" is sent in any MMS HTTP header or message, one need only search the OMA

documents (Exhs. 2007 and 2008) to confirm no such usage. Likewise, a search of Na confirms that Na also does not send "filename" or "file name." *See* Gajendragad [Exh. 2001] at ¶ 187.

The phone MMS server application architecture and requirements are completely different from a regular web-based application architecture. Each segment message body in Na would have extra MMS 1.0 Specification data and header information, (as contemplated in Na at [0051]). *See* Gajendragad [Exh. 2001] at ¶ 188.

Petitioner expressly considers Takahashi as a base, and its website server receives all the commands and messages. Petitioner states: "A POSITA would have recognized and found obvious that Takahashi's server would have hosted a website." If one was to reverse the situation, the solution will still not work, because the MMS Application Server will not understand the regular non-MMS messages sent by Takahashi. This is so because MMS Open Mobile Alliance messages and commands do **not** send filenames of the MMS media to the MMS Server. *See* Gajendragad [Exh. 2001] at ¶ 189.

Another major point rendering Takahashi and Na *incompatible and non-workable* is the fact that Na (MMS) messages are "Binary encoded," and the regular "HTTP" web messages in Takahashi **are not** inherently binary encoded.

See Gajendragad [Exh. 2001] at ¶ 190. The MMS Binary encoding is discussed in detail in the section below.

By way of support, Exhibit 1007 page 396, HTTP uses charset, "character encoding":

Multilingual Character Encoding Primer

The previous section described how the HTTP Accept-Charset header and the Content-Type charset parameter carry character-encoding information from the client and server. HTTP programmers who do a lot of work with international applications and content need to have a deeper understanding of multilingual character systems to understand technical specifications and properly implement software.

It isn't easy to learn multilingual character systems—the terminology is complex and inconsistent, you often have to pay to read the standards documents, and you may be unfamiliar with the other languages with which you're working. This section is an overview of character systems and standards. If you are already comfortable with character encodings, or are not interested in this detail, feel free to jump ahead to "Language Tags and HTTP."

Further, Exhibit 1007, page 401, HTTP uses charset, "character encoding":

Character Encoding Schemes

Character encoding schemes pack character code numbers into content bits and unpack them back into character codes at the other end (Figure 16-7). There are three broad classes of character encoding schemes:

Fixed width
Fixed-width encodings represent each coded character with a fixed number of bits. They are fast to process but can waste space.

Variable width (nonmodal)
Variable-width encodings use different numbers of bits for different character code numbers. They can reduce the number of bits required for common characters, and they retain compatibility with legacy 8-bit character sets while allowing the use of multiple bytes for international characters.

Variable width (modal)
Modal encodings use special "escape" patterns to shift between different modes. For example, a modal encoding can be used to switch between multiple, overlapping character sets in the middle of text. Modal encodings are complicated to process, but they can efficiently support complicated writing systems.

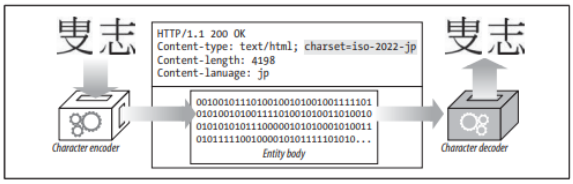


Figure 16-7. Character encoding scheme encodes character codes into bits and back again

Let's look at a few common encoding schemes.

Neither Petitioner nor Dr. Traynor discuss the incompatibility of Na extra MMS 1.0 Specification data and header information and how it will be understood

and rendered by Takahashi web (website) server and vice versa. Absent from the Petition is any explanation of what a POSITA would do with MMS Specification Information and Data, or how the website/webservice would ultimately react to the MMS specification Data in all the messages. *See* Gajendragad [Exh. 2001] at ¶193.

Further, both Petitioner and Dr. Traynor fail to discuss the incompatibility of Na "Binary encoded" MMS message and the regular "HTTP" web messages of Takahashi. *See* Gajendragad [Exh. 2001] at ¶ 194.

In sum, it would not be obvious or straightforward to implement Takahashi's mobile phone to receive from the server a synchronization reply, as taught by Na. *See* Gajendragad [Exh. 2001] at ¶ 195. In any event, such combination would be inoperable, given the fact that MMS Server Messages would not have been understood by the Web Server, and vice versa. Nevertheless, Petitioner broadly concludes that the proposed combination of Takahashi and Na would be operable, yet Petitioner fails to explain precisely how a POSITA would combine the two disparate message (MMS versus Web) formats when their respective header and message bodies are incompatible. *See* Gajendragad [Exh. 2001] at ¶ 195.

Moreover, as of the date of invention, and even today, it is extremely inefficient, from a licensing cost perspective, to use an MMS application server. Nevertheless, all of the messages contemplated in Na are subject to MMS 1.0 Specification requirements for message command, message header, and message

body. However, a POSITA as of 2007 would know not to use MMS-based HTTP messages for an Open Web application because the combination of the two (MMS and Open Web) *would not work and are incompatible*. *See* Gajendragad [Exh. 2001] at ¶ 196.

Second, the Na Segment identifiers are "Binary Encoded" but the claim construction requires a "segment identifier" to be a "a text string that identifies uniquely a data segment." *See* Gajendragad [Exh. 2001] at ¶ 197.

Petitioner relies of Na "X-Total-Seq" and "X-Seq-No" as "segment identifier", in the FIG 2 upload message and FIG 6 synchronization message. However, MMS Application Server and MMS Open Mobile Alliance *make it clear that they use "Binary encoding" for their MMS Header and messages. See* page 10 Exh. 2008 ("This specification defines the structure, content and encoding of the <u>protocol data units (PDUs) for the Multimedia Messaging Service (MMS)</u> Version 1.1." "MMS PDUs are transferred between an MMS Client and an MMS Proxy-Relay in <u>binary encoded</u> message format."; "[WAPWSP] provides mechanisms for binary encoding of such messages and serves as a basis for the <u>binary encoding of MMS PDU</u>s" ; "this specification defines the <u>binary encoding of MMS PDU</u>s to be transferred via WSP/HTTP. <u>Binary codes for header field names as well as header field values</u> are assigned"). *See* Gajendragad [Exh. 2001] at ¶ 198.

Further, "The basic encoding mechanisms for binary encoded MMS PDUs originates from WSP specification [WAPWSP], because this is very tight encoding intended to optimize amount of data transmitted over the air." ; "**Header fields** included in a MM **SHALL be encoded according to WSP Binary Encoding.**" *See* page 32 Exh. 2008; *see also* Gajendragad [Exh. 2001] at ¶ 199:

7. Binary Encoding of Protocol Data Units

The basic encoding mechanisms for binary encoded MMS PDUs originates from WSP specification [WAPWSP], because this is very tight encoding intended to optimize amount of data transmitted over the air.

A POSITA would understand the difference between binary encoding and text strings lies in how data is represented and processed (*see* Gajendragad [Exh. 2001] at ¶ 200):

Binary Encoding:

Representation: Data is represented in binary format (0s and 1s).

Usage: Commonly used for low-level data processing, such as in computer memory, file storage, and network communication.

Efficiency: More efficient for computers to process and store, as it directly corresponds to the machine's native language.

Example: The ASCII character 'A' is represented as 01000001 in binary.

Text String:

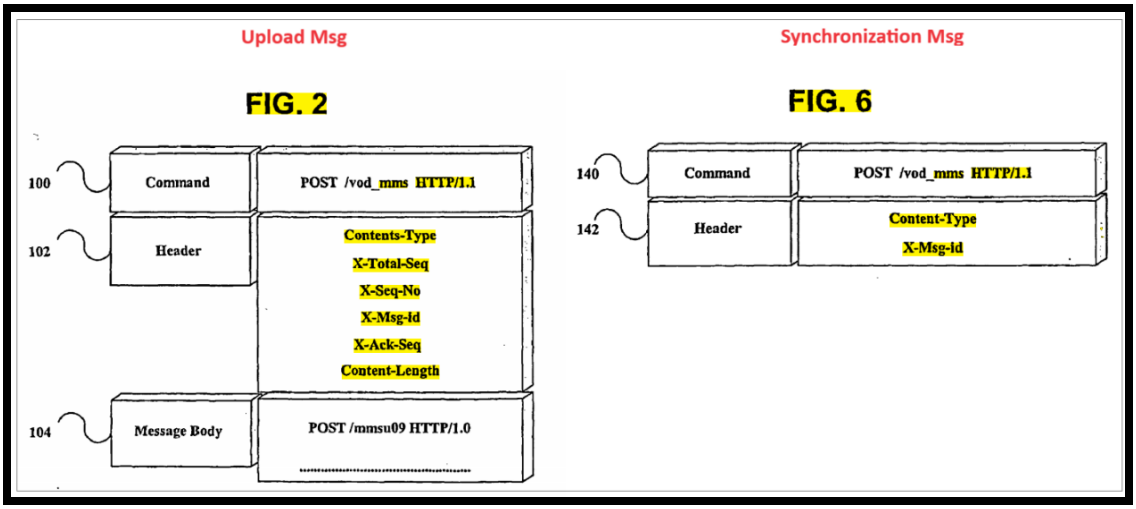
Representation: Data is represented as a sequence of characters.

Usage: Commonly used for human-readable text, such as in documents, messages, and user interfaces.

Efficiency: Easier for humans to read and understand, but requires encoding (like ASCII or UTF-8) to be processed by computers.

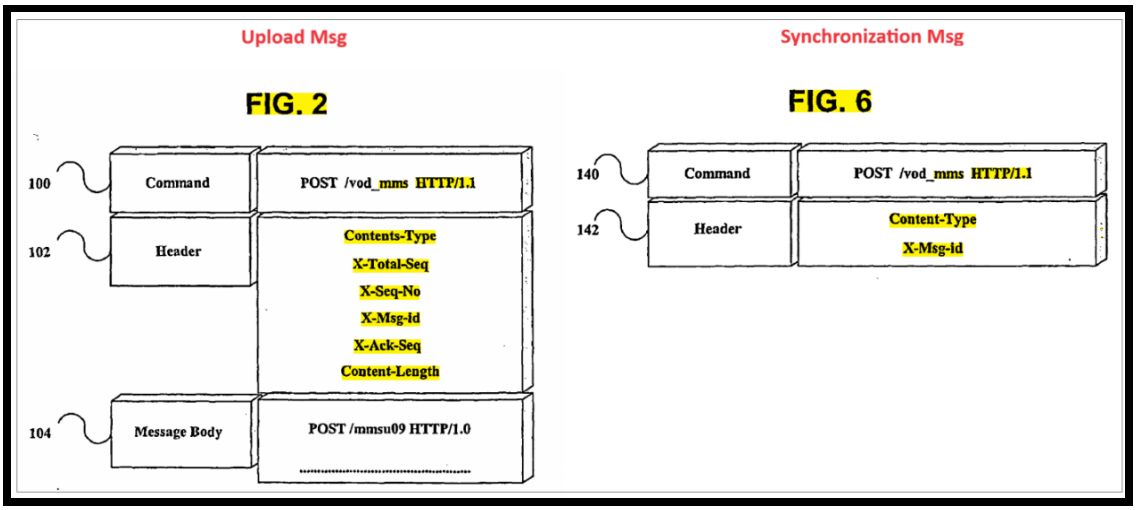
Example: The word "Hello" is represented as a sequence of characters: H, e, l, l, o.

In summary, a POSITA would appreciate the fact that binary encoding is more efficient for computer processing, while text strings are more accessible for human interaction. Both are essential in different contexts of computing and communication. Here, the **MMS header of Na** that has binary encoded header filed values for "X-Total-Seq" and "X-Seq-No" does **NOT satisfy the claim limitation.** *See* Gajendragad [Exh. 2001] at ¶ 201.



A POSITA would know that regular HTTP (Hypertext Transfer Protocol) **is not** inherently binary encoded. Rather, HTTP is a text-based protocol, meaning that the data it transmits is typically in plain text format. This includes the request and response headers, which are human-readable text strings. *See* Gajendragad [Exh. 2001] at ¶ 202.

Third, the **MMS 'upload" and "synchronization" messages disclosed in Na DO NOT contain any "User ID" or even a "File name."** *See* Na (Exh. 1006) at Figs. 2 and 6 below. *See* Gajendragad [Exh. 2001] at ¶ 203.



Na **FIG 2** (above) illustrates the structure of an initial request message for a media data "**upload**", while **FIG 6** illustrates the structure of a request message **("synchronization")** querying the sequence number of the segment from which the upload should be resumed when the connection is closed. As taught, and because there is **no** "User ID" or even a "File name" in Na, Petitioner's Ground 2 fails. *See* Gajendragad [Exh. 2001] at ¶ 204.

The Petition fails on the related basis of Petitioner's *inconsistent* purported disclosures of the recited "user identifier" throughout the Petition, and specifically "user identifier" with respect to Element 1[e], Element 1[j] and Element 1[k]. These are discussed in greater detail below. *See* Gajendragad [Exh. 2001] at ¶ 205.

With respect to Element 1[e], Petitioner relies upon Takahashi and its disclosure of a "file name" in support of its arguments relating to the claimed "applying a user identifier" limitation. *See* '336 Petition at 46 (arguing that the file name is "in the format of user ID + YYYYMMDDHHMISS (year, month, day, hour, minute, and second)") (also '757 Petition at 58 and '260 Petition at 57). Of course, Takahashi teaches that in "transmitting the divided image using HTTP, the generated file name" including the User ID "is also transmitted together." *See* '336 Petition at 47 ("There were multiple different ways to apply Takahashi's file name including the user ID ('*user identifier'*) to each divided image data ('*data segment'*)") (also '757 Petition at 59 and '260 Petition at 58); *see also* Gajendragad [Exh. 2001] at ¶ 206.

Later, and with respect to Element 1[j], Petitioner relies upon Na's "X-Msg-id" or "terminal number" and the HTTP "User Agent" for the "user identifier" in support of its arguments relating to the claimed "for said user identifier" limitation. *See* '336 Petition at 58-62 (also '757 Petition at 72 and '260 Petition at 69). Petitioner cites to the "*terminal number*" of Na as a purported disclosure of the

"user identifier" as recited in the Challenged Claims. *See id.* ("(X-Msg-id, *e.g.,* a *terminal number* and upload time information in the form of MIN_MMDDH HMMSS)"). Petitioner further contends that the "request message" disclosed in Na if "for said user identifier" as follows (*see id.*):

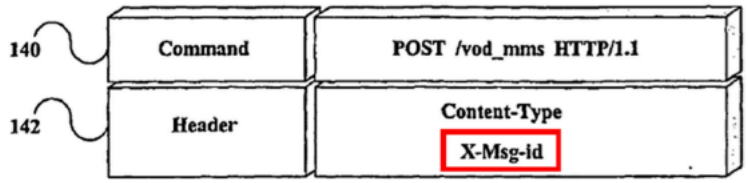
1022, 3:25-27. A POSITA would have recognized or found obvious that the request message would be "***for said user identifier***" because the request message is used to query the data previously uploaded that are associated with the "terminal

59

Attorney Docket No. 50048-0057IP1
IPR of U.S. Patent No. 8,756,336

number" or user ID ("***said user identifier***") to determine "the sequence number of the segment from which the upload should be resumed" for the same user identifier. TTI-1006, ¶[0094]; TTI-1003, ¶139. As such, Takahashi-Na teaches sending, a synchronization request to the web service, for said user identifier, after a network connection to the web service is restored. *Id.*

FIG. 6



TTI-1006, FIG. 6.

Petitioner's argument fails, given the fact that the phone MMS video request messages in Na are based on X-Msg-ID and *not* on any unique user identifier of the website/web service as required by the claim. Still further, the phone MMS-

based "terminal number" inside the X-Msg-ID *is not a unique user identifier of the web service*. *See* Gajendragad [Exh. 2001] at ¶¶ 207-208.

Moreover, Petitioner fails to recognize that **Na header "X-Msg-ID"** are "Binary Encoded" but fundamental claim construction requires a "user identifier" to be a "a text string that identifies uniquely a user of a web service." *See* Gajendragad [Exh. 2001] at ¶ 209.

Petitioner relies on Na "X-Msg-ID" as "user identifier", in the FIG 6 synchronization message. BUT MMS Application Server and MMS Open Mobile Alliance *make it clear that they use "Binary encoding" for their MMS Header and messages. See* page 10 Exh. 2008 ("This specification defines the structure, content and encoding of the protocol data units (PDUs) for the Multimedia Messaging Service (MMS) Version 1.1." "MMS PDUs are transferred between an MMS Client and an MMS Proxy-Relay in binary encoded message format."; "[WAPWSP] provides mechanisms for binary encoding of such messages and serves as a basis for the binary encoding of MMS PDUs" ; "this specification defines the binary encoding of MMS PDUs to be transferred via WSP/HTTP. Binary codes for header field names as well as header field values are assigned"). *See* Gajendragad [Exh. 2001] at ¶ 210.

Still further: "The basic encoding mechanisms for binary encoded MMS PDUs originates from WSP specification [WAPWSP], because this is very tight

encoding intended to optimize amount of data transmitted over the air." ; "**Header fields** included in a MM **SHALL be encoded according to WSP Binary Encoding.**" *See* page 32 Exh. 2008; *see also* Gajendragad [Exh. 2001] at ¶ 211.

Regarding the HTTP "User Agent" reliance for unique user identifier, as of the date of invention and even today, a POSITA would have readily known and understood that the "User Agents" as defined by the HTTP protocol were/are not unique per user. Rather, due to the governing HTTP syntax, literally thousands of individuals at any given point in time could have the exact same HTTP User Agent. More specifically, as defined by the HTTP protocol, the "User Agent" simply identified the browser name and/or version and the PC or Mac version being utilized by the user (see image below). Such "User Agent" identifiers were not unique in any sense of the word, and cannot be understood as satisfying the "User Identifier" as recited in the Asserted Claims. *See* Gajendragad [Exh. 2001] at ¶ 212.

Syntax

HTTP

```
User-Agent: <product> / <product-version> <comment>
```

Common format for web browsers:

HTTP

```
User-Agent: Mozilla/5.0 (<system-information>) <platform> (<platform-details>)
<extensions>
```

Examples

```
Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:47.0) Gecko/20100101 Firefox/47.0
Mozilla/5.0 (Macintosh; Intel Mac OS X x.y; rv:42.0) Gecko/20100101 Firefox/42.0
```

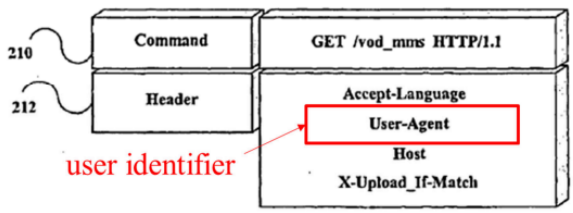
```
Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/51.0.2704.103 Safari/537.36
```

See https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/User-Agent.

Finally, and with respect to **Element 1[k],** "receiving, a synchronization reply back from the web service at the application layer level, ……for **said user identifier,**" Petitioner identifies the HTTP "User Agent" in support of its arguments relating to the claimed "for said user identifier" limitation. *See* '336 Petition at 66 ("the range of data that has been uploaded already that are associated with the 'user agent information' ('said user identifier')") (also '757 Petition at 78 and '260 Petition at 75); *see also* Gajendragad [Exh. 2001] at ¶ 213.

A POSITA would have recognized or found obvious that the response message would be "***for said user identifier,***" for example, because (1) the response message is directly in response to the request message that is "***for said user identifier***" as discussed in element 1[j]; and (2) the response message itself identifies "the range of data that has been uploaded already" that are associated with the "user agent information" ("***said user identifier***") so as to inform the mobile communication terminal "where to resume the upload" the data for the user identifier. TTI-1006, ¶¶[0153]-[0155],[0119]; *see supra* 1[e], [1j]; TTI-1003, ¶¶147-148.

66



TTI-1006, FIG. 16.

Here, Petitioner *does not rely* upon the "terminal number" or "X-Msg-id" for user identifier or even Takahashi's user-ID inside the file name. As such, **Element 1[k]** cannot be satisfied by Petitioner if the Board properly considers the fact that the *HTTP User Agent* is not a unique "user identifier" as required by the Challenged Claims. Again, at any given point in time, thousands of people on a website can have the exact same (non-unique) HTTP User Agent. Again, for multiple reasons, Petitioner's arguments fail. *See* Gajendragad [Exh. 2001] at ¶ 214.

Moreover, even if petitioner was to point to some part of the information in MMS Header like "X-Msg-id" in FIG 2 and FIG 6, Petitioner would still not be able to satisfy the claim limitation, because the claim construction would require a user ID to be at least a "Text String" and all the MMS header and header values are "binary encoded" as per the MMS specifications. *See* page 10 Exh. 2008; *see also* Gajendragad [Exh. 2001] at ¶ 215.

Finally, the Petition fails on the related basis of Petitioner's inconsistent positions relating to the recited "user identifier" in Element **1[e]** (HTTP Web Text Encoding), Element **1[j]** (MMS Binary Encoding) and Element **1[k]** (MMS Binary Encoding)**.** *See* Gajendragad [Exh. 2001] at ¶ 216.

A POSITA would understand that HTTP based identifiers must be *text* based because HTTP is a text-based protocol, meaning that the data that is transmits is typically in plain text format. This includes the request and response headers, which are human-readable text strings. *See* Gajendragad [Exh. 2001] at ¶ 217.

VI. CONCLUSION

For the reasons set forth above, Petitioner has failed to meet its burden of establishing the unpatentability of the Challenged Claims. Accordingly, Patent Owner respectfully submits requests that the relief sought in the Petition be denied, and that all Challenged Claims be upheld as patentable and valid.

//

Date: January 3, 2025

Respectfully submitted,

By: *M. Scott Fuller*
M. Scott Fuller (Lead Counsel)
Reg. No. 54,716
René A. Vazquez (Backup Counsel)
Reg. No. 38,647

Counsel for Patent Owner

CERTIFICATION OF WORD COUNT

The undersigned hereby certifies that the foregoing complies with the type-volume limitations of 37 C.F.R. § 42.24 and contains 13,519 words, excluding those portions exempted under 37 C.F.R. § 42.24(a)(1).

Dated: January 3, 2025

By: *M. Scott Fuller*
M. Scott Fuller (Lead Counsel)
Reg. No. 54,716
René A. Vazquez (Backup Counsel)
Reg. No. 38,647

Counsel for Patent Owner

CERTIFICATION OF SERVICE

Pursuant to 37 C.F.R. §§ 42.6(e) and 42.105(b), the undersigned certifies that on January 3, 2025, a copy of the foregoing document was served by electronic mail on the following:

Lead Counsel

W. Karl Renner, Reg. No. 41,265
Fish & Richardson P.C.
60 South Sixth Street
Minneapolis, MN 55402
Tel: 202-783-5070
Fax: 877-769-7945
Email: IPR50048-0057IP1@fr.com

Back-Up Counsel

Kim H. Leung, Reg. No. 64,399
Baile Xie, Reg. No. 78,380
Fish & Richardson P.C.
60 South Sixth Street
Minneapolis, MN 55402
Tel: 202-783-5070
Fax: 877-769-7945
IPR50048-0057IP1@fr.com

Dated: January 3, 2025

By: *M. Scott Fuller*
M. Scott Fuller (Lead Counsel)
Reg. No. 54,716
René A. Vazquez (Backup Counsel)
Reg. No. 38,647

Counsel for Patent Owner