

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

TIKTOK, INC.,

Petitioner,

v.

CELLSPIN SOFT, INC.,

Patent Owner

IPR2024-00770

U.S. Patent No. 8,904,030 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.....	2
III.	CLAIM CONSTRUCTION	5
IV.	GENERAL RESPONSE	20
A.	Discussion of Prior Art Bluetooth	21
V.	SPECIFIC REBUTTAL TO ALLEGED OBVIOUSNESS COMBINATIONS	31
A.	Ground 1A Fails (Hiroishi in Combination with Takahashi and Bluetooth V2.1)	31
B.	Ground 1B Fails (Hiroishi in Combination with Kahn and Bluetooth 2.1)	57
C.	Ground 2A Fails (Singh 129 in View of Bluetooth)	69
D.	Ground 2B Fails (Singh 129 in View of Bluetooth and Takahashi)	69
VI.	CONCLUSION	81

TABLE OF AUTHORITIES

<i>01 Communique Lab., Inc. v. Citrix Sys.</i> , No. 2017-1869, 2018 U.S. App. LEXIS 10614 (Fed. Cir. Apr. 26, 2018)	7
<i>Amazon.com, Inc. v. Barnesandnoble.com, Inc.</i> , 239 F.3d 1343 (Fed. Cir. 2001)	7
<i>Amgen Inc. v. Hoechst Marion Roussel, Inc.</i> , 314 F.3d 1313 (Fed. Cir. 2003)	7
<i>Belden Inc. v. Berk-Tek LLC</i> , 805 F.3d 1064 (Fed. Cir. 2015)	4
<i>Current Lighting Solns v. Jiaxing Super Lighting</i> , 2023 WL 5319763 (PTAB August 16, 2023).....	5
<i>Graham v. John Deere Co.</i> , 383 U.S. 1 (1966).....	3
<i>In re Kahn</i> , 441 F.3d 977 (Fed. Cir. 2006)	5
<i>In re Stepan Co.</i> , 868 F.3d 1342 (Fed. Cir. 2017)	2
<i>Intercontinental Great Brands LLC v. Kellogg</i> , 869 F.3d 1336 (Fed. Cir. 2017)	2
<i>InTouch Techs., Inc. v. VGO Commc'ns, Inc.</i> , 751 F.3d 1327 (Fed. Cir. 2014).....	3
<i>Merck Sharp & Dohme B.V. v. Warner Chilcott Co., LLC</i> , 711 F. App'x 633 (Fed. Cir. 2017).....	4
<i>Metalcraft of Mayville, Inc. v. Toro Co.</i> , 848 F.3d 1358 (Fed. Cir. 2017)	3

<i>Orexo AB v. Actavis Elizabeth LLC</i> , 903 F.3d 1265 (Fed. Cir. 2018)	4
<i>Otsuka Pharm. Co. v. Sandoz, Inc.</i> , 678 F.3d 1280 (Fed. Cir. 2012)	4
<i>Hulu, LLC v Soundview Innovations LLC</i> , IPR2018-01039, Paper 34, 22 (August 5, 2019).....	4
<i>KSR Int’l Co. v. Teleflex Inc.</i> , 550 U.S. 398 (2007).....	3, 5
<i>PAR Pharm., Inc. v. TWI Pharms., Inc.</i> , 773 F.3d 1186 (Fed. Cir. 2014).....	2
<i>Personal Web Techs., LLC v. Apple, Inc.</i> , 848 F.3d 987 (Fed. Cir. 2017)	5
<i>Phillips v. AWH Corp.</i> , 415 F.3d 1303 (Fed. Cir. 2005)	10
<i>Polaris Indus., Inc. v. Arctic Cat, Inc.</i> , 882 F.3d 1056 (Fed. Cir. 2018)	5
<i>TQ Delta v. Cisco Sys., Inc.</i> , 942 F.3d 1352 (Fed. Cir. 2019)	4
<i>Trivascular, Inc. v. Samuels</i> , 812 F.3d 1056 (Fed. Cir. 2016)	3
<i>Unigene Labs., Inc. v. Apotex, Inc.</i> , 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, 2, 4-7, and 9-18 of United States Patent No. 8,904,030 B2 (“the ’030 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) pairing, (ii) paired connection maintained on a continuous basis, (iii) cryptographic authentication, (iv) signal to notify, (v) event notifications, (vi) creating an object, (vii) sending and object, (viii) listening for signal to notify in an object, (ix) polling for new data, and (x) using HTTP to send data along with user identifier. 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).

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); *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). “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)). 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 key claim construction terms that parties have agreed to (Partial Agreed Construction) and there are some terms that are still at issue in the District Court for the ‘766, ‘121, ‘381, and ‘030 Patents (*see* Joint Claim Construction and Prehearing Statement, Exhibit 2005):

Terms	[Partial] Agreed Constructions
“Bluetooth enabled data capture device”	“separate and apart from the mobile device and has the ability to send and/or receive Bluetooth wireless messages”
“short-range wireless enabled data capture device”	“separate and apart from the short-range wireless enabled cellular phone and has the ability to send and receive short-range wireless messages”
“paired”	“in the condition of having an exchanged link key (either before connection establishment was requested or during connecting phase)”
“after the paired secure Bluetooth connection is established” / “after establishing a paired Bluetooth connection” “after establishing the short-range paired wireless connection” / “after the short-range paired wireless connection is established”	“after the paired connection is established and maintained on a continuous basis between the Bluetooth enabled data capture device and the Bluetooth enabled mobile device” / “after the paired connection is established and maintained on a continuous basis between the Bluetooth enabled data capture device and the Bluetooth enabled cellular phone” “after the paired connection is established and maintained on a continuous basis between the short-range wireless enabled data capture device and the short-range wireless enabled cellular phone”
“short-range paired wireless connection”	“in the condition of having an exchanged link key (either before connection establishment was requested or during connecting phase) between the short-range wireless enabled data capture device and the short-range wireless enabled cellular phone”
“cryptographically authenticating”	“verified as legitimate by use of encryption and decryption involving an algorithm”
“data signal	“ an indication of the presence of newly acquired

notification(s)”	data”
“poll” / “polling”	“checking status [of] on a predetermined basis,” where “predetermined” refers to programming that sets the frequency of the status checks (e.g., at set intervals, dynamic intervals, randomly, etc.)
“user authentication credential”	“data used to verify the identity of the user”

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 term “after the paired secure Bluetooth connection is established,” as well as similar terms such as “after establishing the short range paired wireless connection” are NOT disputed in the Texas District Court. These terms have been previously construed (by the Northern District of California) to mean: “after the paired connection is established and maintained on a continuous basis.” *See* Exhibit 2022 at This proposed construction is fully supported by the opinions of Pavan M. Gajendragad. *See* Gajendragad Declaration [Exhibit 2001] at ¶ 36.

Again, this construction has now been stipulated by the parties in the District Court.

In the Northern District of California, the Court found as follows:

“[P]rosecution disclaimer limits the term to maintaining continuous paired connections while new data is acquired in this case. During prosecution of the ‘794 Patent, Cellspin added the limitation in question in order to overcome prior art. The examiner rejected the unamended claims over a ‘Kennedy’ reference, which disclosed uploading new data in one of three ways: (1) automatically, upon sensing that the device is full, (2) manually, when the user decides to perform the transfer, or (3) in real-time, as soon as data is acquired. Cellspin argued that Kennedy did not disclose whether the connection is opened before or after the data acquisition and whether it is closed or kept open after data transfer. It later clarified that Kennedy took the conventional approach of disconnecting the paired connection after data transfer to save power, while the patent ‘took the opposing view,’ which is ‘to . . . maintain the BT paired connection on a continuous basis for the application’ and ‘not to disconnect the BT paired connection after the transfer of data.’ The reason for this approach was a belief ‘that having a constant connection would be the key to the overall improvement in the application architecture and responsiveness of the application.’ The Court finds that the above statements constitute clear and unambiguous disclaimer of paired connections that are not maintained on a continuous basis. *Aylus Networks, Inc. v. Apple Inc.*, 856 F.3d 1353, 1359 (Fed. Cir. 2017). Cellspin had ‘unequivocally and unambiguously’ disavowed systems that fail to maintain a continuous paired connection, characterizing this feature as ‘key’ to the improvements of the invention, and cannot now interpret the claims to cover such systems. Moreover, **because the ‘794 Patent is related** to the other asserted patents, **the disclaimer extends to all** asserted patents having this limitation” (internal citations omitted).

In a footnote, the Court also noted: “Defendants point out, Cellspin made similar arguments related to a ‘permanent’ paired wireless connection during

prosecution of the ‘847 Patent, which further confirms the common understanding of this term in **all** patents.” Thus, the Court found that the disclaimer extended to ALL related patents because of *Microsoft Corp. v. MultiTech Sys., Inc.*, 357 F.3d 1340, 1349-50 (Fed. Cir. 2004).

Given the agreement between the parties in the ongoing District Court case and the earlier construction, the Board should construe the terms consistently as “after the paired connection is established and **maintained on a continuous basis.**”

In addition, the claim term “along with” is disputed at the District Court (as it concerns the ’030 Patent), and Cellspin is asking both the District Court and this Board 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 ¶ 37.

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. *See* Gajendragad Declaration [Exhibit 2001] at ¶ 38.

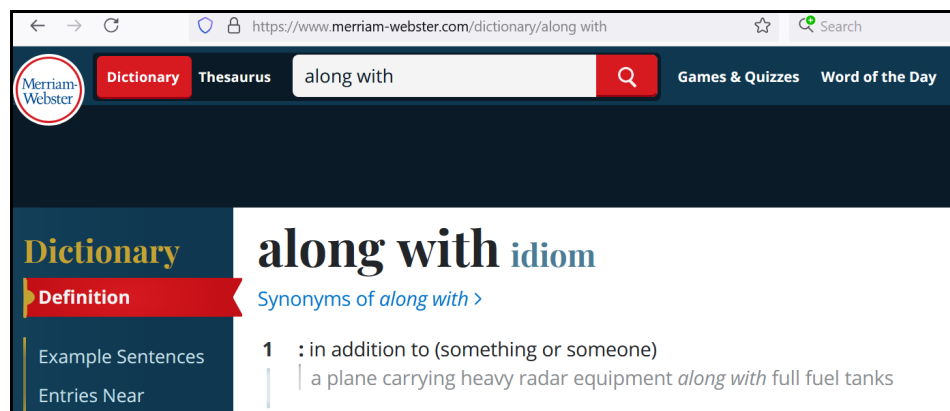
Intrinsic Evidence from the patent specification

[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. *See* Gajendragad Declaration [Exhibit 2001] at ¶¶ 39-40.

Still further, dictionary evidence (*see* <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 Declaration [Exhibit 2001] at ¶ 41):



Moreover, it is incorrect to suggest that the plain and ordinary meaning of “along with” could include “inside.” *See* Gajendragad Declaration [Exhibit 2001] at ¶ 42. 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 Declaration [Exhibit 2001] at ¶ 43.

In sum, a POSITA would have understood the term “along with” to mean simply “in addition to,” and not including the concept of “inside.” *See* Gajendragad Declaration [Exhibit 2001] at ¶ 44.

Still further, the term “object” is also disputed by the parties in the District Court case (as relates to the ’030 and ’766 Patents), and Cellspin proposes the

following: “An object can be a variable, a data structure, a function, or a method. Object contains value and is referenced by an identifier.” This proposed construction is fully supported by the opinions of Pavan M. Gajendragad. *See* Gajendragad Declaration [Exhibit 2001] at ¶ 45.

With respect to the “intrinsic record,” it is clear to me that an “object” is different from a “file.” The specification uses multiple terms and does not conflate them. I note that the specification uses the words “object,” “file,” and “data” many times, and further differentiates between “data” and “file” because it describes a “*a data **and** file monitoring and detection module 203b.*” *See* Gajendragad Declaration [Exhibit 2001] at ¶ 46.

Further, the specification explains: “The digital data capture device 201 then automatically **transfers** 106 the captured **data**, the multimedia content, and the associated files to the client application 203 on the mobile device 202 **using one or a combination of file transfer protocols**. The file transfer protocols may, for example, be one or a combination of BT profile protocols **such as the object exchange (OBEX) protocol**, the generic **object exchange profile (GOEP)** protocol, etc. The file transfer protocols may, for example, also be the media transfer protocol (MTP), the picture transfer protocol **(PTP)**, and the **PictBridge** protocol implemented using a USB.” The specification is very clear that “Data” can be transferred by using Object based methods like (OBEX and GOEP) or **non-**

object-based methods like MTP and PTP and PictBridge. See Gajendragad

Declaration [Exhibit 2001] at ¶ 47.

What the Petition fails to recognize is that, while it is technically possible to send/transfer a “File” using OBEX protocol, *a “File” has to be converted into a “Generic File Object”* for it to be transferred using OBEX. This is illustrated below (Exhibit 2009 (IrDA - OBEX Version 1.3) at 73), wherein “File” is sent as a Generic File Object, but the “File” is not an OBJECT. Rather, it must first be converted into an Object (although the file and file object are different). See Gajendragad Declaration [Exhibit 2001] at ¶ 48.

9.2 Generic File Object

9.2.1 Introduction

The purpose of this chapter is to outline a method for the exchange of a basic file object [blob] from one device to another. While the OBEX protocol provides many useful headers and capabilities for exchanging objects. This section is designed to provide the reader with an overview of the basic structure used when exchanging generic file data. The file object is assumed to be sent by a generic client to the receiving device's inbox. In a manner consistent to the way QuickBeam and IrXfer applications work. Based on these assumptions, the file exchange should follow the guidelines shown here for headers used and response codes expected.

9.2.2 Commonly Used Headers

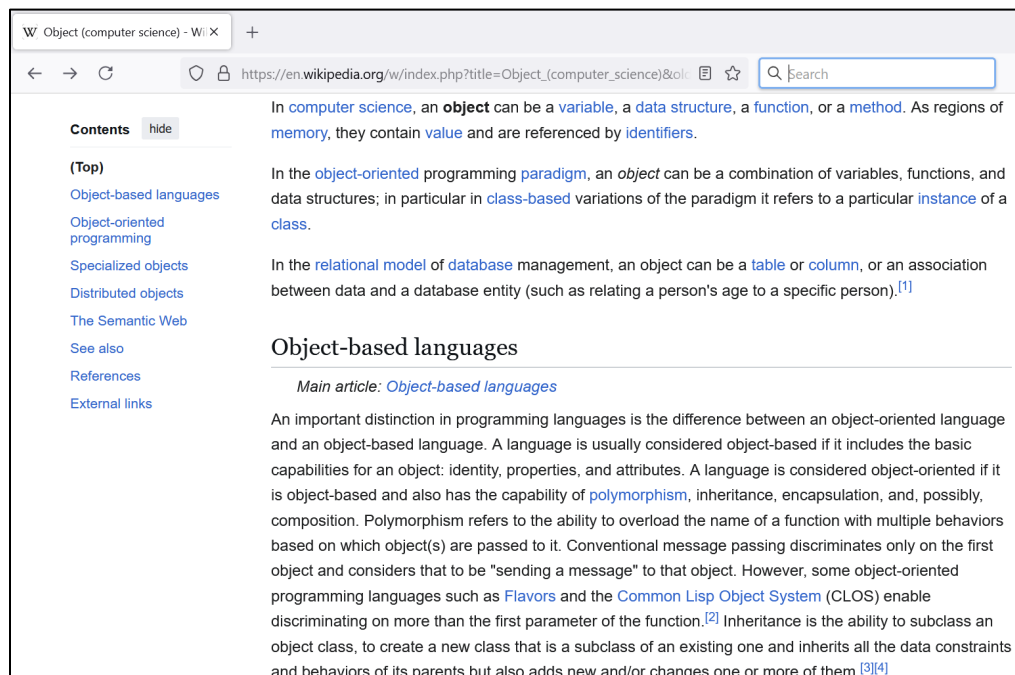
- **Name:** *(Required)* This header is used to convey the full name of the object to exchange. The **Name** header should not contain any path information. If it is necessary to specify a path, it should be done with the **SETPATH** command. All information in this header is interpreted as the objects name.
- **Length:** *(Recommended)* This header is used to convey the size of the file object in bytes. This can be used for verification of storage space requirements. The sum of the bytes passed in **Body** headers should be this many bytes (but is not guaranteed).
- **Body/End-of-Body:** *(Required)* Body headers are used to transfer the file data itself. They are sent repeatedly as long as file data exists to be exchanged. The **End-of-Body** header is used to indicate to the receiving application that this is the last piece of the file data. Applications frequently close the file and consider the transfer complete upon reception of this header. The data contained in the **Body/End-of-Body** headers should be concatenated to form the file.
- **Time:** *(Recommended)* This header is used to exchange the time that the file was last modified. Current file transfer applications use both the simple 4-byte **Time** header and the ISO format. It is recommended that all applications use the more robust ISO **Time** header. For backward compatibility it should be noted that the “IrXfer” application provided by Microsoft only accepts the 4-byte **Time** header. Attempts to send the ISO header will result in the rejection of the operation.
- **Type:** *(Optional)* The MIME type of the file object. This is not often used but can be helpful in some cases to identify the application to associate with the file object.

9.2.3 Response Codes Commonly Used in File Exchange

The following is a list of the known OBEX Response codes that are used during File Object Exchange.

Again, and as a POSITA would readily appreciate, a “file” has to be converted to be an “Object” to be sent using OBEX. Similarly, “Data” (that is NOT itself a “File”) can also be converted into an Object and then sent/transferred using OBEX protocol. OBEX is an **object exchange** protocol, **NOT** a file exchange or data exchange protocol, but both file and data can be sent using it by creating an Object first. *See* Gajendragad Declaration [Exhibit 2001] at ¶ 49.

It is further noted that Wikipedia confirms the fact that “An **object** can be a variable, a data structure, a function, or a method. Object contains value and is referenced by an identifier.” *See* Exhibit 2021; *see also* Gajendragad Declaration [Exhibit 2001] at ¶ 50.



Still further, the Microsoft Computer Dictionary (Fifth Ed.) (pages 372-373) provides a substantially similar definition of the word “object,” which it defines as:

“a variable comprising both routines and data that is treated as a discrete entity”

Object is “that are self-contained collection of data structures.” *See* Gajendragad

Declaration [Exhibit 2001] at ¶ 51.

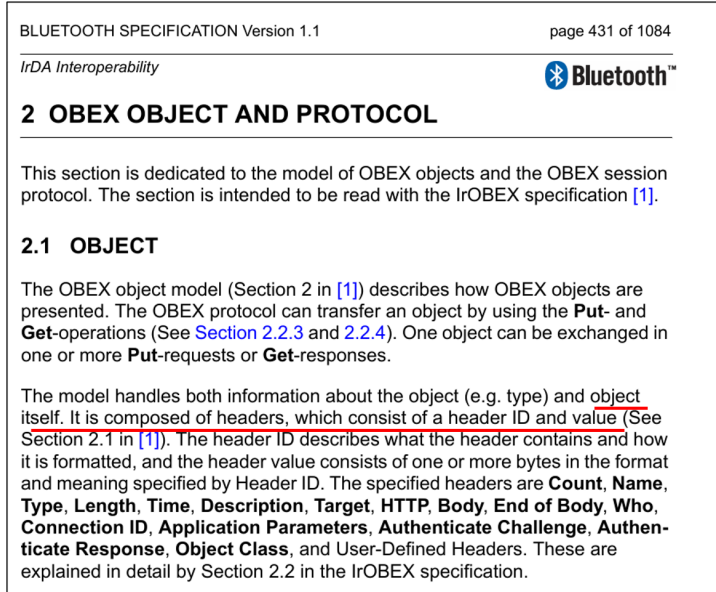
object *n.* **1.** Short for object code (machine-readable code). **2.** In object-oriented programming, a variable comprising both routines and data that is treated as a discrete entity. *See also* abstract data type, module (definition 1), object-oriented programming. **3.** In graphics, a distinct entity. For example, a bouncing ball might be an object in a graphics program. **4.** A single, runtime instance of object type that the operating system defines. Objects visible in user mode include event, file, I/O completion port, key, object directory, port, process, section, semaphore, symbolic link, thread, timer, and token objects. Many user-mode objects are implemented through the use of a corresponding kernel-mode object. Kernel-mode-only objects

object-oriented programming *n.* A programming paradigm in which a program is viewed as a collection of discrete objects that are self-contained collections of data structures and routines that interact with other objects. *Acronym:* OOP. *See also* C++, object (definition 2), Objective-C.

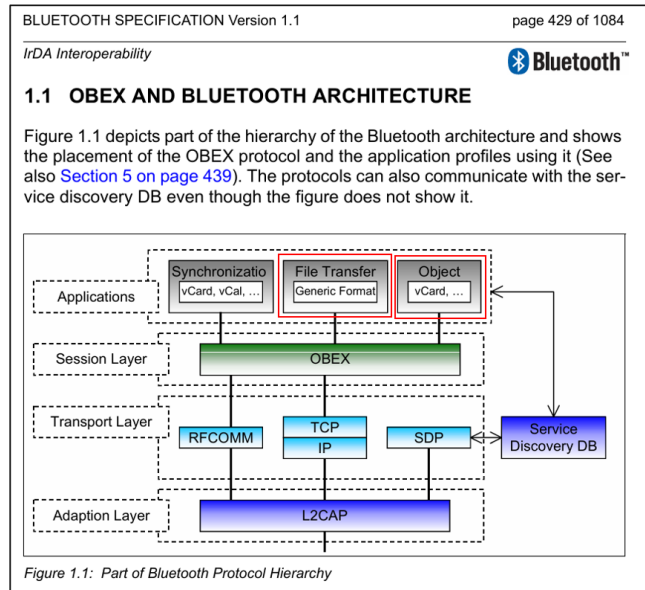
Object Pascal *n.* An object-oriented derivative of Pascal. *See also* Pascal.

None of these sources suggest that “File” and “Object” are (or can be) the same. *See* Gajendragad Declaration [Exhibit 2001] at ¶ 52.

Yet still further, the Bluetooth specification itself defines “Object” as: “is composed of Headers which consists of Header ID and Value.” *See* Gajendragad Declaration [Exhibit 2001] at ¶ 53.



A POSITA would understand and appreciate the fact that Object and File are completely different concepts in Bluetooth as well. As shown in OBEX AND BLUETOOTH ARCHITECTURE (2003 Bluetooth Specification, Version 1.1, at page 429) (Exhibit 2016), File Transfer and Object Push are **two different Applications running on top of OBEX Protocol**. See Gajendragad Declaration [Exhibit 2001] at ¶ 54.



In sum, a POSITA would understand that **Object and File are different**.

Thus, a proper construction should account for this fact. *See* Gajendragad

Declaration [Exhibit 2001] at ¶ 55.

Further, the term “user identifier” is contested by the parties in the District Court case, as it relates to the ‘336, ‘757, ‘030, and ‘260 Patents. The primary dispute here concerns a proposal from Petitioner (TikTok) 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 Cellspin agreed. A POSITA would understand that HTTP (HyperText 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 the website / web service. *See* Gajendragad Declaration [Exhibit 2001] at ¶ 67.

	Claim Term	Proposed Constructions
	“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”

IV. GENERAL RESPONSE

None of the alleged prior art references would have been combined by a POSITA in the manner described by David Lett. Moreover, the various alleged prior art references and their combinations fail to teach many of the elements of the Challenged Claims, and in certain cases combinations actually lead to “unworkable” solutions, when compared with the inventions as claimed in the Challenged Claims. A POSITA would have had NO reasonable expectation of success in implementing the combinations. *See* Gajendragad Declaration [Exhibit 2001] at ¶ 68.

Further, none of the asserted combinations would have worked and would lead to unworkable implementation. A POSITA would have had NO reasonable expectation of success in implementing the combinations. *See* Gajendragad Declaration [Exhibit 2001] at ¶ 69.

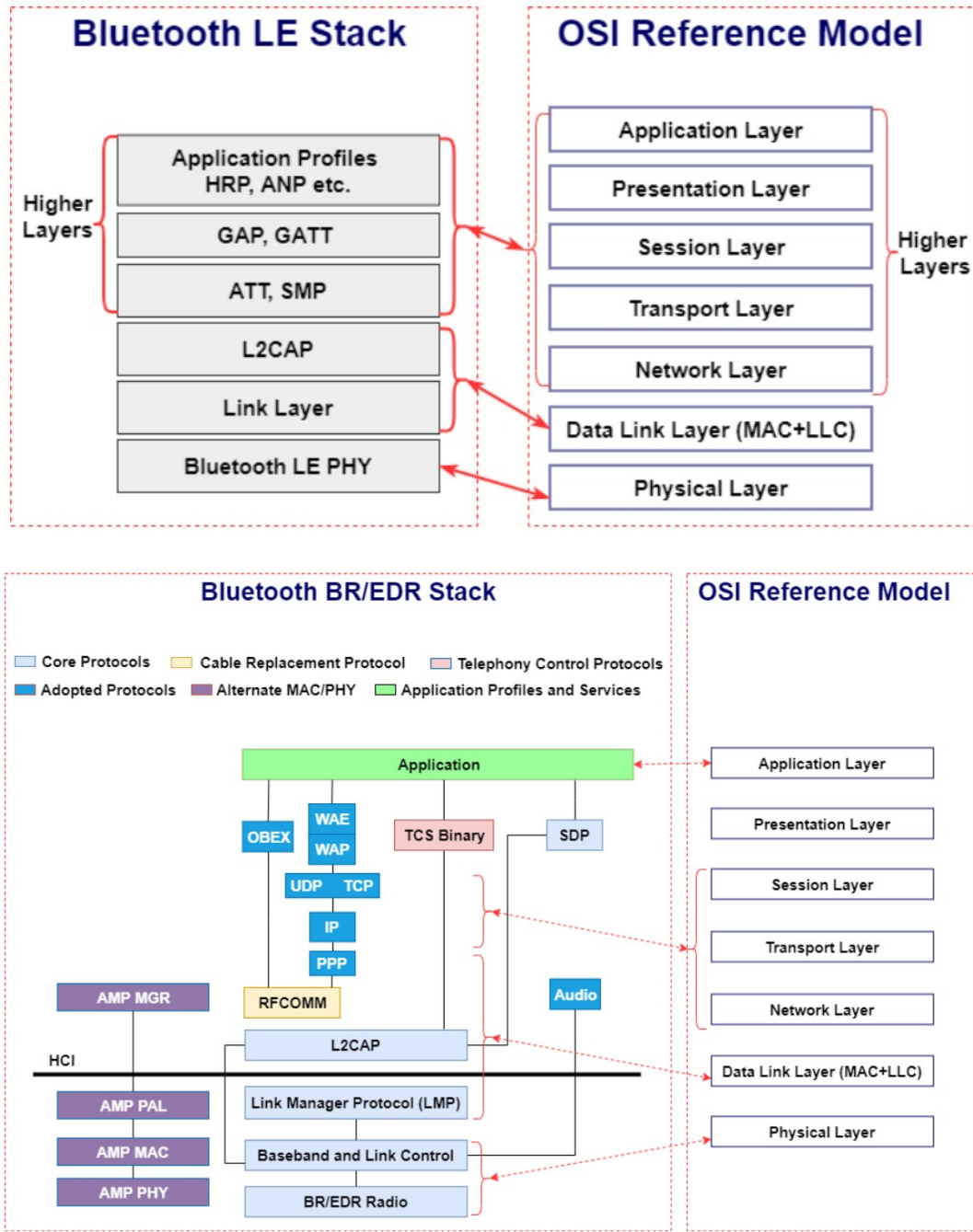
Still further, the specific references cited by Petitioner (alone or in combination) fail to teach many of the claim limitations, and a POSITA would not

have been motivated to combine them in the first place. *See* Gajendragad Declaration [Exhibit 2001] at ¶ 71. 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. A POSITA would have had NO reasonable expectation of success in implementing the combinations. *See* Gajendragad Declaration [Exhibit 2001] at ¶ 71.

A. Discussion of Prior Art Bluetooth

In order to understand the invention and the Challenged Claims and the prior art it is important to establish an industry standard and definition of certain terms in 2007.

It was known and understood that Bluetooth technology maps to the Open Systems Interconnection (OSI) model in specific ways (*see* Gajendragad Declaration [Exhibit 2001] at ¶ 76). Illustrated below are the Bluetooth protocol mapping to OSI stack Math works. *See* Gajendragad Declaration [Exhibit 2001] at ¶ 77; *see also* www.mathworks.com/help/bluetooth/ug/bluetooth-protocol-stack.html



In the preceding figure, the Bluetooth protocol stack is shown along with the OSI reference model. There is one-to-one mapping at the physical layer (PHY). The OSI data link layer (DLL) maps to the Bluetooth LE logical link control and adaptation protocol (L2CAP) and link layer (LL). In the Bluetooth stack, the

higher layers provide application layer services, device roles and modes, connection management, and security protocol. The functionality of the Bluetooth protocol stack is divided between three main layers: the Controller, the Host, and Application Profiles and Services. *See* Gajendragad Declaration [Exhibit 2001] at ¶ 78.

As would have been further understood by a POSITA, “Pairing” in Bluetooth is different from “Authentication” and both are different from “Encryption.” As shown in the figure below [Exhibit TTI 1019, p. 19], there are 15 different optional activities after a Bluetooth connection is established, with Pairing, Authentication, and Encryption each being different optional activities. Also, as shown in Figure 3.1 in the v2.1+EDR Bluetooth Core Specification [Exhibit TTI 1019, p. 861] Pairing is different from Authentication and both of them are different from Encryption. *See* Gajendragad Declaration [Exhibit 2001] at ¶ 79.



2.1	Remote Name Request.....	685
2.2	One-time Inquiry.....	687
2.3	Periodic Inquiry	689
3	ACL Connection Establishment and Detachment.....	691
3.1	Connection Setup	692
4	Optional Activities After ACL Connection Establishment.....	699
4.1	Authentication Requested.....	699
4.2	Simple <u>Pairing</u> Message Sequence Charts.....	700
4.3	Link Supervision Timeout Changed Event	713
4.4	Set Connection <u>Encryption</u>	714
4.5	Change Connection Link Key.....	715
4.6	Change Connection Link Key with Encryption Pause and Resume.....	715
4.7	<u>Master Link Key</u>	717
4.8	Read Remote Supported Features	719
4.9	Read Remote Extended Features.....	719
4.10	Read Clock Offset.....	720
4.11	Role Switch on an <u>Encrypted Link using Encryption Pause and Resume</u>	720
4.12	<u>Refreshing Encryption Keys</u>	721
4.13	Read Remote Version Information	722
4.14	QOS Setup.....	722
4.15	Switch Role	723

All 15 are Optional

3 ACL CONNECTION ESTABLISHMENT AND DETACHMENT

A flow diagram of the establishment and detachment of a connection between two devices is shown in [Figure 3.1 on page 691](#). The process is illustrated in 9 distinct steps. A number of these steps may be optionally performed, such as authentication and encryption. Some steps are required, such as the Connection Request and Setup Complete steps. The steps in the overview diagram directly relate to the steps in the following message sequence charts.

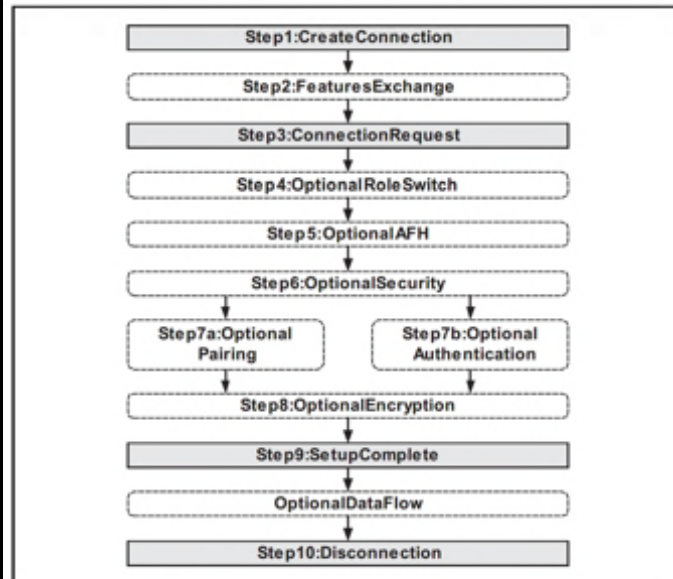


Figure 3.1: Overview diagram for connection setup.

For a POSITA these options are chosen for a connection depending on the requirements of the solution being implemented. In the Bluetooth family of specifications, Profile specifications are used to describe how to implement a particular use case or family of related use cases. A POSITA would listen to (and comply with) recommendations from the v2.1+EDR Bluetooth Core specification or the Bluetooth Basic Imaging Profile (BIP) specification. *See Gajendragad Declaration [Exhibit 2001] at ¶ 80.*

With 15 optional activities, there are tens of thousands of combinations of optional activities after ACL connection establishment. Even if all the options are not mutually exclusive, the number of potential and available combinations is nevertheless extremely large. The Asserted Patents teach one particular combination from this large number of available options. It is only through hindsight reasoning that this one particular option is selected by Petitioner, although it clearly was not a simple or obvious design decision in 2007 when the original patent application was filed. *See Gajendragad Declaration [Exhibit 2001] at ¶ 81.*

Moreover, Petitioner fails to mention the known 2007 pairing disadvantages, such as a poor user experience and battery drain. Even in 2024, there are widely-known “Bluetooth pairing battery issues” that usually refer to a situation where the Bluetooth device, such as headphones or a speaker, rapidly loses battery life when

attempting to pair with another device, often due to factors like a weak battery, improper charging, or the Bluetooth connection itself being unstable or not optimized for low power usage. *See* Gajendragad Declaration [Exhibit 2001] at ¶ 82.

As a POSITA would have understood, maintaining a Bluetooth connection requires power. For example, the Bluetooth radio in your phone consumes energy to stay connected to the earphones, which can lead to faster battery drain. In 2007, a common issue with Bluetooth pairing and battery life was that 2007 Bluetooth technology significantly drained the battery on devices due to its high-power consumption, making it a major concern when pairing devices, especially on portable electronics like phones, as the constant search for connections could quickly deplete the battery. *See* Gajendragad Declaration [Exhibit 2001] at ¶ 83.

Likewise, known Bluetooth battery issues in 2007 were (*see* Gajendragad Declaration [Exhibit 2001] at ¶ 84):

High power usage. Early Bluetooth versions used more power than newer iterations, leading to quicker battery drain when actively searching for or connected to other devices.

Limited battery capacity: Phones and other portable devices in 2007 generally had smaller battery capacities compared to today's standards, making the impact of Bluetooth usage even more noticeable.

Frequent pairing attempts: If you were trying to pair multiple Bluetooth devices or frequently switched between them, it could further exacerbate battery drain due to the continuous search process.

The best way to manage Bluetooth battery usage in 2007 was to turn off Bluetooth when not in use: This was the most effective way to conserve battery life as Bluetooth would only drain power when actively searching for or connected to a device. *See Gajendragad Declaration [Exhibit 2001] at ¶ 85.*

As a POSITA would know, it was these battery issues which led to the development of Bluetooth Low Energy (BLE) because the primary goal of BLE was to significantly reduce power consumption compared to traditional Bluetooth, making it ideal for battery-powered devices like wearables and sensors where long battery life is crucial; essentially, the need to extend battery life in small devices drove the creation of a more power-efficient Bluetooth standard like BLE. As compared to Classic 2007 Bluetooth 2.1, Bluetooth Low Energy was intended to provide considerably reduced power consumption and cost while maintaining a similar communication range. *See Gajendragad Declaration [Exhibit 2001] at ¶ 86; see also en.wikipedia.org/wiki/Bluetooth_Low_Energy*

As of 2007, a POSITA would **NOT** have been inclined to create a system which required pairing devices because the Bluetooth specifications didn't require pairing. Even in 2007 (with Bluetooth 2.1 Security Mode 4) the default security

requirement was an unauthenticated link key, which does not require the entry of a PIN, passkey or comparison of a passkey by the user. More specifically, the Bluetooth specification clearly states that: “[w]hen both devices support Secure Simple Pairing, GAP shall default to requiring an unauthenticated link key and enabling encryption.” (Exhibit TTI 1019, p. 1273 and 1286). *See* Gajendragad Declaration [Exhibit 2001] at ¶ 87.

It would have been these “shall default” Bluetooth options that would have provided for the data security benefits desired by POSITA in 2007. The default connection mode described in the v2.1+EDR Bluetooth core specification is for the connection to be *unpaired, encrypted, unauthenticated and unauthorize*. Exhibit TTI 1019, p. 1273 and 1286. Without some other specific requirement in place for a connection to be paired, there is no reason that a POSITA would not choose the default connection (unencrypted, authenticated and/or authorized). *See* Gajendragad Declaration [Exhibit 2001] at ¶ 88.

In 2007, a POSITA would NOT choose to “maintain a paired connection on a continuous basis,” and Petitioner makes no showing otherwise. *See* Gajendragad Declaration [Exhibit 2001] at ¶ 89. The claim construction the parties have already agreed upon at the District Court is “after the paired connection is established and maintained on a continuous basis.”

In 2007 A POSITA would have understood that the Basic Imaging Profile (BIP) describes all the Kahn, Hiroishi, Feinberg, and Endsley scenarios. However, BIP did not require a paired Bluetooth connection for all these scenarios. Rather, for the scenarios most in line with Kahn, Hiroishi, Feinberg, and Endsley (*i.e.* pull based or push based image transfer), BIP was not recommending using a paired Bluetooth connection. *See* Gajendragad Declaration [Exhibit 2001] at ¶ 90.

NONE of the cited references (Kahn, Hiroishi, Feinberg, Endsley) even mention the word “pair” or “pairing,” let alone the timing for pairing as recited in the Challenged Claims. Applications and systems would connect to Bluetooth and then disconnect after the job was done. The v2.1+EDR Bluetooth Core Specification lists many optional activities, with Pairing and Authentication being only some of them, and neither were the default *for any scenarios*. Data Encryption between the two devices was the default option in 2007, not Pairing and not Authenticated. *See* Gajendragad Declaration [Exhibit 2001] at ¶ 91.

It is apparent that Petitioner and David Lett are relying upon hindsight reasoning to suggest that a POSITA would NOT go with the Default option provided by the Bluetooth Standard of 2007. *See* Gajendragad Declaration [Exhibit 2001] at ¶ 92. According to Petitioner and David Lett, a POSITA would NOT listen to the v2.1+EDR Bluetooth Core specification or NOT go with the recommendation provided by the Bluetooth Basic Imaging Profile (BIP)

specification, and furthermore would NOT care about pairing disadvantages, such as a poor user experience and battery drain. Such hindsight is not warranted and does not reflect the thought process of a POSITA. Even today, a POSITA puts a lot of weight and follows the Bluetooth standards body recommendations. There is reason for the standard bodies to come up with default recommendation, because that is what the latest and best understanding of the problem and the solutions are at the time. To suggest that a POSITA would proceed in a manner that was not being recommended by any standard body is not supported and is unreasonable. *See* Gajendragad Declaration [Exhibit 2001] at ¶ 92.

In *Medichem, S.A. v. Rolabo, S.L.*, 437 F.3d 1157 (Fed. Cir. 2006), the Federal Circuit emphasized that the mere existence of *multiple options in the prior art does not automatically render an invention obvious*. Thousands of permutations and combinations in the Bluetooth specification do not provide a clear direction to combine the references in a way that leads to the claimed invention, and Petitioner makes no showing (absent hindsight) establishing a basis for a POSITA to actually select the combinations used in the Challenged Claims. *See also In re Kubin*, 561 F.3d 1351 (Fed. Cir. 2009) (same).

Similarly, in *Raytheon Technologies Corp. v. General Electric Co.*, 2021 U.S. App. LEXIS 10961 (Fed. Cir., Apr. 16, 2021), the Federal Circuit held that the relied-upon prior art failed to enable one of ordinary skill to make and use the

claimed invention. This decision emphasizes that non-enabled prior art cannot be used to establish obviousness. Here, the 1400 page long Bluetooth Specification with thousands of options is a non-enabled prior art reference that does not provide sufficient information to enable a person of ordinary skill in the art to make and use the claimed invention.

V. SPECIFIC REBUTTAL TO ALLEGED OBVIOUSNESS COMBINATIONS

It is noted that Petitioner has not identified any alleged anticipatory references under 35 U.S.C. § 102. It has, however, identified a number of combinations which purportedly comprise invalidating disclosures under 35 U.S.C. § 103. These arguments are directly refuted by Mr. Gajendragad. *See* Gajendragad Declaration [Exhibit 2001] at ¶ 93.

A. Ground 1A Fails (Hiroishi in Combination with Takahashi and Bluetooth v2.1)

Petitioner and its expert suggest suggests that a POSITA would have been motivated to combine the teachings of Hiroishi in Combination with Takahashi and Bluetooth 2.1 in order to arrive at the inventions of the Challenged Claims. For a variety of reasons, Patent Owner disagrees.

With respect to teachings of Hiroishi, Takahashi, and Bluetooth v2.1, David Lett provides a claim-by-claim articulation of his thoughts regarding how the Hiroishi in combination with Takahashi and Bluetooth V2.1 reference renders the

Challenged Claims obvious. Again, Patent Owner disagrees, and herein addresses the most glaring deficiencies in Petitioner’s analysis. *See* Gajendragad [Exh. 2001] at ¶ 98.

The **first** glaring deficiency is that BOTH parties have already agreed and adopted the claim construction for certain key terms, including:

“after the paired secure Bluetooth connection is established” /	“after the paired connection is established and maintained on a continuous basis between the Bluetooth enabled data capture device and the Bluetooth enabled mobile device” /
“after establishing a paired Bluetooth connection”	“after the paired connection is established and maintained on a continuous basis between the Bluetooth enabled data capture device and the Bluetooth enabled cellular phone”
“after establishing the short-range paired wireless connection” / “after the short-range paired wireless connection is established”	“after the paired connection is established and maintained on a continuous basis between the short-range wireless enabled data capture device and the short-range wireless enabled cellular phone”

Indeed, Petitioner proposed these claim constructions by itself in its District Court disclosures and then agreed to those claim constructions. Petitioner apparently seeks to manipulate these proceedings by advocating inconsistent claim construction positions at this Board and in District Court.

In fact, none of Hiroshi, Takahashi, Kahn or Bluetooth disclose a paired connection that is “established and maintained on a continuous basis,” and

Petitioner has not asserted that any cited reference discloses such a connection. *See* Gajendragad [Exh. 2001] at ¶ 100. Indeed, NONE of the prior art of record (Kahn, Hiroishi, Takahashi) even mention the word “pair” or “pairing,” let alone any **paired connection which is “established and maintained on a continuous basis.”** *See* Gajendragad [Exh. 2001] at ¶ 101.

In 2007, a POSITA would definitely NOT “maintain a paired connection on a continuous basis.” The claim construction the parties have already agreed upon at the District Court is “after the paired connection is established and maintained on a continuous basis.” Once again, it is emphasized that NONE of cited prior art discloses a paired connection that is **“established and maintained on a continuous basis.”** *See* Gajendragad [Exh. 2001] at ¶ 102.

The **second** glaring deficiency is that “pairing” is not a “must” for a Bluetooth connection as suggested by the Petitioner, and both Pairing and Cryptographic Authentication are *optional* and not obvious. *See* Gajendragad [Exh. 2001] at ¶ 103.

Petitioner relies solely on the Bluetooth Specification for support for pairing and cryptographically authenticating – not on any disclosure in Hiroshi or Takahashi or even Kahn. BUT, in the Bluetooth standard 2.1, the default for bi-directional communication with another Bluetooth device is NOT paired and NOT Authenticated; rather, it is just Encrypted. *See* Gajendragad [Exh. 2001] at ¶ 104.

Petitioner argues: “Hiroishi does **not** expressly disclose that the Bluetooth **pairing** between the cellular phone and the digital camera includes using a cryptographic encryption key, it would have been obvious to a POSITA to modify Hiroishi to include this limitation.”

Petitioner further argues: “Bluetooth Core specification describes the process for establishing a secure pairing.” Even though the ‘030 Patent claims do not require cryptographic authentication, Petitioner has included arguments about it in its Petition.

In its institution decision (page 42 for ‘121 Patent), the Board stated that “Petitioner, for example, has demonstrated a reasonable likelihood of prevailing in showing that the connection between Hiroishi’s camera and mobile phone *is “paired,” given that there is two-way communication between the devices via Bluetooth.*”

As a POSITA would understand and appreciate, it is a fact that two Bluetooth devices can have bi-directional two-way communication between them **without ever pairing** (even today). Patent Owner thus disagrees with the above-referenced statement by Petitioner. In fact, in the Bluetooth standard 2.1, the default for bi-directional two-way communication with another Bluetooth device is NOT paired and NOT Authenticated. It is just Encrypted. *See* Gajendragad [Exh. 2001] at ¶ 108.

The words “pair,” “paired,” and “cryptographically authenticating” do NOT appear in any cited art, except the Bluetooth Specification. Once again, the Bluetooth standard makes pairing OPTIONAL. This is because it was NOT done before and not recommended for variety of reasons in 2007 and before. *See* Gajendragad [Exh. 2001] at ¶ 109.

The construction for both “Paired” and “Cryptographically Authenticating” are agreed by the parties in the District Court.

Clearly, the mere mention of Bluetooth in a reference does not mean that it supports pairing or it would be obvious to do so, because several devices in 2007 did not even support pairing in Bluetooth hardware chips. Again, the default 2.1 Bluetooth standard does not require pairing. For example, Bluetooth EDR 2.1 has error codes for when one device attempts to pair with another that does not allow pairing: (1) Pairing Not Allowed (0X18); and (2) Simple Pairing Not Supported (0X37): *See* Gajendragad [Exh. 2001] at ¶ 111.

Bluetooth Pairing is Optionally Practiced and Optionally Supported

BLUETOOTH SPECIFICATION Version 2.1 + EDR [vol 0]		page 15 of 74
Bluetooth		
5.2	Parameter Definitions	313
5.3	LMP Encapsulated	323
5.4	Default Values	324
6	List of Figures	325
7	List of Tables	329
Part D ERROR CODES		
Contents 333		
1	Overview of Error Codes	335
1.1	Usage Descriptions	335
1.2	HCI Command Errors	335
1.3	List of Error Codes	336
2	Error Code Descriptions	338
2.1	Unknown HCI Command (0X01)	338
2.2	Unknown Connection Identifier (0X02)	338
2.3	Hardware Failure (0X03)	338
2.4	Page Timeout (0X04)	338
2.5	Authentication Failure (0X05)	338
2.6	PIN or key Missing (0X06)	338
2.7	Memory Capacity Exceeded (0X07)	338
2.8	Connection Timeout (0X08)	339
2.9	Connection Limit Exceeded (0X09)	339
2.10	Synchronous Connection Limit to a Device Exceeded (0X0A)	339
2.11	ACL Connection Already Exists (0X0B)	339
2.12	Command Disallowed (0X0C)	339
2.13	Connection Rejected due to Limited Resources (0X0D)	339
2.14	Connection Rejected due to Unacceptable BD_ADDR (0X0E)	339
2.15	Connection Rejected due to Security Reasons (0X0F)	340
2.16	Connection Accept Timeout Exceeded (0X10)	340
2.17	Unsupported Feature or Parameter Value (0X11)	340
2.18	Invalid HCI Command Parameters (0X12)	340
2.19	Remote User Terminated Connection (0X13)	340
2.20	Remote Device Terminated Connection due to Low Resources (0X14)	341
2.21	Remote Device Terminated Connection due to Power Off (0X15)	341
2.22	Connection Terminated by Local Host (0X16)	341
2.23	Repeated Attempts (0X17)	341
2.24	Pairing not Allowed (0X18)	341
2.25	Unknown LMP PDU (0X19)	341
26 July 2007		

BLUETOOTH SPECIFICATION Version 2.1 + EDR [vol 0]		page 16 of 74
Bluetooth		
2.26	Unsupported Remote Feature / Unsupported LMP Feature (0X1A)	341
2.27	SCO Offset Rejected (0X1B)	341
2.28	SCO Interval Rejected (0X1C)	342
2.29	SCO Air Mode Rejected (0X1D)	342
2.30	Invalid LMP Parameters (0X1E)	342
2.31	Unspecified Error (0X1F)	342
2.32	Unsupported LMP Parameter Value (0X20)	342
2.33	Role Change Not Allowed (0X21)	342
2.34	LMP Response Timeout (0X22)	342
2.35	LMP Error Transaction Collision (0X23)	343
2.36	LMP PDU Not Allowed (0X24)	343
2.37	Encryption Mode Not Acceptable (0X25)	343
2.38	Link Key Can Not be Changed (0X26)	343
2.39	Requested QoS Not Supported (0X27)	343
2.40	Instant Passed (0X28)	343
2.41	Pairing with Unit Key Not Supported (0X29)	343
2.42	Different Transaction Collision (0X2A)	343
2.43	QoS Unacceptable Parameter (0X2C)	343
2.44	QoS Rejected (0X2D)	344
2.45	Channel Classification Not Supported (0X2E)	344
2.46	Insufficient Security (0X2F)	344
2.47	Parameter out of Mandatory Range (0X30)	344
2.48	Role Switch Pending (0X32)	344
2.49	Reserved Slot Violation (0X34)	344
2.50	Role Switch Failed (0X35)	344
2.51	Extended Inquiry Response Too Large (0X36)	344
2.52	Simple Pairing Not Supported By Host (0X37)	345
2.53	Host Busy-Pairing(0X38)	345
Part E HOST CONTROLLER INTERFACE FUNCTIONAL SPECIFICATION		
Contents 349		
1	Introduction	357
1.1	Lower Layers of the Bluetooth Software Stack	357
2	Overview of Host Controller Transport Layer	359
3	Overview of Commands and Events	360
3.1	Generic Events	361
3.2	Device Setup	361
3.3	Controller Flow Control	362
3.4	Controller Information	362
26 July 2007		

In Bluetooth “Pairing” is different from “Authentication” and both are different from “Encryption. As shown and discussed above, there are 15 different **optional** activities after a Bluetooth connection is established. Pairing, Authentication, and Encryption are different optional activities. Also, as shown in Figure 3.1 in the v2.1+EDR Bluetooth Core Specification [Exhibit TTI 1019, p. 861] it again clearly shows that the Pairing is different from Authentication and both of them are different from Encryption. See Gajendragad [Exh. 2001] at ¶ 112.

Bluetooth Pairing, Authentication and Encryption are Distinct Issues and Each is *Optional* NOT Required or MUST!

BLUETOOTH SPECIFICATION Version 2.1 + EDR [vol 0] page 19 of 74

Bluetooth

- 2.1 Remote Name Request.....685
- 2.2 One-time Inquiry.....687
- 2.3 Periodic Inquiry.....689
- 3 ACL Connection Establishment and Detachment.....691**
 - 3.1 Connection Setup.....692
- 4 Optional Activities After ACL Connection Establishment.....699**
 - 4.1 Authentication Requested.....699
 - 4.2 Simple Pairing Message Sequence Charts.....700
 - 4.3 Link Supervision Timeout Changed Event.....713
 - 4.4 Set Connection Encryption.....714
 - 4.5 Change Connection Link Key.....715
 - 4.6 Change Connection Link Key with Encryption Pause and Resume.....715
 - 4.7 Master Link Key.....717
 - 4.8 Read Remote Supported Features.....719
 - 4.9 Read Remote Extended Features.....719
 - 4.10 Read Clock Offset.....720
 - 4.11 Role Switch on an Encrypted Link using Encryption Pause and Resume.....720
 - 4.12 Refreshing Encryption Keys.....721
 - 4.13 Read Remote Version Information.....722
 - 4.14 QOS Setup.....722
 - 4.15 Switch Role.....723

BLUETOOTH SPECIFICATION Version 2.1 + EDR [vol 2] page 691 of 906

Bluetooth

Message Sequence Charts

3 ACL CONNECTION ESTABLISHMENT AND DETACHMENT

A flow diagram of the establishment and detachment of a connection between two devices is shown in Figure 3.1 on page 691. The process is illustrated in 9 distinct steps. A number of these steps may be optionally performed, such as authentication and encryption. Some steps are required, such as the Connection Request and Setup Complete steps. The steps in the overview diagram directly relate to the steps in the following message sequence charts.

```

graph TD
    S1[Step 1: Create Connection] --> S2[Step 2: Features Exchange]
    S2 --> S3[Step 3: Connection Request]
    S3 --> S4[Step 4: Optional Role Switch]
    S4 --> S5[Step 5: Optional AFH]
    S5 --> S6[Step 6: Optional Security]
    S6 --> S7a[Step 7a: Optional Pairing]
    S6 --> S7b[Step 7b: Optional Authentication]
    S7a --> S8[Step 8: Optional Encryption]
    S7b --> S8
    S8 --> S9[Step 9: Setup Complete]
    S9 --> S10[Step 10: Disconnection]
    
```

For a POSITA these options are chosen for a connection depending on the requirements of the solution being implemented. In the Bluetooth family of specifications, Profile specifications are used to describe how to implement a particular use case or family of related use cases. A POSITA would abide by the recommendations from the v2.1+EDR Bluetooth Core specification or the Bluetooth Basic Imaging Profile (BIP) specification. *See Gajendragad [Exh. 2001] at ¶ 113.*

As noted above, there are tens of thousands of possible combinations of optional activities after ACL connection establishment in Bluetooth. Even if all the options are not mutually exclusive, the number of combinations remain a very large number. The Asserted Patents teach one particular combination from the large number of available options. Again, Petitioner is using hindsight reasoning

for something which was not a simple design decision in 2007. *See* Gajendragad [Exh. 2001] at ¶ 114.

Moreover, and as noted and discussed in detail above, Petitioner fails to account for the known pairing disadvantages, such as a **poor user experience and battery drain**. *See* Gajendragad [Exh. 2001] at ¶ 115.

Again, and as noted and discussed above, the best way to manage Bluetooth battery usage in 2007 *was to turn off Bluetooth when not in use*: This was the most effective way to conserve battery life as Bluetooth would only drain power when actively searching for or connected to a device. *See* Gajendragad [Exh. 2001] at ¶ 116.

Further, as noted and discussed in detail above, as of 2007, a POSITA would **NOT** be inclined to create a system which required pairing devices because the **Bluetooth specifications didn't require** pairing. Even in 2007, and for Bluetooth 2.1 Security Mode 4, the **default security requirement was an unauthenticated** link key, which doesn't require the entry of a PIN, passkey or comparison of a passkey by the user. *See* Gajendragad [Exh. 2001] at ¶ 117.

Moreover, Hiroshi [0067] teaches one to follow the Bluetooth Standards: “Therefore, the communication unit 49 of the mobile phone 40 and the communication unit 59 of the digital camera 50 are configured so as to allow a communication according to the standard of Bluetooth, respectively.” For

Petitioner to suggest otherwise is contrary to Hiroshi's teaching. *See* Gajendragad [Exh. 2001] at ¶ 118.

Again, and as noted and discussed above, the Bluetooth specification clearly states twice that, “[w]hen both devices support Secure Simple Pairing, GAP shall default to requiring an unauthenticated link key and enabling encryption.” (Exhibit TTI 1019, p. 1273 and 1286). *See* Gajendragad [Exh. 2001] at ¶ 119. This statement also means that NOT ALL Bluetooth devices supported pairing. (Conditional statement when both device). *See* Gajendragad [Exh. 2001] at ¶ 120.

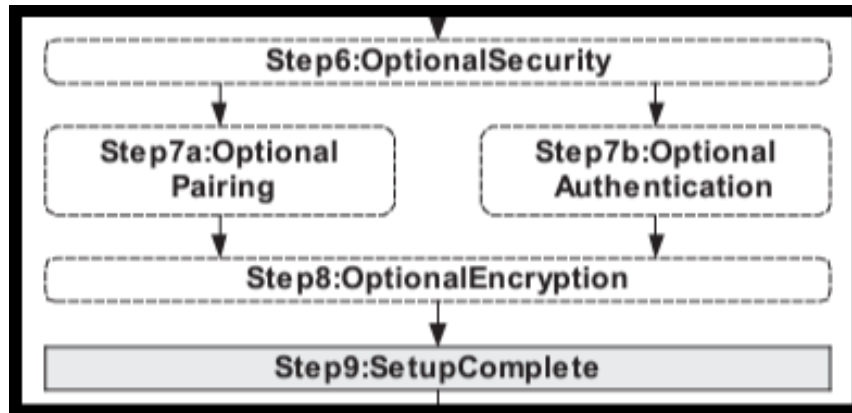
As noted and discussed above, it would have been these “shall default” Bluetooth options that would have provided for the data security benefits desired by POSITA in 2007. The default connection mode described in the v2.1+EDR Bluetooth core specification is for the connection to be *unpaired, encrypted, unauthenticated and unauthorize*. Exhibit TTI 1019, p. 1273 and 1286. Without some other specific requirement in place for a connection to be paired, there is no reason that a POSITA would not choose the default connection (unencrypted, authenticated and/or authorized). *See* Gajendragad [Exh. 2001] at ¶ 121.

Even if one were to look at only the security modes described in the 2006 v2.1+EDR Bluetooth Core specification, which is what a POSITA would have consulted for Bluetooth in the 2007 timeframe, there are significantly more options than just paired or unpaired. *See* Gajendragad [Exh. 2001] at ¶ 122.

As noted and discussed in detail above, in 2007 a POSITA would have understood that the Basic Imaging Profile (BIP) describes all the Kahn, Hiroishi, Feinberg, and Endsley scenarios. However, BIP did not require a paired Bluetooth connection for all these scenarios. For the scenarios most in line with Kahn, Hiroishi, Feinberg, and Endsley, *i.e.* pull based or push based image transfer, BIP was not recommending using a paired Bluetooth connection. *See* Gajendragad [Exh. 2001] at ¶ 123.

As discussed above, NONE of the cited references (Kahn, Hiroishi, Feinberg, Endsley) even mention the word “pair” or “pairing,” let alone the timing for pairing as recited in the Challenged Claims. Applications and systems would connect to Bluetooth and then disconnect after the job was done. The v2.1+EDR Bluetooth Core Specification lists many optional activities, with Pairing and Authentication being only some of them, and neither were the default for any scenarios. Data Encryption between the two devices was the default option in 2007, not Pairing and not Authenticated. *See* Gajendragad [Exh. 2001] at ¶ 124.

Moreover, Petitioner wrongly asserts that: “cryptographically authenticating the identity of phone 40 was a required to pair Hiroishi’s Bluetooth camera 50 and phone 40.” This is unsupported and incorrect. One can have a paired connection without being authenticated, and one can have a authenticated connection without being paired. *See* Gajendragad [Exh. 2001] at ¶ 125.



Hiroshi Does Not Disclose the “Cryptographically Authenticating Identity” Portion of the Limitation. And in Bluetooth specification both are independent and optional elements that can exist and perform without the other. *See* Gajendragad [Exh. 2001] at ¶ 126.

As noted above, Petitioner and David Lett are relying upon hindsight reasoning to suggest that a POSITA would NOT go with the Default option provided by the Bluetooth Standard of 2007. *See* Gajendragad [Exh. 2001] at ¶ 127. According to Petitioner and David Lett, a POSITA would NOT listen to the v2.1+EDR Bluetooth Core specification or NOT go with the recommendation provided by the Bluetooth Basic Imaging Profile (BIP) specification, and furthermore would NOT care about pairing disadvantages, such as a poor user experience and battery drain. Such hindsight is not warranted and does not reflect the thought process of a POSITA. Even today, a POSITA puts a lot of weight and follows the Bluetooth standards body recommendations. There is reason for the

standard bodies to come up with default recommendation, because that is what the latest and best understanding of the problem and the solutions are at the time. To suggest that a POSITA would proceed in a manner that was not being recommended by any standard body is not supported and is unreasonable. *See* Gajendragad [Exh. 2001] at ¶ 127.

The **third** glaring deficiency in Petitioner’s analysis is that Hiroshi or Kahn or Bluetooth does not support event notification for new data *See* Gajendragad [Exh. 2001] at ¶ 128..

Petitioner asserts that the thumbnail sent in Hiroshi is an event notification: “Because thumbnails provide a straight-forward event notification to the user” and “Hiroishi teaches sending a message (thumbnail transmission instruction) to the Bluetooth enabled camera to enable event notifications (thumbnail images).”

The parties agreed to the following constructions in the District Court:

“data signal notification(s)”	“an indication of the presence of newly acquired data”
“poll” / “polling”	“checking status [of] on a predetermined basis,” where “predetermined” refers to programming that sets the frequency of the status checks (e.g., at set intervals, dynamic intervals, randomly, etc.)

In Hiroshi (0083) it is clear that camera receives an instruction/request from the mobile to send a thumbnail. (“If no in Step 118, *i.e.*, when the instruction information received from the mobile phone 40 in Step 104 is not the thumbnail

transmission instruction information, the process goes to Step 122”). *See* Gajendragad [Exh. 2001] at ¶ 131.

In Hiroshi Step 118, a mobile in a remote control mode sends a instruction request for a thumbnail to the camera, and the camera then responds by sending the requested thumbnail to the phone. It is simple request-response scenario. *See* Gajendragad [Exh. 2001] at ¶ 132.

In software design there are two very *different* categories of application architectures. **Request driven and Event Driven.** *See* Gajendragad [Exh. 2001] at ¶ 133. A Request-Driven Architecture uses a **synchronous** request-response model where a component directly asks for information from another *and waits for a response* before proceeding. Request-Driven is more suitable for traditional client-server interactions where a direct response is required. *See* Gajendragad [Exh. 2001] at ¶ 134. An Event-Driven architecture communicates between components by publishing and reacting to events, essentially working **asynchronously**. Event-Driven architecture is better for real-time, loosely coupled systems that need to react to events as they occur. *See* Gajendragad [Exh. 2001] at ¶ 135.

The Key Differences are (*see* Gajendragad [Exh. 2001] at ¶ 136):

Communication Pattern for Event-Driven: Publishes/sends events that any interested component can subscribe/listen to and process it *asynchronously*.

Communication Pattern for Request-Driven: Sends direct requests to a specific component and waits for a response.

Support for these assertions can be found in Exhibit 2010 that has disclosures from three different books: (i) 2003 Book, *C# in a Nutshell*; By Peter Drayton, Ben Albahari, Ted Neward; (ii) 2006 Book, Event-Based Programming, Taking Events to the Limit. By Ted Faison; and (iii) 2009 Book, *C# 3.0 in a Nutshell, A Desktop Quick Reference*, By Joseph Albahari, Ben Albahari. *See* Gajendragad [Exh. 2001] at ¶ 137.

The Challenged Claims require Enabling “Event Notifications” and sending “Event Notification” to a listener without caring for a response from the listener, is completely different from the sending a “request and waiting for a response” architecture in BOTH Hiroshi and Kahn. *See* Gajendragad [Exh. 2001] at ¶ 138.

Moreover, the Challenged Claim uses **plural** for “enabling event notification(S).” But when the claim is sending the event notification it is for single event “sending event notification”. Reaffirming that it is completely different from single request and single response architecture. Enabling event notifications is done **once** for multiple events to be received in the future. Unlike simple request response architecture. Petitioner’s argument conflates two vastly different scenarios, which a POSITA would not do. *See* Gajendragad [Exh. 2001] at ¶ 139.

Petitioner relies on the *thumbnail transmission instruction* as a purported “message to enable event notifications.” (“Hiroishi teaches sending a message (thumbnail transmission instruction) to the Bluetooth enabled camera to enable event notifications (thumbnail images)”; “the mobile phone sends the camera a *thumbnail transmission instruction*, which is a message to enable event notifications corresponding to the new data”). See Gajendragad [Exh. 2001] at ¶ 140.

First, it is undisputed that Hiroishi clearly says that the referenced “*thumbnail transmission instruction*” is the information for instructing the transmission of the image data. See Hiroishi (Exh. 1011) at [0068] (“*Thumbnail transmission instruction* information ... is the information for instructing the transmission of the image data showing the thumbnail of the image shown according to the digital image data stored in SDRAM 55A”). As such, it is plainly *not* “to enable event notifications.” (Plural Notifications not single request response). See Gajendragad [Exh. 2001] at ¶ 141.

Second, the “enablement of event notifications” as claimed is a single occurrence, and would not be characterized or satisfied by a transmission associated with each and every new image taken. Under Petitioner’s view; namely, that the transmission of thumbnail instructions is performed “to enable event notifications,” then the process of “enabling event notifications” would be required

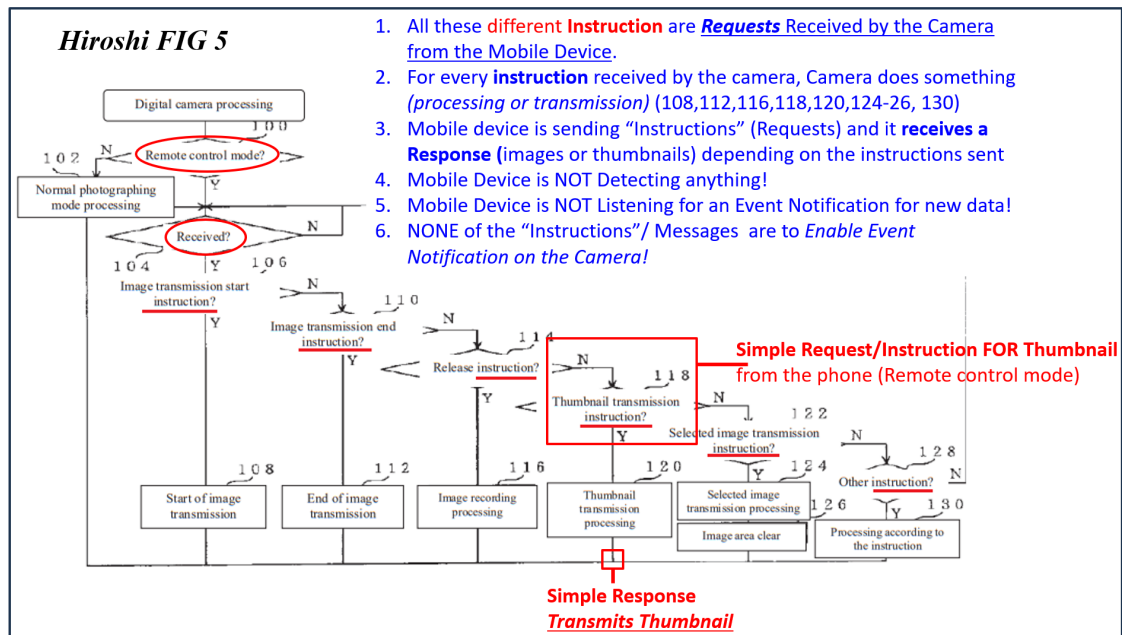
for every new image requested by the phone. This is plainly not the way in which a POSITA would understand “enabling event notifications” to work, and would wholly defeat the purposes of enabling event notifications in the first place. *See* Gajendragad [Exh. 2001] at ¶ 142.

A reasonable POSITA would have understood that a mobile device sending a “thumbnail transmission instruction” to the camera, followed by the receipt of the thumbnail, *is a simple request-response transaction* between the two devices that needs to be done *for each new thumbnail*. It is plainly *not* an “enabling of event notifications for the new data” by any means. (enabling plural number of notifications is NOT the same as single request response for thumbnail). *See* Gajendragad [Exh. 2001] at ¶ 143.

The word “event” or “event notification” itself does not exist in either Kahn or Hiroshi. This is unsurprising, given the fact that both those state machines are based on request-response architecture. For example, ALL instructions follow the same request-response pattern in Hiroshi and Khan. The phone sends an instruction to the camera and then waits for a response. *See* Hiroshi at [0092-93]: “In Step 206, image transmission start instruction information is transmitted via the communication unit 49 to the digital camera 50. In response to this, the digital camera 50 performs Step 108 of the digital camera processing (also see FIG. 5) to start an image transmission processing. So, in the following Step 208, [the phone]

waits for the reception of the digital image data transmitted from the digital camera 50,” See Gajendragad [Exh. 2001] at ¶ 144.

As illustrated in the annotated Figure below (see Hiroishi (Exh. 1011) at Fig. 5, as annotated), the process of Hiroishi simply calls for a sequence of instructions sent from the mobile device to the camera (see Gajendragad [Exh. 2001] at ¶ 145):



As shown above all the instruction in Hiroshi (108, 112, 116, 118, 120, 124, 126, and 130) follow a simple request-response architecture. For every instruction sent by the mobile device to the camera, the phone waits for the response back from the camera. After receiving the request from the phone, the camera does the processing and transmits the data back for the waiting phone. This is a classic **synchronous** request-response model, and has NOTHING to do with Event

Notifications that are *asynchronous in nature*. See Gajendragad [Exh. 2001] at ¶ 146.

As a POSITA would have readily understood, a “request-response” model involves a client actively sending a specific request to a server and waiting for a direct response, while an “event notification” is a message sent out when a significant event occurs in a system, allowing any interested components to react asynchronously without needing to explicitly request the information; essentially, the key difference is that request-response is a direct, *synchronous* communication while event notification is a broadcast-like, *asynchronous* communication where the receiver decides when to act upon the information. Support for these assertions can be found in the Exhibit 2010. See Gajendragad [Exh. 2001] at ¶ 147.

In request-response, the client initiates communication by sending a request. In event notification, the system generates an event that triggers the notification. Request-response is synchronous, meaning the client waits for the server’s response before continuing. Event notification is asynchronous, meaning components can react to events at their own pace. Under that architecture, the mere receipt of a reply to an instruction/request in Hiroshi cannot be reasonably understood as an “event notification” as claimed. See Gajendragad [Exh. 2001] at ¶ 148.

Petitioner initially relies on the *thumbnail transmission* as a purported event notification. More specifically, Petitioner asserts that the thumbnail sent in Hiroshi is an event notification: “a thumbnail and/or data related to the thumbnail would operate as a notification signal,” and “transmission of thumbnails as event notifications.” Petitioner’s position is untenable, inasmuch as **there is nothing in Hiroishi that performs as an event notification**. Again, Step 118 of receiving a request from the mobile to send a thumbnail and then camera sending the thumbnail to the mobile is NOT an event notification, in any sense of the word for a POSITA. *See* Gajendragad [Exh. 2001] at ¶ 149.

Petitioner fails to appreciate that in any application data model, you have to follow **either** request-response or event-based notification. Here the Petitioner seeks to combine two orthogonal data architectures. One does not “wait for a response” in an “event notification” architecture. Nevertheless, Petitioner acknowledges and asserts that Hiroshi is “waiting for a response” which, by definition, is not an “event notification.” Because in Event Notifications there is NO waiting for a response. Simply put, Hiroishi has very little in common with the teachings and claims of the ‘030 Patent. *See* Gajendragad [Exh. 2001] at ¶ 150.

Petitioner further takes the position that the “waiting” component of Hiroishi “enables event notification listening” for event notifications that are for new data. On inspection, a POSITA would readily appreciate that the application of the

“wait” in Hiroishi varies throughout the disclosure, but in no sense is it *ever* used to define any act of enabling event notification listening. *See* Hiroishi (Exh. 1011) at [0002] (“... waits for the shutter to release”); at [0073] (“... [the camera] waits to receive the instruction information from the mobile phone”); at [0091] (“... [the phone] waits for an operator to specify a desired mode”); at [0093] (“... [the phone] waits for the reception of the digital image data transmitted from the digital camera”); at [0097] (“... [the phone] waits until an operator presses any key”); at [0104] (“...[the phone] waits until it receives the image data that can display the thumbnail”); at [0106] (“... [the phone] waits for the end of selection of the desired image”); at [0107] (“... [the phone] waits for the reception of the digital image data”); and at [0111] (“... the device waits for the operator to finish entering each information”). *See* Gajendragad [Exh. 2001] at ¶ 151.

Plainly, the act of merely “waiting” on an event to occur, as described in Hiroishi, cannot reasonably be understood to convey an “*enabling event notification listening* on Bluetooth enabled mobile device, to listen for event notifications sent from the Bluetooth enabled data capture device” corresponding to existence of new data. Again, simply waiting for a response to an earlier instruction is *not* an enabling of event notifications for new data. *See* Gajendragad [Exh. 2001] at ¶ 152.

A POSITA would know and appreciate that every file that is created on a computer or camera or phone has a timestamp. And a POSITA would further know that a timestamp inside a file is not an event notification in any sense of the word. *See* Gajendragad [Exh. 2001] at ¶ 153.

Neither Hiroshi nor Kahn has been configured to “listen” to those timestamps. Receiving a file that you requested and that every file has timestamp is not “listening” for a timestamp, as any POSITA would recognize. *See* Gajendragad [Exh. 2001] at ¶ 154.

The word “event” or “event notification” itself is not found in either Kahn or Hiroshi. This is unsurprising, given the fact that both those state machines are based on request-response architecture. A “request-response” model involves a client actively sending a specific request to a server and waiting for a direct response, while an “event notification” is a message sent out when a significant event occurs in a system, allowing any interested components to react *asynchronously* without needing to explicitly request the information. Support for these assertions can be found in Exhibit 2010. *See* Gajendragad [Exh. 2001] at ¶ 155.

As noted and discussed in detail above, under proper construction, it is very clear (including based on Exhibit 2010) that waiting for a response is outside of the

meaning of the claim term “listen[ing]” for “event notification.” *See* Gajendragad [Exh. 2001] at ¶ 156.

Still further, Petitioner makes the curious argument that the “Sniff Mode” described in the Bluetooth Specification somehow illustrates a form of “sending a message to enable event notifications.” The deficiencies in this argument are many, but primarily it must be recognized that “Sniff Mode” (as defined in the Bluetooth Specification) is clearly a Layer 2 feature (LMP Link Manager Protocol), and has *nothing* to do with sending a message from the mobile device to the camera for enabling event notifications. LMP_Snif_req and LMP_unsnif_req have nothing to do with new data event notification *See, e.g.*, Bluetooth Specification (Exh. 1019) at 456, as excerpted and annotated below. *See* Gajendragad [Exh. 2001] at ¶ 157.

When the ACL logical transport is in **sniff mode** the master shall only start a transmission in the sniff slots. Two parameters control the listening activity in the slave: the sniff attempt and the sniff timeout. The sniff attempt parameter determines for how many slots the slave shall listen when the slave is not treating this as a scatternet link, beginning at the sniff slot, even if it does not receive a packet with its own LT_ADDR. The sniff timeout parameter determines for how many additional slots the slave shall listen when the slave is not treating this as a scatternet link if it continues to receive only packets with its own LT_ADDR. It is not possible to modify the sniff parameters while the device is in **sniff mode**.

M/O	PDU	Contents
O(7)	<u>LMP_sniff_req</u>	timing control flags D _{sniff} T _{sniff} sniff attempt sniff timeout
O(7)	<u>LMP_unsniff_req</u>	-

Table 4.28: **Sniff mode** PDUs

Sniff Mode is a Layer 2 concept and is used to reduce power

- 1. It has *Nothing to do* with Event Notification for new images!**
- 2. It has *Nothing to do* with Enabling Event Notifications**

**LMP_sniff_req and LMP_unsniff_req
LMP Stands for “Link Manager Protocol”**

26 July 2007

Procedure Rules

456

As a POSITA would recognize, mobile applications don’t even have access to hardware level Link Layer (layer 2) LMP Link Manager Protocol. *See also id.* at 386 and 395 (as excerpted and annotated below) (*see* Gajendragad [Exh. 2001] at ¶ 158):

Sniff mode	This feature indicates whether the device is able to support sniff mode as defined in baseband Section 8.7 on page 175 using the LMP sequences defined in Section 4.5.3 on page 286 .
Sniff subrating	This feature indicates whether the device is able to support sniff subrating as defined in [PART B] Sniff Subrating on page 177 using the LMP sequences as defined in Sniff Subrating on page 288 . This feature bit shall be set if the " Sniff mode " feature bit is set.
Park state	This feature indicates whether the device is able to support Park state as defined in baseband Section 8.9 on page 179 using the LMP sequences defined in Section 4.5.2 on page 279 .

Table 3.1: Feature Definitions

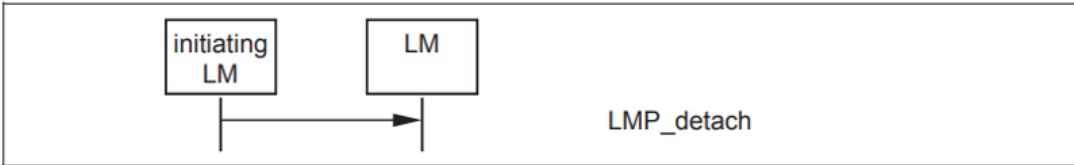
26 July 2007
Device Features

386

BLUETOOTH SPECIFICATION Version 2.1 + EDR [vol 2]
page 225 of 906

Link Manager Protocol


mode or park state, the initiating LM shall perform the procedure to exit **sniff mode** or park state before initiating the procedure defined above. If the procedure to exit **sniff mode** or park state does not complete within the LMP response timeout (30 seconds) the procedure defined above shall be initiated anyway.



```

sequenceDiagram
    participant A as initiating LM
    participant B as LM
    A->>B: LMP_detach

```

Sequence 1: Connection closed by sending LMP_detach.

A POSITA would understand that Bluetooth Sniff Mode messages have nothing to do with event notifications, and Bluetooth Sniff Mode has nothing to do with new data/photos being taken by the camera. **The term “Event Notification”**

does not exist in the Bluetooth 2.1 Specification. *See* Gajendragad [Exh. 2001] at ¶ 159.

The **fourth** glaring deficiency is that “object” and “file” are NOT the same. While Claims 6 and 15 recite creating an “Object,” the regular file with a timestamp of Hiroshi or Khan **cannot** be considered as an “Object.” *See* Gajendragad [Exh. 2001] at ¶ 160.

Petitioner asserts that: “Takahashi discloses that the new data picture is saved in a file along with a first timestamp of the time the picture was taken,” and further that: “Kahn teaches that “the image data already includes timestamp information (*i.e.*, timestamp indicating when the image is captured).”

As noted and discussed in detail above, the term “object” is also disputed by the parties in the District Court case, and that Cellspin proposes the following: “An object can be a variable, a data structure, a function, or a method. Object contains value and is referenced by an identifier.” *See* Gajendragad [Exh. 2001] at ¶ 162.

As further noted and discussed in detail above, the evidence clearly shows that an Object is different from a File. As further discussed in detail above, a POSITA would readily appreciate the fact that a “file” has to be converted to be an “Object” to be sent using OBEX. Similarly, “Data” (that is NOT itself a “File”) can also be converted into an Object and then sent/transferred using OBEX protocol. OBEX is an **object exchange** protocol, **NOT** a file exchange or data

exchange protocol, but both file and data can be sent using it by creating an Object first. *See* Gajendragad [Exh. 2001] at ¶ 163.

The **fifth** glaring deficiency is that (assuming “along with” is properly construed), Takahashi *does not* teach sending User Identifier “Along With” the New Data. *See* Gajendragad [Exh. 2001] at ¶ 164.

As noted and discussed in detail above, the term “along with” should be construed as “in addition to,” and which expressly excludes simply “inside.”

Petitioner argues that combinations involving the Takahashi reference (Exh. 1014) obviate the Challenged Claims. One primary use of Takahashi, according to Petitioner, is its disclosure of the User ID with the file name. Indeed, Petitioner claims: “A POSITA would have been motivated, and found it obvious, to include user identifier that uniquely identifies a particular user of the Internet service as taught by Takahashi.” Petitioner also argues: “If desired, a POSITA would have found it obvious to also include the user identifier with the images when transferring the images to the remote internet server.” (citing Takahashi at [0066] (“user ID in the file name of image data”)).

These teachings of Takahashi, however, do not satisfy the “along with” limitation which, as discussed above, is “in addition to.” **As taught in Takahashi, the User ID is part of the filename, not an identifier transmitted “in addition to” the new data.** *See* Gajendragad [Exh. 2001] at ¶¶ 166-167.

B. Ground 1B Fails (Hiroishi in Combination with Kahn and Bluetooth v2.1)

Petitioner and its expert suggest suggests that a POSITA would have been motivated to combine the teachings of Hiroishi in Combination with Kahn and Bluetooth 2.1 in order to arrive at the inventions of the Challenged Claims. For a variety of reasons, Patent Owner disagrees.

With respect to teachings of Hiroishi, Kahn, and Bluetooth v2.1, David Lett provides a claim-by-claim articulation of his thoughts regarding how the Hiroishi in combination with Kahn and Bluetooth v2.1 reference renders the Challenged Claims obvious. Again, Patent Owner disagrees, and herein addresses the most glaring deficiencies in Petitioner's analysis. *See* Gajendragad [Exh. 2001] at ¶ 169.

As an initial point, the main reason Petitioner relies on Kahn is for HTTP and “object creation” support. *See* Gajendragad [Exh. 2001] at ¶ 170.

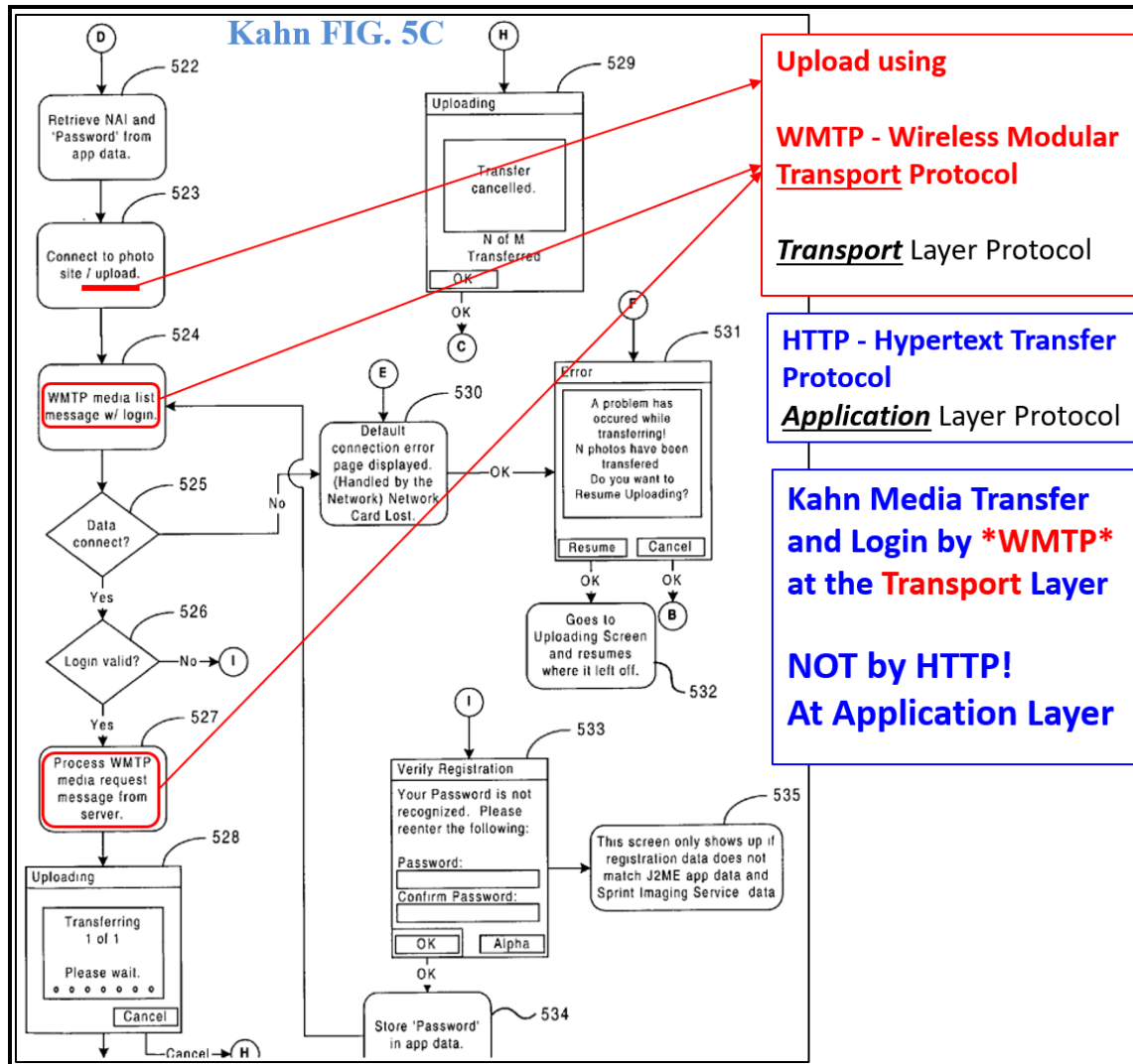
As discussed in detail above, a File is different from an Object, and this same reasoning applies the timestamp of Kahn. As a POSITA would recognize, every computer file has a timestamp. In any event, Kahn does *not* teach sending user information and user data (photos/media) using HTTP. *See* Gajendragad [Exh. 2001] at ¶ 171.

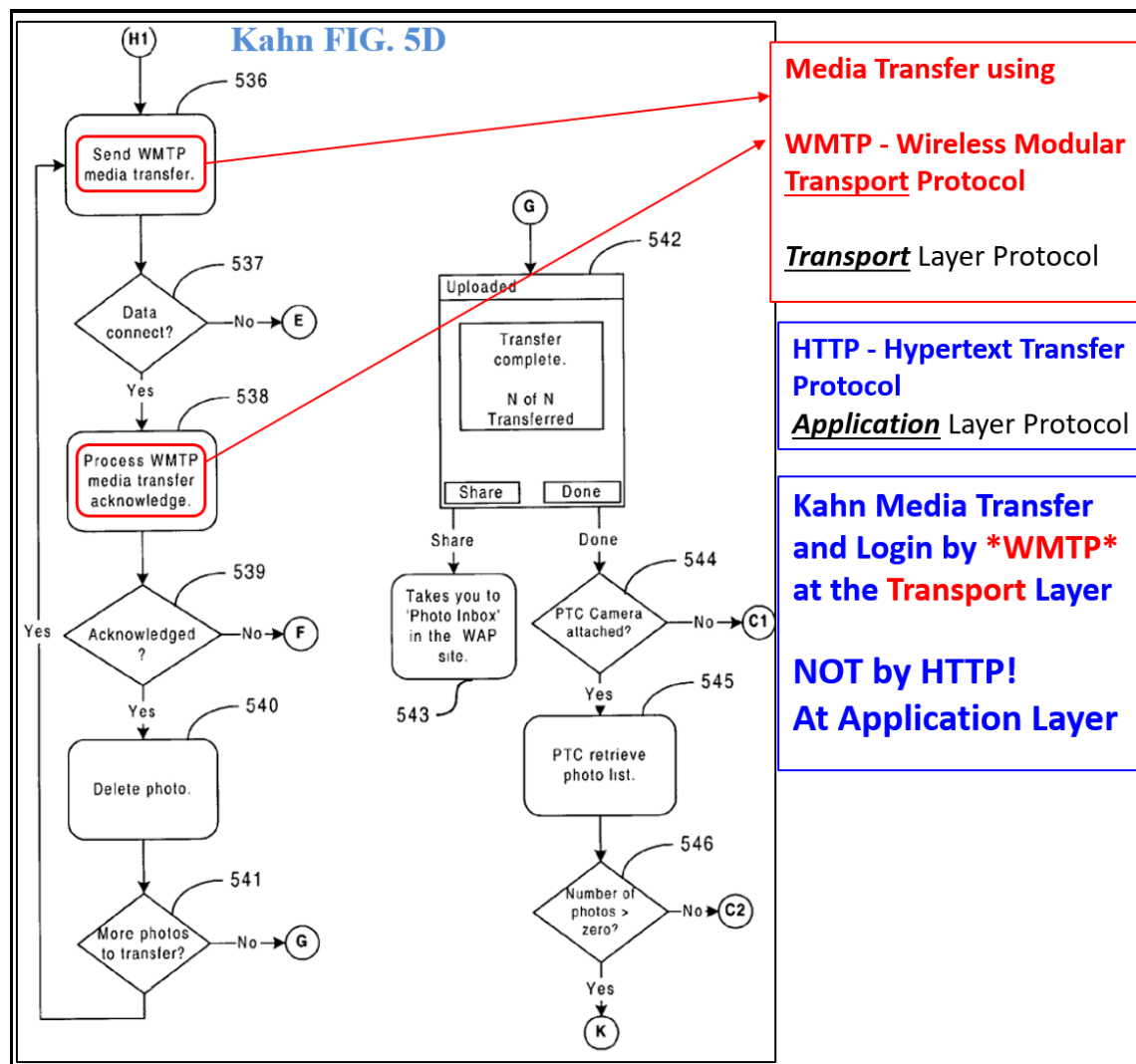
The biggest and most obvious deficiency is that Kahn does NOT teach use of HTTP for uploading photos/media. Instead, Kahn required using WMTP for

media upload. However, a POSITA would recognize the fact that HTTP and WMTP are NOT interchangeable. *See* Gajendragad [Exh. 2001] at ¶ 172.

Wireless Application Protocol (WAP) and Wireless Modular Transport Protocol (WMTP) are the wireless communication technique (circa 2004), and Hyper Text Transfer Protocol (HTTP) is the *standard protocol for the web*. When Kahn says the upload process “occurs via wireless communication technique,” it is in reference to wireless WAP and wireless WMTP protocols (*not* HTTP). *See* Gajendragad [Exh. 2001] at ¶ 173.

It is very clear with respect to **media transfer and login** that Kahn teaches the *exclusive use of WMTP*. *See* Kahn Fig 5C and Fig 5D as reproduced below (*see* Gajendragad [Exh. 2001] at ¶ 174):





Kahn [0057] makes it even more clear that the **Wireless** Modular Transport Protocol (WMTP), a wireless communication technique, is exclusively used for uploading. See Kahn at [0057]: “[T]he *uploading process occurs via **wireless** communication technique*, with the local host (e.g., cell phone, wireless PDA, or the like) assuming responsibility for establishing a wireless data connection.” As taught by Kahn, a POSITA is to use WMTP as described in Fig 5C and Fig 5D. It is

understood that WMTP is a wireless protocol and HTTP is and was a web protocol. *See* Gajendragad [Exh. 2001] at ¶ 175.

Kahn Fig 5D, **element 543**, also mentions another wireless communication technique: WAP (**Wireless** Application Protocol). In 2004 (the time of Kahn), the industry was pushing for wireless protocols like WAP and WMTP (**Wireless** Modular Transport Protocol) for wireless mobile device for web access. WAP and WMTP were specifically designed for mobile devices with limited bandwidth, with WAP acting as a gateway to translate requests from a mobile phone into standard HTTP requests, whereas HTTP (Hypertext Transfer Protocol) is the standard protocol used for web browsing on traditional computers. WMTP and WAP were particularly relevant in the early days of mobile internet when data capabilities were significantly lower. *See* Gajendragad [Exh. 2001] at ¶ 176.

When Kahn references “wireless communication techniques” in [0057] for “uploading,” it very clearly shows two wireless protocols WMTP and WAP in Fig 5D. It is thus clear that Kahn wants to use wireless protocols like WMTP for upload and NOT web protocol HTTP. *See* Gajendragad [Exh. 2001] at ¶ 177.

In its Institution decision relating to the '121 Patent (at page 41), the Board mentions that: “While the term ‘WMTP’ appears in these figures, it does not seem to be defined or discussed elsewhere in Kahn.”

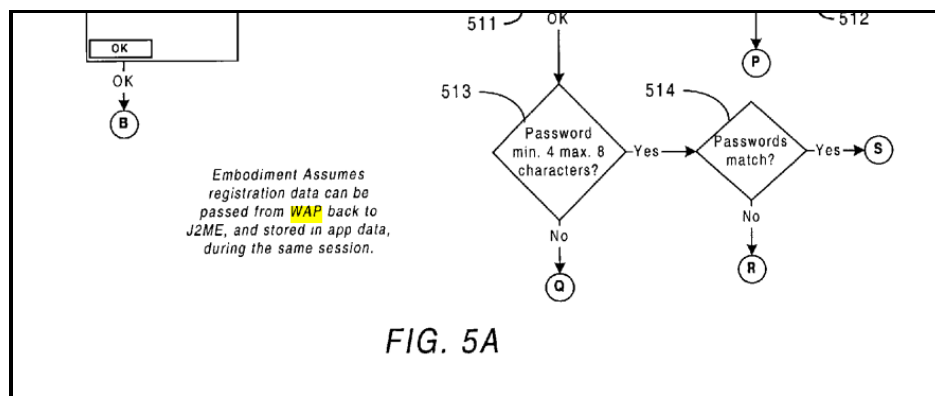
The reason for this is because Kahn assumes the POSITA knows about WAP architecture and usage in the wireless domain. Just like a POSITA for the wired network would know about TCP/IP or OSI stack, Kahn assumes the POSITA is knowledgeable about the WAP architecture and WMTp in the WAP system. Indeed, Kahn talks about wireless transmission techniques and in the detailed diagrams it mentions WMTp and WAP. *See* Gajendragad [Exh. 2001] at ¶¶ 178-179.

In a WAP (Wireless Application Protocol) architecture system, the “wireless modular transport protocol” (WMTp) refers to the Wireless Datagram Protocol (WDP), which acts as the transport layer within the WAP stack, providing a flexible way to transfer data across different network types, even when dealing with unreliable or limited bandwidth connections typical in mobile networks; essentially acting as a modular adaptation layer for various underlying bearer services. *See* Gajendragad [Exh. 2001] at ¶ 180.

Again, it is clear that Kahn intends for the POSITA to use WAP architecture and system. **Kahn talks in great detail about WAP and WAP specification in [0017]** and assumes POSITA to know about WAP architecture, and WAP usage in the wireless domain (WMTp being the transport layer protocol in the WAP architecture in 2004). *See* Khan at [0017]: “**WAP**: Short for the Wireless Application Protocol, a Secure Specification that allows users to access

information instantly via handheld wireless devices Such as mobile phones, pagers, two-way radios, ‘Smart’ phones, and communicators. WAP uses WML or ‘Wireless Markup Language’ which is a tag-based display language providing navigational Support, data input, hyperlinks, text and image presentation, and forms, it is a browsing language Similar to Internet HTML. **WAP supports most wireless networks**, including CDMA and GSM. The **WAP 2.0 specification** is available from Open Mobile Alliance, Ltd.” *See* Gajendragad [Exh. 2001] at ¶ 181.

Further, WAP is also described in Kahn **Fig 5A**, and it is clear that in Kahn 5A, WAP is used to connect the mobile app (J2ME Applet in 2004) to the outside internet for login registration and password. It is NOT HTTP, but WAP that is used for login password and registration (*see* Gajendragad [Exh. 2001] at ¶ 182):



WAP is also described in Kahn **Fig 5G**, which talks about user account that is a “user’s WAP Account.” This is plainly not an HTTP account. *See* Gajendragad [Exh. 2001] at ¶ 183.

Glossary for J2ME App	
Yes-	Confirms action and takes user to next screen
No-	Cancels action and takes user back to previous screen
OK-	Confirms action and takes user to next screen
Done-	Completes task and takes user to initial page
Cancel-	Cancels action and takes user to previous screen
Exit-	Exits application
Upload-	Initiates transfer of selected photo to user's account
Upload All-	Initiates transfer of all photos to user's account
Delete-	Removes selected photo from camera
Delete All-	Removes all photos from camera
Preview-	View photos in camera
Share-	Closes J2ME app and launches to user's WAP account
Next-	Displays next photo
Prev-	Displays previous photo

FIG. 5G

In 2004, WAP (Wireless Application Protocol) was considered better than HTTP (Hypertext Transfer Protocol) for mobile devices due to several reasons (*see* Gajendragad [Exh. 2001] at ¶ 184):

1. Optimized for Mobile Devices: WAP was specifically designed for the limited capabilities of mobile devices at the time, such as small screens, low memory, and limited processing power. It used WML (Wireless Markup Language) instead of HTML, which was more suitable for mobile displays.

2. Efficient Use of Bandwidth: WAP was optimized for narrow-band bearers with potentially high latency, making it more efficient in terms of bandwidth usage compared to HTTP. This was crucial for mobile networks, which had slower data speeds and higher latency.

3. Security: WAP included Wireless Transport Layer Security (WTLS), which provided a public-key cryptography-based security mechanism similar to TLS. This was important for ensuring secure communication over potentially unprotected mobile networks.

4. Adaptation to Wireless Networks: WAP was designed to work with various wireless data networks, such as GSM and CDMA, and included protocols like Wireless Datagram Protocol (WDP) and Wireless Transaction Protocol (WTP) that were adapted to the wireless environment. *These features made WAP more suitable for the mobile internet experience in that timeframe.*

Kahn [0056] does talk about using HTTP to communicate with the image server, BUT it is for Ecommerce purpose. This whole text of [0056] where HTTP is mentioned is about the cellphone taking to an image server for e-commerce and processing fulfilment or ordering hardcopies and enlargement. In any event, in Kahn [0056] there is no mention of using HTTP to upload photos. *See* Gajendragad [Exh. 2001] at ¶ 185.

Kahn [0062-0063] is another example of Kahn talking about using HTTP for **internal communication** within the Cellphone with hardware GPS chip by Qualcomm SnapCore that is inside the phone itself, by using location API. Here HTTP is being used within the host/cellphone to communicate with the hardware

chip *SnapCore* to get GPS data from the hardware Chip that resides inside the phone itself. The Location API call is **not** going out to the server on the internet, it is to-and-from the Qualcomm chip on the phone. *See* Gajendragad [Exh. 2001] at ¶ 186.

In sum, Kahn is very clear to the POSITA on what protocols to use and where. Specifically: HTTP for e-commerce and internal hardware chip location API. As for uploading/media-transfer and login information, on the other hand, Kahn is very clear to POSITA to use WMTP. *See* Gajendragad [Exh. 2001] at ¶ 187.

Petitioner's assertion that POSITA would use HTTP instead of the clear teaching of WMTP in Kahn for uploading/media-transfer/login is not supported at all in Kahn, and does not reflect the approach to be taken by a POSITA. *See* Gajendragad [Exh. 2001] at ¶ 188.

Moreover, WMTP and HTTP are very different protocols and they are NOT Interchangeable. A POSITA would understand and appreciate that both WMTP and HTTP run on different layers of the OSI stack. They are NOT interchangeable and there are significant differences between these two protocols. The key difference lies in their primary purpose and application domain: HTTP (Hypertext Transfer Protocol) is the standard protocol for web browsing and data transfer on the internet, while Wireless Modular Transport Protocol (WMTP) is

specifically designed for communication between devices within a wireless modular network, often used in industrial scenarios where reliability and low latency are critical. *See* Gajendragad [Exh. 2001] at ¶ 189; *see also* <https://ieeexplore.ieee.org/document/4689157>.

Abstract:

A new modular transport layer protocol for wireless sensor networks (WSNs) is proposed: WMTP - Wireless Modular Transport Protocol. This protocol not only allows the simultaneous use of all the main features commonly found in WSN transport protocols, namely congestion-control, fairness, and reliability, but also does so in a modular fashion. This way, the application layer can choose to use exactly the features that it requires. Additionally, WMTP also provides a set of uncommon features such as throttling, flow-control, transport layer quality- of-service, and optional integration with service-discovery. Focusing specifically on the fairness feature, experimental results show that the network resources are appropriately divided whether a simple fairness semantic is used, or in the presence of weighted differentiation. Additionally, this feature is shown to operate with minimal energy cost, as the protocol overhead is less than 10%.

There are a number of clear differences between HTTP and WMTP. Among them are the following (*see* Gajendragad [Exh. 2001] at ¶ 190):

Key points about HTTP: It is at the *seventh layer* of standards in (OSI) reference model. *Function:* Primarily used for transferring web pages and data between web servers and client devices like web browsers. *Request-Response model:* Operates on a clear request-response mechanism where a client sends a request to a server and receives a response. *Adopted:* as the standard protocol for web communication.

Key points about WMTP: It is at the *fourth layer* of standards in (OSI) reference model. *It Focus on wireless networks:* Designed to facilitate reliable communication between devices within a wireless modular network,

often in industrial settings. *Low latency and reliability*: Prioritizes fast data transfer with high reliability, crucial for real-time control applications. *Specialized use case*: Not as widely used as HTTP, primarily targeted towards applications where devices need to communicate wirelessly in a controlled environment.

Functionality of Transport layer protocols and Application layer protocols are very different. Function of Transport layer protocols like WMTP, TCP, UDP is to manage data segmentation, error checking, and flow control. Function of application layer protocols like HTTP, FTP is to define the structure and content of data for specific applications.

Abstraction level needed and required for both WMTP and HTTP is very different. The transport layer operates at a lower level providing a reliable communication channel. Application layer operates at a much higher layer, allowing applications to focus on their specific tasks without worrying about network complexities.

The reason POSITA will not go against teaching of Kahn for using WMTP and switching to HTTP is because the two protocols are NOT Interchangeable. It is not like switching from FTP to *Secure* FTP. It is even not like switching from FTP to HTTP where both are at the application layer protocol (not recommended). See Gajendragad [Exh. 2001] at ¶ 191.

In sum, HTTP is application layer protocol that is the standard for web-based data transfer, while WMTP is a specialized transport layer protocol for wireless communication in specific industrial environments, emphasizing reliability and low latency within a dedicated network. They are NOT Interchangeable. *See* Gajendragad [Exh. 2001] at ¶ 192.

C. Ground 2A Fails (Singh 129 in View of Bluetooth)

D. Ground 2B Fails (Singh 129 in View of Bluetooth and Takahashi)

Petitioner and its expert suggest suggests that a POSITA would have been motivated to combine the teachings of Singh 129 in view of Bluetooth 2.1 in order to arrive at the inventions of the Challenged Claims. For a variety of reasons, Patent Owner disagrees.

With respect to teachings of Singh 129 and Bluetooth 2.1, David Lett provides a claim-by-claim articulation of his thoughts regarding how the Singh 129 in view of Bluetooth 2.1 reference renders the Challenged Claims obvious. Again, Patent Owner disagrees, and herein addresses the most glaring deficiencies in Petitioner's analysis. *See* Gajendragad [Exh. 2001] at ¶ 194.

For both Grounds 2A and Ground 2B, Petitioner has raised § 112 issues regarding three claim elements of the '030 Patent, and specifically, Petitioner alleges that there is a purported lack of support in the specification for: (i) send a message to the Bluetooth enabled mobile device as claimed / receive a message from the Bluetooth enabled data capture device corresponding to a supported data

capability of the Bluetooth enabled data capture device; (ii) include a timestamp in with the new data object; and (iii) encrypt/decrypt the data. These are addressed below.

It is noted that the Patent Examiner did not raise any § 112 objections for any of the three issues Petitioner has now raised. This is unsurprising, given the fact that each of these limitations are disclosed in the specification. *See* Gajendragad [Exh. 2001] at ¶ 197.

FIRST: Petitioner complains about sending/receiving messages to negotiate “supported data capability” for claim 1 and 12.

It would have been well known to a POSITA that “Handshake Protocol” is used for communicating / negotiating the data-and-other capabilities from one device to another device. *See* Gajendragad [Exh. 2001] at ¶ 199.

In this regard, the specification is very clear about implementing the *handshake protocol*: “**By implementation of a handshake protocol**, the BT communication device **201** *a* automatically transfers captured data, the multimedia content, and the associated files to the client application **203** on the mobile device **202.**” *See* ’030 Patent at 4:59-62.¹ *See* Gajendragad [Exh. 2001] at ¶ 200.

¹ The ’030 Patent specification will be used as a reference for the written description throughout (unless noted otherwise), as it contains the same disclosure for all 7 Challenged Patents.

The claim limitation Petitioner is complaining about was suggested, proposed, and added by the Examiner himself during the **Examiner Initiated Interview**. (Page 30,31 and 44 of Exhibit 2011) (*see* Gajendragad [Exh. 2001] at ¶ 201):

Examiner-Initiated Interview Summary	Application No. 14/295,353	Applicant(s) SINGH ET AL.	
	Examiner SULAIMAN NOORISTANY	Art Unit 2478	

All participants (applicant, applicant's representative, PTO personnel):

(1) SULAIMAN NOORISTANY. (3) ____.

(2) Ashok Tankha (Reg. # 33,802). (4) ____.

Date of Interview: 23 September 2014.

Type: ☒ Telephonic ☐ Video Conference
☐ Personal [copy given to: ☐ applicant ☐ applicant's representative]

Exhibit shown or demonstration conducted: ☐ Yes ☐ No.
If Yes, brief description: ____.

Issues Discussed ☐101 ☐112 ☐102 ☐103 ☒Others
(For each of the checked box(es) above, please describe below the issue and detailed description of the discussion)

Claim(s) discussed: ____.

Identification of prior art discussed: ____.

Substance of Interview
(For each issue discussed, provide a detailed description and indicate if agreement was reached. Some topics may include: identification or clarification of a reference or a portion thereof, claim interpretation, proposed amendments, arguments of any applied references etc...)

Applicant agreed that the examiner's amendment, authorized by the above attorney, would place the application in condition for allowance.

Examiner Amendment

An examiner's amendment to the record appears below. Should the changes and/or additions be unacceptable to applicant, an amendment may be filed as provided by 37 CFR 1.312. To ensure consideration of such an amendment, it MUST be submitted no later than the payment of the issue fee.

Authorization for this examiner's amendment was given in a telephone interview with Ashok Tankha (Reg. # 33,802) on 9/23/2014.

Claims Are Amended As Follows:

CLAIMS:

Claim 1 (Currently Amended): A method for transferring data from a Bluetooth enabled data capture device to a remote internet server via a Bluetooth enabled mobile device comprising:

performing at a Bluetooth enabled data capture device:

establishing a paired secure Bluetooth connection between the Bluetooth enabled data capture device and the Bluetooth enabled mobile device, wherein the paired secure Bluetooth connection uses cryptographic encryption key;

acquiring new data in the Bluetooth enabled data capture device, wherein new data is data acquired after the paired secure Bluetooth connection is established;

sending a message to the Bluetooth enabled mobile device, corresponding to supported data capability of the Bluetooth enabled data capture device;

Below are excerpts from three different sources (collectively, Exhibit 2012) about what *handshaking* and *handshake protocol* entails, as well as what a POSITA understands the handshake protocol to be. These sources are: (i) A 2007 Book:

“PC-BASED INSTRUMENTATION, CONCEPTS AND PRACTICE, By N. MATHIVANAN” (*Page 383*); (ii) Wikipedia page about handshake protocol; and (iii) (2002) Microsoft Computer Dictionary Fifth Edition (page 245). *See* Gajendragad [Exh. 2001] at ¶ 202.

First, the 2007 book clearly shows what “handshake protocol” is (as well as how it would have been understood by a POSITA), and it is being used to exchange supported data “capabilities of a device.” (EXHIBIT 2012); *see*

Gajendragad [Exh. 2001] at ¶ 203; *see also*

www.google.com/books/edition/PC_BASED_INSTRUMENTATION/OB65rMNQDo8C?hl=en&gbpv=

Table 8.2 GPIB capabilities, their symbols and functional descriptions

Interface capability Symbol Function		Description
T	Talker	Capability of a device to talk.
TE	Extended Talker	Capability of a talker to extended addressing.
L	Listener	Capability of a device to listen.
LE	Extended Listener	Capability of a listener to extended addressing.
SH	Source Handshake	Capability of a device to send command or data byte (multiline message) over GPIB using the three-wire handshake protocol.
AH	Acceptor Handshake	Capability of a device to receive command or data bytes (multiline message) from GPIB using the three-wire handshake protocol.
RL	Remote/Local	Capability of the device to switch between response to its front panel controls and response to commands sent over GPIB.
SR	Service Request	Capability of a device to assert service request to obtain service from controller.
PP	Parallel Poll	Capability of device to indicate that it needs service when the controller requests the status.
DC	Device Clear	Capability of a device to be reset.
DT	Device Trigger	Capability of a device to be triggered by a GPIB command to perform a device dependent function.
C	Controller	Capability of a device to send addresses, universal commands and addressed commands to other devices on the bus. It also includes the capability of the device to conduct polling to determine a device requesting a service.
E	Drivers	It describes the type of electrical bus drivers used in the device.

The IEEE-488 further defines subset of the above capabilities by a number appended to the capability symbol. The standard recommends that the GPIB capabilities of a device be symbolically marked near the GPIB connector on each device to indicate the interface functions the device supports. For example, the symbolic code 'SH1, LH1, T6, L4, SR1, RL1, PP0, DC1, DT0, C0, E2' indicates that the device has the following GPIB capabilities.

- (a) SH1 : Capability to transmit multiline messages across GPIB.
- (b) LH1 : Capability to receive multiline messages from GPIB.
- (c) T6 : Capability to become a talker when talk address with ATN asserted is sent by controller on the bus. Capability to respond to serial poll. The device ceases to be a talker when talk address of another device with ATN asserted is sent by the controller on the bus. The device does not have the talk-only capability.
- (d) L4 : Capability to become a listener when a listen address with ATN asserted is sent by controller on the bus. The device does not have the listen-only capability.
- (e) SR1 : Capability to assert SRQ line when the device requires a service.
- (f) RL1 : Capability to respond to both Go To Local (GTL) and Local Lock Out (LLO) command messages.
- (g) PP0 : Does not have the capability to parallel poll.
- (h) DC1 : Capability to respond to Device Clear (DCL) and Selected Device Clear (SDC) command messages.

Second, Wikipedia states: “Handshaking can negotiate parameters that are acceptable to equipment and systems at both ends of the communication channel, including **information [data] transfer rate**, coding alphabet, parity, interrupt procedure, and other protocol or **hardware features**.” See Gajendragad [Exh. 2001] at ¶ 204; *see also* [en.wikipedia.org/wiki/Handshake_\(computing\)](http://en.wikipedia.org/wiki/Handshake_(computing)).

In computing, a **handshake** is a signal between two devices or programs, used to, e.g., authenticate, coordinate. An example is the handshaking between a [hypervisor](#) and an application in a [guest virtual machine](#).

In [telecommunications](#), a **handshake** is an automated process of negotiation between two participants (example "[Alice and Bob](#)") through the exchange of information that establishes the protocols of a communication link at the start of the communication, before full communication begins.^[1] The handshaking process usually takes place in order to establish rules for communication when a [computer](#) attempts to communicate with another device. Signals are usually exchanged between two devices to establish a communication link. For example, when a computer communicates with another device such as a [modem](#), the two devices will signal each other that they are switched on and ready to work, as well as to agree to which protocols are being used.^[2]

Handshaking can negotiate parameters that are acceptable to equipment and systems at both ends of the communication channel, including [information transfer rate](#), coding [alphabet](#), [parity](#), [interrupt](#) procedure, and other [protocol](#) or hardware features. Handshaking is a technique of communication between two entities. However, within TCP/IP RFCs, the term "handshake" is most commonly used to reference the TCP three-way handshake. For example, the term "handshake" is not present in RFCs covering FTP or SMTP. One exception is Transport Layer Security, TLS, setup, FTP RFC 4217. In place of the term "handshake", FTP RFC 3659 substitutes the term "conversation" for the passing of commands.^{[3][4][5]}

Third, the Microsoft Computer Dictionary Fifth Edition (page 245) defines “handshake” as: “**A series of signals acknowledging** that communication or the **transfer of information.**” See Gajendragad [Exh. 2001] at ¶ 205.

handshake *n.* A series of signals acknowledging that communication or the transfer of information can take place between computers or other devices. A hardware handshake is an exchange of signals over specific wires (other than the data wires) in which each device indicates its readiness to send or receive data. A software handshake consists of signals transmitted over the same wires used to transfer data, as in modem-to-modem communications over telephone lines.

245

There are various handshake protocols depending on the application. The most common are the “three-way handshake” or “4-way handshake” protocol that

are used to negotiate data capabilities, security parameters and more. TCP/IP communication between devices uses handshake protocol before data transfer, SSL/TLS uses handshake protocol for negotiating security parameters, wireless protocol like WPA/WPA2 use four-way handshake for negotiating authentication parameters and other things. *See* Gajendragad [Exh. 2001] at ¶ 206.

Below is another example of SSL/TLS uses handshake protocol for negotiating security parameters from the 2006 Edition of the Information Security Management Handbook. *See* Gajendragad [Exh. 2001] at ¶ 207; *see also* www.google.com/books/edition/Information_Security_Management_Handbook/5sIJ4Ho5yLUC?hl=en&gbpv=1&kptab=getbook

Understanding the TLS Handshake Protocol

The TLS Handshake Protocol allows two peers to agree upon security parameters for the TLS Record layer, authenticate, initiate those negotiated security parameters, and report errors to each other.

During the session negotiation phase, the Handshake Protocol on each peer negotiates the security parameters in Table 6.1.

Once the session is initiated, however, the application can request a change in the cryptographic elements of the connection. The change is handled through the “change cipher spec protocol,” which sends a message to the peer requesting a change to the cipher properties. The change itself is encrypted with the current cipher values to ensure the request and associated information cannot be deciphered if intercepted.

How the Protocol Works

For TLS to properly protect a session using cryptographic features, it must negotiate the cryptographic parameters. Figure 6.2 illustrates establishing the session.

Upon initiating a TLS connection, the two nodes must establish a “handshake” and negotiate the session parameters. These parameters include the cryptographic values, optional authentication, and generated shared secrets.

Dissecting the Handshake Protocol

When the client contacts the server to establish a connection, the client sends a client hello message to the server. The server must respond with a server hello message or the connection fails. This is extremely important, as the hello messages provide the security capabilities of the two nodes.

Specifically, the hello message provides the following security capabilities to the other node:

- TLS protocol version
- Session ID
- Available cipher suite
- Compression method

As mentioned, both nodes compute a random value *I* that is also exchanged in the hello message.

Exchanging the keys can involve up to four discrete messages. The server first sends its certificate, provided the server is to be authenticated. If the certificate is only for signing, the server then sends its public key to the client. The server then sends a “server done” message, indicating that it is waiting for information from the client.

© 2006 by Taylor & Francis Group, LLC

Copyrighted material

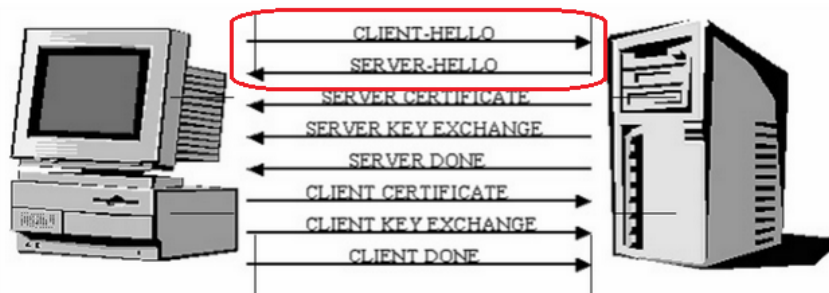


FIGURE 6.3 Handshake exchange.

A POSITA in 2007 clearly knew that the handshake protocol is to negotiate parameters such as data capabilities. As such, the specification’s disclosure of “implementing a Handshake Protocol” tells that messages are sent and received for data and other capabilities. The disclosure supports the claim limitations for a POSITA about sending messages to negotiate “supported data capability” for claim 1 and 12. *See* Gajendragad [Exh. 2001] at ¶ 208.

Next, Petitioner complains about support for “timestamp in the data Object” for Claim 15, which recites: “wherein new data object comprises of new data, data associated with new data and timestamp, and wherein the timestamp corresponds to a time when new data was acquired.” *See* Gajendragad [Exh. 2001] at ¶ 209.

Petitioner does NOT complain about Object being created with data and associated data. The Object is being transferred by using OBEX, as explained in the specification: “The digital data capture device 201 then automatically **transfers** 106 the captured **data**, the **multimedia** content, and the **associated** files to the client application 203 on the mobile device 202 using one or a combination of file **transfer protocols**. The file transfer protocols may, for example, be one or a combination of BT profile protocols such as the **object exchange (OBEX) protocol**, the generic object exchange profile (GOEP) protocol, etc.” *See* ’030 Patent at 4:34-41. *See* Gajendragad [Exh. 2001] at ¶ 210.

Petitioner fails to realize that when the Object is created for OBEX transfer/exchange, “Time” is one of the headers in the OBEX object. A POSITA also knows that **OBEX has a timestamp header** that is filled when the data object or file object is created. *See* Gajendragad [Exh. 2001] at ¶ 211; *see also* Exhibit 2009 (IrDA - OBEX Version 1.3 Document - Page 73):

9.2 Generic File Object

9.2.1 Introduction

The purpose of this chapter is to outline a method for the exchange of a basic file object [blob] from one device to another. While the OBEX protocol provides many useful headers and capabilities for exchanging objects. This section is designed to provide the reader with an overview of the basic structure used when exchanging generic file data. The file object is assumed to be sent by a generic client to the receiving device's inbox. In a manner consistent to the way QuickBeam and IrXfer applications work. Based on these assumptions, the file exchange should follow the guidelines shown here for headers used and response codes expected.

9.2.2 Commonly Used Headers

- **Name:** *(Required)* This header is used to convey the full name of the object to exchange. The **Name** header should not contain any path information. If it is necessary to specify a path, it should be done with the **SETPATH** command. All information in this header is interpreted as the objects name.
- **Length:** *(Recommended)* This header is used to convey the size of the file object in bytes. This can be used for verification of storage space requirements. The sum of the bytes passed in **Body** headers should be this many bytes (but is not guaranteed).
- **Body/End-of-Body:** *(Required)* Body headers are used to transfer the file data itself. They are sent repeatedly as long as file data exists to be exchanged. The **End-of-Body** header is used to indicate to the receiving application that this is the last piece of the file data. Applications frequently close the file and consider the transfer complete upon reception of this header. The data contained in the **Body/End-of-Body** headers should be concatenated to form the file.
- **Time:** *(Recommended)* This header is used to exchange the time that the file was last modified. Current file transfer applications use both the simple 4-byte **Time** header and the ISO format. It is recommended that all applications use the more robust ISO **Time** header. For backward compatibility it should be noted that the "IrXfer" application provided by Microsoft only accepts the 4-byte **Time** header. Attempts to send the ISO header will result in the rejection of the operation.
- **Type:** *(Optional)* The MIME type of the file object. This is not often used but can be helpful in some cases to identify the application to associate with the file object.

9.2.3 Response Codes Commonly Used in File Exchange

The following is a list of the known OBEX Response codes that are used during File Object Exchange.

A POSITA further knows that when a File / Data object is created for OBEX **timestamp is one of the info in the header that is put in an object**. The OBEX transfer disclosure above supports the claim limitations for POSITA about **data object** comprises of new data, data associated and time. *See Gajendragad [Exh. 2001] at ¶ 212.*

Next, Petitioner complains about sending “encrypted data” over the “paired secure Bluetooth connection” for Claim 1 and 6 as purportedly lacking support. Patent Owner disagrees. *See Gajendragad [Exh. 2001] at ¶ 213.*

The specification is clear for the POSITA on the use of data encryption and secure pairing. *See Gajendragad [Exh. 2001] at ¶ 214.* For example, the specification of the '030 Patent teaches:

[4:09-21] “In order to initiate the **pairing** process, the BT communication device 201 *a* will send the BT communication device name of a predefined number of characters . . . On entering the passkey by the user of the mobile device 202, the **entered passkey is matched** with the passkey of the BT communication device 201 *a*. If a match is found, a **trusted pair** is automatically established”

[4:20-21] “If a match is found, a trusted pair is automatically established.”

[3:61-64] “A BT device that wants to communicate only with a **trusted device** can **cryptographically** authenticate the identity of another BT device.”

[10:54-56] “The program will use various security, **encryption** and compression **techniques** to enhance the overall user experience.”

A POSITA knows that **when passkeys are matched and if a *trusted (secure) pair* is formed**, then Bluetooth cryptographic *link keys* have been exchanged. Link keys are the cryptographic encryption keys. Those same link keys trusted (secure) pair are used to encrypt/decrypt the data to communicate with the secure *trusted (secure) device*. See Gajendragad [Exh. 2001] at ¶ 215.

The disclosures above support the claim limitations about secure trusted pairing, cryptographic, and data encryption. Patent Owner thus disagrees with

Petitioner that any limitation lacks adequate written description support. *See* Gajendragad [Exh. 2001] at ¶¶ 216-217.

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 13,728 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: IPR50048-0060IP1@fr.com

Back-Up Counsel

Kim H. Leung, Reg. No. 64,399
Steven Katz, Reg. No. 43,706
Rishi Gupta, Reg. No. 64,768
Ryan O'Connor, Reg. No. 60,254
Fish & Richardson P.C.
60 South Sixth Street
Minneapolis, MN 55402
Tel: 202-783-5070
Fax: 877-769-7945
IPR50048-0060IP1@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