IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent of:	Singh		
U.S. Patent No.:	**8,904,030**	**Attorney Docket No.:**	**50048-0060IP1**
Issue Date:	December 2, 2014		
Appl. Serial No.:	14/295,353		
Filing Date:	June 4, 2014		
U.S. Patent No.:	**9,900,766**	**Attorney Docket No.:**	**50048-0056IP1**
Issue Date:	February 20, 2018		
Appl. Serial No.:	15/640,594		
Filing Date:	July 3, 2017		
U.S. Patent No.:	**11,234,121**	**Attorney Docket No.:**	**50048-0055IP1**
Issue Date:	January 25, 2022		
Appl. Serial No.:	16/532,507		
Filing Date:	August 6, 2019		
U.S. Patent No.:	**11,659,381**	**Attorney Docket No.:**	**50048-0054IP1**
Issue Date:	May 23, 2023		
Appl. Serial No.:	17/542,373		
Filing Date:	December 4, 2021		
Title:	**AUTOMATIC MULTIMEDIA UPLOAD FOR PUBLISHING DATA AND MULTIMEDIA CONTENT**		

SECOND DECLARATION OF DAVID B. LETT

TTI-1048
TikTok v. Cellspin
IPR2024-00770

TABLE OF CONTENTS

I. QUALIFICATIONS AND BACKGROUND INFORMATION 5

II. RESPONSES TO STATEMENTS MADE BY MR. GAJENDRAGAD 6

A. A File Is an Object 6

B. The "Along With" Proposal Is Unnecessary and Irrelevant.....13

C. The "User Profile" Construction Is Unnecessary, Confusing, and Ambiguous 16

D. The Prior Art Combinations Provide a Bluetooth Connection Maintained on a Continuous Basis 18

E. Bluetooth renders a paired, authenticated connection obvious 21

F. Kahn teaches HTTP 32

1. Kahn literally teaches HTTP.....32

2. WAP / WMTP still requires HTTP through a gateway.....33

G. The Prior Art Combinations Describe Event Notifications.....34

H. Bluetooth Polling Satisfies the Claims 38

I. Signal to Notify Is Processed.....41

J. Mischaracterizes Kahn+Bluetooth+Feinberg Combination 41

K. The Term "Handshaking Protocol" In The Specification Does Not Show Possession Of The Concept Of Messaging Support Supported Data Capability 42

L. Timestamp.....49

M. Client application 50

III. Conclusion 52

EXHIBITS

EX. 2001	Gajendragad Declaration
EX. 2007	Wap MMS Arch
EX. 2009	Object Exchange Protocol Version 1.3
EX. 2012	Wikipedia article on handshaking
EX. 2016	BLUETOOTH SPECIFICATION Version 1.1
TTI-1001	’030 patent
TTI-1003	’766 patent
TTI-1005	’121 patent
TTI-1007	’381 patent
TTI-1009	Singh129
TTI-1010	Singh906
TTI-1011	Certified translation of Hiroishi
TTI-1014	Certified translation of Takahashi
TTI-1017	U.S. Pub. 2004/0004737 (“Kahn”)
TTI-1018	U.S. 8,385,950 (“Wagner”)
TTI-1019	Bluetooth™ Core Specification 2.1 + EDR (“Bluetooth”)
TTI-1021	U.S. 7,639,943 (“Kalajan”)
TTI-1022	U.S. Pub. 2006/0173985 (Moore)
TTI-1024	U.S. Pub. 2006/0139474 (Endsley)
TTI-1025	U.S. Pub. 2007/0073937 (Feinberg)
TTI-1030	Lett Declaration
TTI-1031	Archived Webpage Bluetooth™ Core Specification v2.1 + EDR (https://web.archive.org/web/20071011031607/http:/bluetooth.com/Bluetooth/Learn/Technology/Specifications/Default.htm)
TTI-1032	C. Bisdikian, “An overview of the Bluetooth wireless technology,” in IEEE Communications Magazine, vol. 39, no. 12, pp. 86-94, Dec. 2001, doi: 10.1109/35.968817. (“The Bluetooth Overview”)
TTI-1033	RESERVED
TTI-1034	Memorandum, Interim Procedure for Discretionary Denials in AIA Post-Grant Proceedings with Parallel District Court Litigation (USPTO June 21, 2022) (“Interim Guidance”)
TTI-1035	TikTok Stipulation letter regarding IPR grounds in District court litigation.

TTI-1036	CellSpin Preliminary Infringement Contentions in *CellSpin Soft, Inc. v. ByteDance, Ltd. et al.*, Case No. 2:23-cv-496 (E.D. Tex.)
TTI-1037	Consolidated Trial Practice Guide, https://www.uspto.gov/patents/ptab/trials/practice-guides
TTI-1038	Transcript of the Deposition of Pavan M. Gajendragad of March 21, 2025
TTI-1039	Transcript of the Deposition of Pavan M. Gajendragad of March 28, 2025
TTI-1040	Microsoft Corporation, "Media Transfer Protocol Enhanced", Revision 0.96 August 31, 2006
TTI-1041	Bluetooth Basic Imaging Profile v1.2.1 January 21, 2019
TTI-1042	NIST "Guide to Bluetooth Security", Special Publication 800-121 September, 2008, downloaded from the internet at https://www.govinfo.gov/content/pkg/GOVPUB-C13-PURL-LPS121099/pdf/GOVPUB-C13-PURL-LPS121099.pdf
TTI-1043	Claim Construction Order, CellSpin v. Fitbit, Case No. 17-cv-05928-YGR (April 14, 2021 N.D. Cal)
TTI-1044	Excerpts from the Prosecution History of U.S. Patent Application Serial No. 13/922,227 ("the '227 application")
TTI-1045	Merriam Webster definition for "along with," retrieved March 24, 2025
TTI-1046	U.S. Patent No. 7,120,148 to Batz et al., filed February 12, 2002, Issued October 10, 2006
TTI-1047	Order granting Stay Pending Inter Partes Review, CellSpin Soft, Inc. v. ByteDance, Ltd. et al., Case No. 2:23-cv-496, E.D. Tex., granted Jan. 26, 2025

I. QUALIFICATIONS AND BACKGROUND INFORMATION

1. My name is David B. Lett, and I have been retained on behalf of TikTok Inc. to offer technical opinions relating to U.S. Patent Numbers 8,904,030 ("the '030 patent"),[1] 9,900,766 ("the '766 patent"), 11,234,121 ("the '121 patent"), and 11,659,381 ("the '381 patent") (together, "the Challenged Patents") and prior art references relating to its subject matter. My experience with digital multimedia, wireless communications, and web application technologies provides me with an understanding of the subject matter described and claimed in the Challenged Patents. I submit this Declaration in support of Petitioner's Reply to the Patent Owner Response and to respond to statements made by Mr. Gajendragad in Exhibit 2001.

2. This declaration contains statements of my opinions formed to date, and the bases and rationale for these opinions. I may offer additional opinions based on further review of materials in this case, including opinions and/or testimony of other expert witnesses.

3. For my efforts in connection with the preparation of this declaration, I have been compensated at my usual and customary rate for this type of consulting

[1] For ease, I will use the '030 patent specification's Detailed Description and Drawings when referring to entries to the specifications. The 030 specification is representative of the specifications of the other Challenged Patents.

activity. My compensation is in no way contingent on the results of these or any other proceedings related to the ’030 patent, the ’121 patent, the ’381 patent, and the ’766 patent.

II. RESPONSES TO STATEMENTS MADE BY MR. GAJENDRAGAD

A. A File Is an Object

4. I have been asked to comment on Mr. Gajendragad’s discussion of the term “object” as it appears in claims 6 and 15 of the ’030 patent and claims 1, 4, 9, and 13 of the ’766 patent. EX. 2001, ¶¶45-55. Mr. Gajendragad opines on the Patent Owner’s construction of “object” as: “An object can be a variable, a data structure, a function, or a method. Object contains value and is referenced by an identifier.” *Id*., ¶45. In this context, the definition proposed by the Patent Owner is incorrect, for the reasons I explain.

5. At the outset, I note that the term in question is not “object,” but rather “new data object” (or likewise “new-data object”). *E.g*., TTI-1001, claim 6. This point is important because Mr. Gajendragad’s discussion relies, in part, on sources that pertain to object-oriented programming, which is not a technology that is relevant to the claims at issue. *E.g*., EX. 2001, ¶¶50-52 (citing EX. 2021 Wikipedia - Object (computer science) and Microsoft Computer Dictionary for object-oriented programming entries). The claims at issue here do not implicate object-oriented programming, so referring to these sources to understand the term

"new data object" and its variants is incorrect. For example, the "new data object" of the claims is information being transferred from one device to another. It would therefore not be a "method" as the Wikipedia object-oriented programming definition suggests.

6. In the context of packaging data into objects for transmission, an "object" is a generic identifiable collection of data, and a file is an example of an object. Thus, if data is collected into a file, the file would be an exemplary object corresponding to that data. An object can, of course, be other collections of data as well. It could be an identifiable stream of data or some other identifiable collection of data. In the '030 patent claims 6 and 15, for example, after new data is acquired, a new data object is created that is defined to include 1) new data, 2) data associated with new data, and 3) a timestamp. *E.g.*, '030 patent, claims 6 (where the time-stamp attribute is a data signal notification) & 15. Likewise, in the '766 patent, the new-data object includes 1) signal to notify, 2) the acquired new data, and associated new-data. TTI-1003, claims 1, 4, 9, & 13. The claims provide details of what must be within the specific data "objects" of the claims.

7. The patent specifications do not refer to "new data objects." And, although the patent specifications use the term "object," that term is used in different contexts and context matters. For example, when the term "object" is used in contexts other than data transfer, the usage does not help a reader

understand the term "new data object" in the claims. *See, e.g.*, TTI-1001, 10:56-57 (software programs may be stored on or in one or more mediums as an "object" code, which means binary code generated from source code by a compiler as an intermediate step towards an executable program), 11:7-16 (database formats include "object-based models"). The claims are not referring to executable code or database formats.

8. The specifications do refer to a relevant type of "object." For example, the specification discusses transferring files between the data capture device and the cellular phone, including using specific "file transfer protocols," including OBject EXchange (OBEX). TTI-1001, 4:38-5:3. The general purpose of OBEX was to provide a protocol to support sending "an arbitrary 'thing,' or data object, from one device to another." EX. 2009, 8. OBEX thus provides a particular format for the exchange of data so that anyone complying with the format will be able to transmit and receive data. OBEX is consistent with the broad generic concept of "object." An OBEX object is one kind of object. A Windows file is another kind of object. An Android file is another kind of object. Etc. There are many types of objects. OBEX provides a common format so that one type of object (an Android file for example) can be sent to a computer expecting another type of object (for example, a Windows file). All three are objects and OBEX simply provides a common format for interoperability and

reliability. EX. 2009, 9, 11. Nothing in the OBEX specification suggests that files are not objects. OBEX defines a specific common format whereby one type of object (a file) is converted to an OBEX format (another type of object) for ease of transfer.

9. OBEX makes no such distinction between an object and a file. Quite the opposite: OBEX states quite plainly that the term "object" carries no "lofty technical meaning" and "is designed to transfer flexibly defined 'objects'; for example files, diagnostic information, electronic business cards, bank account balances, electrocardiogram strips, or itemized receipts at the grocery store." EX. 2009, 8.

> PCs, pagers, PDAs, phones, printers, cameras, auto-tellers, information kiosks, calculators, data collection devices, watches, home electronics, industrial machinery, medical instruments, automobiles, and office equipment are all candidates for using OBEX. To support this wide variety of platforms, OBEX is designed to transfer flexibly defined "objects"; for example, files, diagnostic information, electronic business cards, bank account balances, electrocardiogram strips, or itemized receipts at the grocery store. "Object" has no lofty technical meaning here; it is intended to convey flexibility in what information can be transferred. OBEX can also be used for Command and Control functions - directives to TVs, VCRs, overhead projectors, computers, and machinery. Finally, OBEX can be used to perform complex tasks such as database transactions and synchronization.

EX. 2009 at 8 (annotated)

The term "object" conveys "flexibly in what information may be transmitted"; and a file is one type of information that may be transmitted using OBEX.

10. The specifications of the patents provide examples for transferring files, including the use of a "file event listener" that is used to detect the "creation of a new file." *Id*. Thus, the specification does not distinguish between a file and

an object. Instead, the specification would lead a POSITA to understand that the claimed "new-data object" would include a "file" created by the data capture device that has "captured data, the multimedia content, and files associated with the captured data and the multimedia content." *See, e.g.*, *id.*, 4:38-5:17.

11. I note that Mr. Gajendragad refers to MTP as a non-object-based method. EX. 2001, ¶47. MTP, however, describes the MTP Object Model that only further demonstrates the genericness of the term "object." TTI-1040, 18. In that model, which is used to transfer media, the MTP Object Model provides a definition of "object," which includes a "file." *Id*.

1.3 MTP Object Model

The term *media* in *Media Transfer Protocol* is used to identify any binary data and is not restricted to audio/video formats to which it is commonly applied. Some examples of non-audio/video objects include contacts, programs, scheduled events, and text files.

Media objects are required to be represented as atomic binary objects during transfer, but are not required to be stored in the same format or structure on the device. Objects may be created on-demand, as long as they are accurately represented during content enumeration.

MTP objects consist of not only the binary content of the file, but also the descriptive metadata and references. MTP is designed such that objects can be recognized through the mechanisms that are provided in the protocol without requiring an understanding of the binary format of the file itself.

In MTP, an *object* is the combination of the binary file, its descriptive metadata and any intra-object references.

TTI-1040, at 18 (annotated).

12. I am aware that Mr. Gajendragad cites to a figure in the Bluetooth specification that he asserts suggests that a "file" is not an "object." Ex-2001, ¶48.

He is misinterpreting the figure. Ex-2001, ¶54.

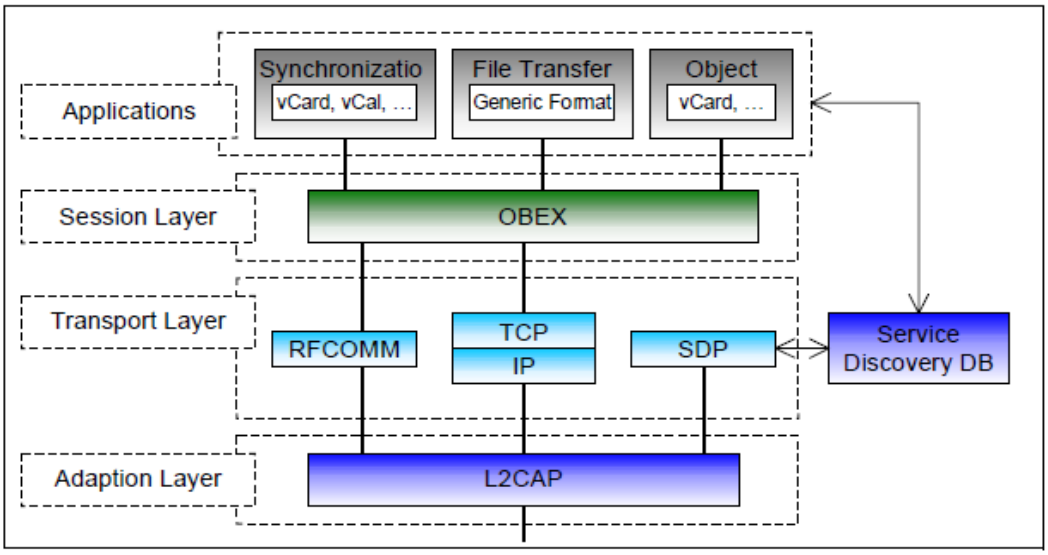


Figure 1.1: Part of Bluetooth Protocol Hierarchy

In the Bluetooth system, the purpose of the OBEX protocol is to enable the exchange of data objects. The typical example could be an object push of business cards to someone else. A more complex example is synchronizing calendars on multiple devices using OBEX. Also, the File Transfer applications can be implemented using OBEX. For the Object Push and Synchronization applications, content formats can be the vCard [4], vCalendar [5], vMessage [6], and vNotes [6] formats. The vCard, vCalendar, vMessage, and vNotes describe the formats for the electronic business card, the electronic calendaring and scheduling, the electronic message and mails, and the electronic notes, respectively.

EX. 2016 at 429 (annotated)

Nothing in this figure suggests that a "file" is not an "object." What this figure shows is that three applications are supported by OBEX: Synchronization, File Transfer, and Object Push. *E.g.*, EX. 2016 at 429. "File Transfer" and "Object Push" are two different applications that are used for different purposes, as described in the snippet above. *Id*. A file is an example of a typically large object that requires different handling for transfer than, for example, a very small object

such as a "vCard" or a "vMessage." *Id*. Moreover, OBEX describes Object Push as "a basic form of file transfer" accomplished using "OBEX **PUT** operation." EX. 2009 at 44, 50 (§7.1 – Simple Put – file/note/ecard transfer); *see also* EX. 2016 at 433 (§2.2.3 Put Operation for pushing an object).

13. The OBEX evidence is contrary to CellSpin's argument that the term "object" supposedly excludes files. Moreover, PO's argument runs contrary to its proposed construction, which does not exclude files from being objects. A file is data within a specific structure recognized by a file system.

14. Takahashi discloses the "new data object" of the claims, even under CellSpin's erroneous construction of "object." Takahashi discloses that the picture (new data) is saved in a file (new data object) along with a filename (data about the new data) and a timestamp (data signal notification). TTI-1014, [0055], [0070]. This mapping applies to CellSpin's constructions because a file is a structure for organizing data, and it has value (encoding of the picture as the data or payload of the file structure) and an identifier (filename). *Id*.

15. Similarly, in Kahn, the image file includes an image, metadata, such as a timestamp in the image file and location ID, an image ID, and a timestamp (signal to notify). TTI-1017, [0057], [0058] (discussing header information), [0060]. This image file is an "object" even under CellSpin's proposed constructions because a file is a structure for organizing data, and it has value (the

picture and metadata) and an identifier (unique image identifier). Kahn, [0058]. Indeed, Kahn expressly describes that there are "a variety of techniques in which an attribute (e.g., location ID) maybe associated with or stored with an underlying **object** (capture image)." Kahn, [0059] (emphasis added)

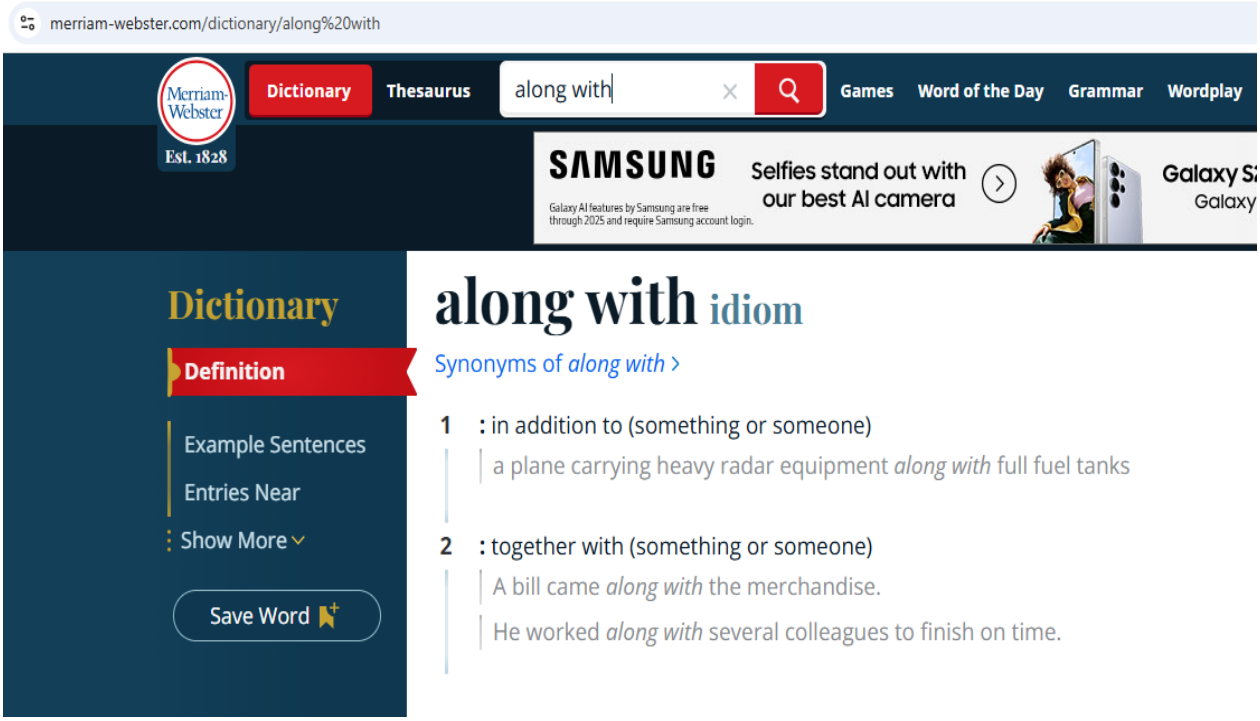
B. The "Along With" Proposal Is Unnecessary and Irrelevant

16. I have been asked to comment on Mr. Gajendragad's discussion of the term "along with" as it appears in claims 1, 6, 12, & 15 of the '030 patent, claims 1, 4, 9, and 13 of the '381 patent, and claims 1, 7, 12 of the '766 patent. EX. 2001, ¶¶37-44. Mr. Gajendragad opines on the Patent Owner's construction of "along with" to mean "in addition to." *Id*., ¶38. By way of support, Gajendragad reproduces a block quotation from U.S. Patent Application No. 11/901,802 ("the '802 application) and a portion of a Merriam Webster definition from the present day. EX. 2001, ¶¶38-42. Neither of these sources suggest a meaning other than the plain ordinary lay meaning of this non-technical term.

17. First, nothing in the '802 specification block quotation demonstrates redefinition of the phrase "along with" that would disavow any plain meaning. *Id*. The block quotation from the '802 application does not suggest any reason to depart from the term's ordinary meaning.

18. Mr. Gajendragad still cites to an online dictionary citation, but only shows one of the two definitions. EX. 2001, ¶41; TTI-1045. I have reproduced the

full Merriam Webster online dictionary entry below:



TTI-1045

19. Mr. Gajendragad appears to acknowledge that "along with" is not limited to "in addition to," and can extend to other definitions such as "together with," indicating that "along with" would have been interpreted broadly to have multiple meanings. EX. 2001, ¶43. Mr. Gajendragad does not specify why "in addition to"—what Patent Owner has advanced—is preferable to "together with."

20. Using its preferred definition, Patent Owner sets out to disqualify Takahashi prior art, and declares that, because Takahashi's user identifier is part of the filename, it is not transmitted "along with" ("in addition to") the new data. EX. 2001, ¶¶164-167. This argument is incorrect for multiple reasons.

21. First, Mr. Gajendragad misunderstands the mapping of the prior art to the claims. The "new data" is the data in the file and the user identifier in the filename is sent "along with" the new data. The filename is not "inside" the new data. It is metadata sent together with (or even in addition to) the data within the file. Mr. Gajendragad's discussion of "inside" does not respond to the proposed combination. EX. 2001, ¶¶166-167. A POSITA would have recognized that both the filename (which has the user identifier) and the new data would have been sent together, the filename being metadata (or data associated with new data).

22. The ordinary meaning of "along with" is broad and would not exclude data sent "inside of" other data. If I ship a bookcase, I can send books "along with" the bookcase by putting books in the bookcase before shipping. Or if I give someone a picture frame with a photo inside, I'm giving them the picture frame along with the photo. Nonetheless, the mapping in the Petition establishes invalidity under either construction because the User ID is not sent "inside" the new data. For example, in Takahashi, the image data is transmitted in addition to a filename that serves as a user identifier. *E.g.*, TTI-1012, [0066]. Likewise, Hiroishi sends the new data in addition to user authentication credential and user instructions. TTI-1030, ¶¶335, 337. Kahn sends the new data in addition to user information. TTI-1030, ¶589. A POSITA would have recognized that a filename is metadata that would have been included in the file that includes the image data.

Thus, a POSITA would have recognized that the filename was sent "along with" or "in addition to" the data. A POSITA would have recognized that image data is not sent "inside" a filename.

C. The "User Profile" Construction Is Unnecessary, Confusing, and Ambiguous

23. I have been asked to provide my opinion on the term "user profile." I understand the Patent Owner is suggesting a construction for the term "user profile" to mean "a collection of information associated with a user. It contains information such as their Username, Password, Email, Age, Gender, Country/Address." EX. 2001, ¶20. A POSITA would have understood the term "User Profile" to mean information associated with a user. A POSITA would not interpret the term any more narrowly than that. Thus, I do not believe any construction is necessary. The Patent Owner and Mr. Gajendragad provide exemplary data that may be in a User Profile. Some profiles may contain such information and other profiles would not. If the Patent Owner means to suggest that the User Profiles of the challenged claims requires the precise collection of information in the "such as" clause, then I disagree, and I see no justification for such a restrictive reading of the term. In fact, the use of "such as" in the construction contradicts any suggestion of a restrictive requirement.

24. My opinion is that the term "user profile" does not require a construction beyond its plain and ordinary meaning. I do not see the claims or the

common specification using the term "user profile" in a special way that would warrant a construction. A POSITA would have understood what a user profile was.

25. The specification does not include the term "user profile" at all. Indeed, this term only appears in the '802 application text. TTI-1010, [0034]. But even in the '802 application text, the term "user profile" is not described to be something that is outside of the plain and ordinary usage; nor is it described with any sort of specificity that would warrant the particular definition described by Mr. Gajendragad. EX. 2001, ¶¶56-66.

26. Mr. Gajendragad cites to articles about MySpace in support of his discussion of the meaning of User Profile. But, of course, a POSITA would not limit the general term "User Profile" to the specific profile content that MySpace happened to use. It is simply one out of thousands of User Profiles that a system designer may create. *Id*., (citing EX. 2017).

27. Moreover, if Patent Owner's constructions are adopted, it would not change my opinion that the prior art cited in the '381 patent Petition renders claims 1 and 6 unpatentable. I have identified prior art user profiles that are each a collection of information about a user. *E.g*., TTI-1017, [0098]. FIG. 6. Even under the Mr. Gajendragad's proposed construction, there is no requirement, for example, to include any particular field, such as gender, email, or address. These

follow the non-restrictive phrase "such as" in the unnecessary proposed construction.

D. The Prior Art Combinations Provide a Bluetooth Connection Maintained on a Continuous Basis

28. Mr. Gajendragad asserts that my analysis regarding the paired secure Bluetooth connection is incomplete because I did not discuss the construction of the phrase "after the paired secure Bluetooth connection is established" applied in the District Court litigation. There, an agreed construction was provided by which "after the paired secure Bluetooth connection is established" was agreed to mean "after the paired connection is established and maintained on a continuous basis."

29. In the context of the patents, I assume this construction simply clarifies that connection is maintained and used by some of the subsequent steps of the claims. That is what I assumed when I performed my analysis, and thus my analysis applies with equal force to the agreed construction. That is, a connection is established, and while the connection is maintained, additional steps are performed using the established connection.

30. For example, Hiroishi describes systems and techniques for a mobile phone to remotely control a digital camera using a Bluetooth connection. TTI-1011, Abstract, [0043], [0066]. A POSITA would have recognized or found obvious that Hiroishi establishes and then maintains the Bluetooth connection to allow all of the communication exchanges between the camera and the mobile

phone to facilitate remote control operation of the camera to take photographs, including transferring thumbnails and photographs from the camera to the mobile phone, transferring instructions, sending images to the webserver, etc. TTI-1011, [0044], [0077]-[0107] (describing processes illustrated in FIG. 5). Although not required under the claims or claim construction, Hiroishi also maintains the Bluetooth connection during the transmission of the photographs to the remote terminal. TTI-1011, [0044], [0107].

31. Likewise, in the Kahn-Bluetooth combinations (e.g., for the '766 patent claims), Kahn's camera and mobile phone are connected by a paired, secure Bluetooth connection, which is maintained continuously to allow the user to control the camera using the host device. TTI-1017, [0046]-[0059]. First, Kahn teaches that the camera device is preferably a wireless camera device that includes wireless connectivity and that it can be "persistently" connected to the host device. TTI-1017, [0055]. Thus, Kahn teaches (or at least suggests) that the camera is persistently linked to cell phone over the wireless link. *Id*. In the context of the Kahn-Bluetooth combinations, a POSITA would have understood that to mean that the paired Bluetooth connection between the camera and the host device was maintained on a continuous basis (persistently coveys this concept).

32. Additionally, a POSITA reading Kahn would have recognized the benefits of establishing the wireless link connection prior to using the camera for

capturing photographs and for uploading photographs to the image management server. For example, Kahn is concerned, in part, with recording "where" images are captured, in addition to "when." Kahn, [0008], [0011]-[0012], [0051] (describing that "location information is automatically captured for individual photographic images … to assist with organizing those images"). To accurately know where a photograph was taken, Kahn teaches using a GPS module embedded in the host device (e.g., cell phone) to automatically and immediately associate GPS location information with a photograph "upon capture." TTI-1017, [0051], [0055], [0058]. Taken together with Kahn's teaching that the camera is persistently connected to the host device through wireless connectivity, a POSITA would have recognized that the Kahn+Bluetooth combination teaches maintaining the paired Bluetooth connection.

33. Mr. Gajendragad states that "Application and systems would connect to Bluetooth then disconnect after the job was done." *E.g.*, EX. 2001, ¶91. Mr. Gajendragad does not provide support or corroboration for this statement. Also, Mr. Gajendragad does not specify what "job" he is talking about. Also, for accuracy, systems did not connect "to Bluetooth." They used Bluetooth to connect the devices to each other. But in any case, as I have discussed in connection with the specific combinations in the Petitions, a POSITA would have established the Bluetooth connection prior to captured data and would have maintained the

Bluetooth connection.

34. Mr. Gajendragad suggests that a POSITA would not use a paired Bluetooth connection and would not maintain a Bluetooth connection because of battery considerations due to pairing. EX. 2001, ¶82. The cited evidence does not support the conclusions. For example, Mr. Gajendragad suggests that the pairing process is battery draining. First, he does not quantify the battery drain or justify his conclusion that a skilled artisan would not use pairing as a result—a conclusion that contradicts the Bluetooth specification, which mandates a paired connection except for backward compatibility. Second, it is important to distinguish the process of pairing with the establishment of a paired connection using link keys. The process of pairing need only be done once for a pair of devices. And once paired, the allegedly battery intensive operation does not have to be repeated. Thus, a POSITA would certainly create a paired connection for the benefits it provides.

E. Bluetooth renders a paired, authenticated connection obvious

35. Mr. Gajendragad does not argue that the prior art combinations used in the four Challenged Patent petitions are lacking a description of pairing, authentication, and encryption. Nor could he. The Bluetooth specification clearly teaches pairing, authentication, and encryption. *E.g.*, TTI-1019, 1058 (describing secured simple pairing); EX. 2001, ¶¶79, 87-88 (citing TTI-1019 at 1273 & 1286

describing that exchanging link keys and encryption was the default). Mr. Gajendragad's argument is merely that a POSITA would not have found it obvious to employ pairing and authentication—allegedly because of the number of permutations of Bluetooth connection states. He is incorrect.

36. By way of explanation, Bluetooth 2.1+EDR introduces a pairing technical called Secure Simple Pairing. Secure Simple Pairing involves Authentication and Encryption, as shown below:

BLUETOOTH SPECIFICATION Version 2.1 + EDR [vol 2] page 888 of 906

Security Specification Bluetooth

7 SECURE SIMPLE PAIRING

The Secure Simple Pairing security functions and procedures are described in this section. In addition, a cryptographic analysis of each procedure is provided.

There are five phases of Secure Simple Pairing:

- Phase 1: Public key exchange
- Phase 2: Authentication Stage 1
- Phase 3: Authentication Stage 2
- Phase 4: Link key calculation
- Phase 5: LMP Authentication and Encryption

Phases 1, 3, 4 and 5 are the same for all protocols whereas phase 2 (Authentication Stage 1) is different depending on the protocol used. Distributed through these five phases are 13 steps.

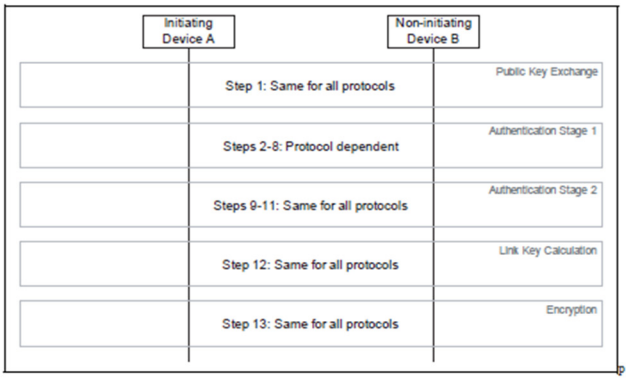


Figure 7.1: Simple Pairing Security Phases

TTI-1019, 1058

The Bluetooth Specification describes "Secure Simple Pairing security functions

and procedures" between connected devices. As part of Authentication, "a link key is computed from the derived shared key and the publicly exchanged data" and the "final phase [phase 5] in Simple Pairing is generating the encryption keys." TTI-1019, 1067. A link key is a "secret key that is known by two devices and is used in order to authenticate each device to the other." TTI-1019, 91.

37. After pairing is complete, a secure, authenticated link or connection is established. TTI-1019, 135. That is, once two devices have been paired, they may later connect automatically using the authentication and encryption protocols established during pairing, and without having to repeat the pairing process. *E.g.*, TTI-1019, 1270, FIG. 5.1, 1273, FIG. 5.4, *see also*, U.S. Pat. No. 8,260,324, 3:55-61 (describing that if a device is trusted, e.g., "it has already been paired" using Bluetooth, a wireless link may be automatically established to expedite link establishment). As explained by Bluetooth, part of the "Connection Establishment" (or Channel Establishment) procedure, if a Link Key exists and encryption is enabled (which they would be after implementing default parameters for Secure Simple Pairing), then the connection establishment procedure proceeds to the Connection Request. TTI-1019, 1273, FIG. 5.4.

38. The Bluetooth Specification describes one of the "Security Goals" of Secure Simple Pairing to be "protection against passive eavesdropping," among other things. TTI-1019, 131. To protect the user against passive eavesdropping, a

"strong link key coupled with a strong encryption algorithm is necessary." *Id*. For example, the attacker would have to solve "a hard problem in public key cryptography in order to derive the link key from the recorded information." *Id*. In my original Declaration, I described why it would have been obvious to a POSITA to pair the camera and the mobile phone, which include protection against passive eavesdropping, among other benefits. TTI-1030, ¶177, 524, 542-547.

39. The Bluetooth Specification teaches that the Just Works association model employed during Secure Simple Pairing is sufficient to guard against passive eavesdropping. *E.g*., TTI-1019, 131-135, 873, 1273, 1286. The Just Works association model uses an "unauthenticated link key," which is "a link key where the just works Secure Simple Pairing association model was used." *Id*., 1273. As shown below, using the Just Works association model (and unauthenticated link key) still results in a "<u>Secure</u>, <u>Authenticated</u>, Simple <u>Pairing</u>." *Id*., 135:

The following diagram shows Secure Simple Pairing from the point of view of the technology used for discovery and then the different association possibilities.

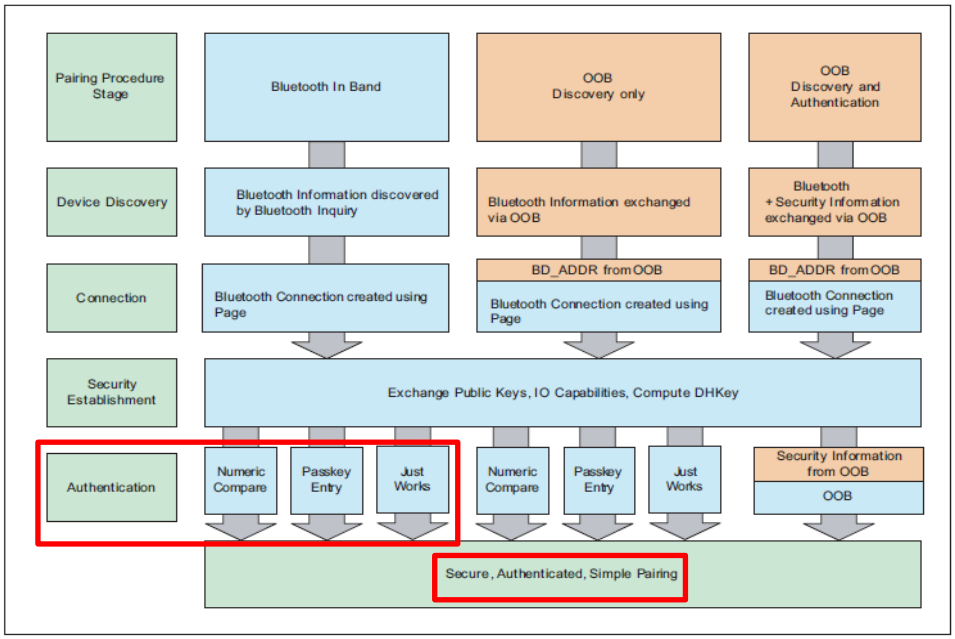


Figure 5.1: Secure Simple Pairing Association Models

TTI-1019, 135 (annotated)

Bluetooth 2.1 also defines creation of a secure connection as "A procedure of establishing a connection, including authentication and encryption." *Id.*, 90. As described in Bluetooth 2.1+EDR, through pairing, such as secure simple pairing, each device recognizes the other device as authenticated.

40. Regarding Mr. Gajendragad's specific statements pertaining to Security Mode 4 defaulting to an unauthenticated link key, Mr. Gajendragad confuses pairing and the associated Authentication stage with whether a link key itself is authenticated or unauthenticated. EX. 2001, ¶¶87-88 (general discussion), 117-121 ('030 patent), 252-255 ('766 patent), & 332-336 ('121 patent), 419-421

('381 patent – no mention of Security Mode 4 here); TTI-1019, 91 (defining a link key as "A secret key that is known by two devices and is used in order to authenticate each device to the other").

41. Mr. Gajendragad declares that Security Mode 4 defaults to requiring an unauthenticated link key as part of the Secure Simple Pairing authentication process. TTI-2001, 87. But this default setting does not help CellSpin . In fact, that the default setting uses a link key only reinforces the point that a POSITA would have found it obvious to establish a paired secured connection using a cryptographic encryption key (link key) when implementing Bluetooth in Hiroishi and Kahn. Indeed, without performing any additional steps laid out in Bluetooth, the Secure Simple Pairing procedure includes Authentication using a link key. *Id*., 131-134, 1273.

42. Also, Mr. Gajendragad's declaration admits that Security Mode 4 defaults to encryption. EX. 2001, ¶87 (admitting that Encryption is the default when implementing Secure Simple Pairing). The default setting confirms that a POSITA would have found it obvious to use encryption when implementing Bluetooth in Hiroishi and Kahn.

43. Additionally, Mr. Gajendragad is incorrect when he suggests that the default setting for Bluetooth is an unpaired connection. Gajendragad, ¶107-108. Mr. Gajendragad makes that statement without any corroborating evidence. By

contrast, Bluetooth 2.1+EDR does not support that statement. In fact, Bluetooth 2.1 + EDR devices **default to a paired, authenticated, encrypted connection**. The default behavior of using a paired connection is primarily dictated by Security Mode 4 (mandatory, "M," for all Bluetooth 2.1 + devices), which requires Secure Simple Pairing before any encrypted connection (the default) can be established. TTI-1019, 1269, 1273, 1276, This idea was recognized by security experts implementing Bluetooth in the Federal Government. *E.g.*, TTI-1042, 16 (National Institute for Science and Technology, Guide to Bluetooth Security, "NIST"). NIST is a Certified publication and is available at https://www.govinfo.gov/content/pkg/GOVPUB-C13-PURL-LPS121099/pdf/GOVPUB-C13-PURL-LPS121099.pdf ("NIST"). TTI-1042, 1. As part of the security protocol implementation recommendations, NIST explains that Security Mode 4 is "mandatory" for communication between 2.1 + EDR devices and uses Secure Simple Pairing, which is consistent with my description of Bluetooth 2.1+EDR above. TTI-1042, 7, 16, 18. NIST demonstrates that a POSITA implementing Bluetooth would have known that Security Mode 4 was required for Bluetooth 2.1+EDR and that Secure Simple Pairing was the default for Security Mode 4.

44. Mr. Gajendragad points to the ACL Link Establishment procedure in Bluetooth 2.1+EDR to show that pairing was optional. But Mr. Gajendragad's

interpretation of ACL Link Establishment procedure is incorrect. During ACL Link Establishment, only if the devices to be connected "do not have a common link key, a pairing procedure is used." TTI-1019, 866. "If a common link key exists between the devices, then pairing is not needed." *Id*., 867. In that sense, for connection establishment, pairing can be considered an optional step. Pairing only needs to occur if two devices are not already paired. If they are already paired, then connection re-establishment does not need to include pairing. Put differently, once the devices are paired, they may remained paired so that two devices do not need to undergo pairing again when establishing a connection. I also note that Bluetooth 2.1+EDR supports various actions for *after* a connection is established, one of which is to trigger simple pairing to implement security for specific services that device is accessing. TTI-1019, 871.

45. Mr. Gajendragad's statements that "two Bluetooth devices can have <u>bi-directional two-way</u> communication between them without ever pairing (even today)" is also uncorroborated by evidence.

46. Mr. Gajendragad also points to certain error codes that he argues allegedly prove that pairing was not required. *E.g*., EX. 2001, ¶111. Having already discussed that pairing was obvious, whether pairing was required misses the point. But that Bluetooth describes certain error codes associated with pairing is not evidence that pairing was optional. Bluetooth's description of the error code

"Pairing Note Allowed (0X18)" reveals that there are instances "when a device only allows pairing during a certain time window after some user input allows pairing." TTI-1019, 511. The "Simple Pairing Not Supported (0X37)" error code is for legacy devices that do not support the new feature of Simple Pairing. TTI-1019, 515. Secure Simple Pairing was introduced into the specification as a new feature in Bluetooth Core Specification 2.1 + EDR to simplify the pairing process and to improve security. TTI-1019, 154. Thus, it made sense to include error codes to indicate whether certain devices supported the new features.

47. Next, I will address Mr. Gajendragad's discussion of the Basic Imaging Profile ("BIP"). *E.g.*, EX. 2001, ¶90. Exhibit 2015 prior to institution was the 2003 DRAFT BIP specification. EX. 2015 is now the prosecution history for the '381 patent.

48. The DRAFT BIP specification does not support Mr. Gajendragad's conclusions about pairing. First, Mr. Gajendragad's statement that "BIP was not recommending using a paired Bluetooth connection" grossly misrepresents what the DRAFT BIP describes. EX. 2001, ¶90-92. First, DRAFT BIP states, "All devices must support pairing as defined in the [Generic Access Profile]." DRAFT BIP at 16. If BIP were recommending against pairing, as Mr. Gajendragad states, then it would not have required that devices support pairing. Second, DRAFT BIP states that "pairing can be performed as necessary and is left to the implementer's

discretion." *Id*. This statement is not recommending against pairing; instead, it is permissive as to how pairing can be implemented while still supporting the Basic Imaging Profile. From my previous discussions, we see that even with the new Secure Simple Pairing procedure in Bluetooth 2.1 + EDR, there were many implementation choices available to the implementor with varying degrees of security and levels of encryption. DRAFT BIP does not require any one specific type of pairing to be implemented.

49. Also, as I discussed before, pairing is not required as part of the connection establishment procedure when two devices are already paired. Thus, in the DRAFT BIP document referred to by Mr. Gajendragad, the capability of pairing was required, and if two devices were not already paired, then they could be paired based on these various implementation choices.

50. Mr. Gajendragad's statements that a POSITA would have followed the DRAFT BIP ignores the fact that BIP was not adopted into Bluetooth 2.1+EDR by 2007. Indeed, Bluetooth did not adopt BIP until 2010, three years after the release of Core 2.1+EDR (and after the patents' priority date), as shown below:

Revision History

Revision	Date	Comments
1.0 Final	2002-11-15	Final draft
1.0 Final_a	2003-07-25	Further changes after BQRB review
V11	2010-08-26	Adopted by the Bluetooth SIG Board of Directors
V12	2012-07-24	Adopted by the Bluetooth SIG Board of Directors
v1.2.1	2019-01-21	Adopted by the Bluetooth SIG Board of Directors

BIP v.1.2.1 at 2 (annotated to show that the first adoption of BIP into Bluetooth

occurred in 2010)

Thus, I disagree with Mr. Gajendragad's statement that a POSITA would have implemented BIP before 2007. In addition, Mr. Gajendragad does not supply evidence that a POSITA would have followed the draft versions of BIP prior to being adopted into Bluetooth. But even assuming that Mr. Gajendragad is correct, and that a POSITA would have followed the DRAFT BIP, this does not help CellSpin. As discussed already, the DRAFT BIP required that both devices support pairing. This is yet another example of Mr. Gajendragad contradicting himself. Likewise, as stated above, a POSITA would have recognized that for Bluetooth 2.1+EDR devices, Security Mode 4 was mandatory. TTI-1019, 1270, Figure 5.1, 1273, NIST, 16. Unsecure communications are only discussed in the Bluetooth 2.1 for backward compatibility. *E.g.*, TTI-1019, 1269-1287 (describing security for backwards compatibility). In the prior art combinations, the phone and camera would use Bluetooth 2.1, and they would therefore be paired and communicate via an authenticated and encrypted channel. TTI-1019, 1269-1287; TTI-1042, 16.

51. Likewise, Mr. Gajendragad's arguments are contradictory with respect to his conclusion that the "conditional statement" means that not all Bluetooth devices supported pairing:

119. 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).

120. This statement also means that NOT ALL Bluetooth devices supported pairing. (Conditional statement **when** both device).

EX. 2001, ¶¶119-120

What the cited passages from Bluetooth imply is that there are scenarios when both devices do not support "Secure Simple Pairing," which make sense because Secure Simple Pairing was a new feature in Bluetooth 2.1+EDR. Mr. Gajendragad's conclusion that this means that not all Bluetooth devices support pairing generally is another gross misunderstanding.

F. Kahn teaches HTTP

1. Kahn literally teaches HTTP

52. First, Mr. Gajendragad is reading Kahn too narrowly. Kahn does not "require using WMTP for media upload." EX. 2001, ¶172. Kahn expressly teaches using HTTP for communication between the mobile phone and the image management server. TTI-1017, [0056]. Further, Kahn broadly teaches that the upload process "occurs via wireless communication technique." *Id.*, [0057]. A POSITA reading Kahn as a whole would have known or found it obvious to use

HTTP to send images from the mobile phone to the remote server because doing so was common practice at that time. *See, e.g.*, TTI-1017, [0056]-[0057].

53. Contrary to Mr. Gajendragad' statements, HTTP was used at that time in mobile devices to upload images and other data to a web server. For example, Takahashi so teaches. *E.g.*, TTI-1014, [0005]-[0008], [0020], [0026]-[0028], FIG. 2, etc. I note that Mr. Gajendragad does not suggest that Takahashi's teaching of HTTP is somehow incorrect or inapplicable. Likewise, Kalajan characterized using HTTP to "upload images from the mobile phone to Internet-connected servers" as "conventional." TTI-1021, 5:49-56.

54. Kahn teaches that the host device (i.e., the cellular phone) "communicates with an online Web-based image management server 330" using, for example, "HTTP protocol." Kahn, [0056]. Kahn teaches that the Internet connection established using the HTTP protocol is for transferring images to the image management server for fulfillment. Thus, a POSITA would have recognized that, according to Kahn, the HTTP protocol is used to establish an internet connection between the mobile phone and the image management server for the purpose of transferring images.

2. WAP / WMTP still requires HTTP through a gateway

55. In addition, the WAP MMS Architecture Overview (EX. 2007) shows that an MMS client would still be configured to use HTTP to upload the received

new-data to the publishing website. EX. 2007, 15. Specifically, a POSITA would have known that in WAP, messages that "transit over HTTP from the WAP Gateway to the MMS Proxy-Relay," as shown below:

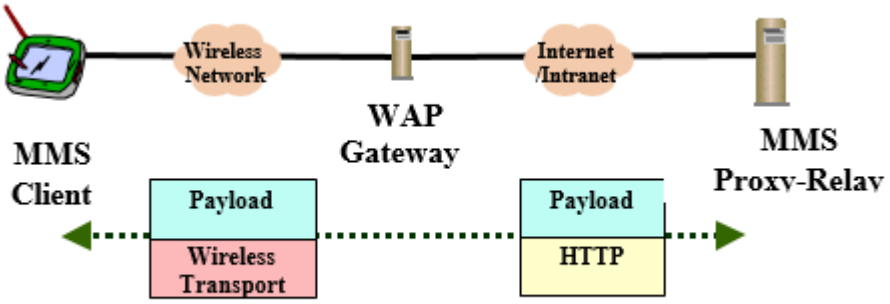


Figure 3 Logical Architecture of MMS Client to MMS Proxy-Relay Link

EX. 2007, 15.

"The network view also shows a payload that is carried by the wireless transport and HTTP." *Id*.

G. The Prior Art Combinations Describe Event Notifications

56. Mr. Gajendragad's assertion is incorrect that "Hiroishi or Kahn or Bluetooth does not support event notification for new data." EX. 2001, ¶¶128, 342, 424. In fact, the claims are literally met by the prior art combinations as explained in my first declaration, for example when I state "[i]n the Hiroishi-Takahashi-Bluetooth combination, there are a variety of signals that would perform the event notification function." TTI-1030, ¶190.

57. Mr. Gajendragad seems to misunderstand Hiroishi's disclosure of Hiroishi's thumbnails event notifications as a " simple request-response transaction

between the two devices that needs to be done for each new thumbnail." TTI-2001, ¶354 (emphasis added). What he fails to realize is that there is one request for thumbnails (enable event notification) followed by one or multiple thumbnails, each of which is an event notification of corresponding new data. In other words, receiving multiple thumbnails (event notifications for each of the new data full resolution images) by the mobile phone from the camera only requires one request for thumbnails (one enable event notification).

58. Each thumbnail (low resolution image) is an event notification corresponding to new data acquired. Hiroishi makes it clear that "the image data for displaying thumbnail**s** is transmitted to the mobile phone 40." TTI-1011, [0102] (plural emphasis added). Any number of thumbnails provide event notifications that there are that number of full resolution image**s** that can then be selected and then transmitted from the camera to the phone. For example, if there are 10 thumbnails (10 event notifications), there are up to 10 images that can be selected to cause up to 10 full resolution images (new data) to be transmitted from the camera to the mobile phone. That fact that in step 126 all images are deleted from the camera after transmission to the mobile phone proves that any new thumbnails that are sent to the mobile phone constitute notification of new data. TTI-1011, [0088], [0106], Fig. 5. Thus, any new images (new data) acquired by the camera result in thumbnail images (event notifications).

59. Stated simply in sequence, starting with no images or thumbnails stored on the camera: (1) any number of images are taken and stored by the camera; (2) the phone asks the camera once (enabling event notifications) if there is any new data (full resolution images), and listens for a response, (3) the camera sends the number of thumbnails (event notifications) that correspond to the number of new data full resolution images, (4) the mobile phone user selects by viewing the thumbnails which of the full images (new data) they would like to receive, (5) the mobile phone requests all the selected full resolution images (new data) to be transmitted from the camera to the phone; (5) the mobile phone requests the camera to delete all image data; (6) return to step (1).

60. The above interpretation conforms to how the ’030 patent, the ’121 patent, and the ’381 patent claims treat event notifications. For example, in the ’030 patent, claim 1, it recites the following capture device operations in order: (1) receiving a request to get new data, and then 2) receiving a request to enable event notifications for new data. Thus, Mr. Gajendragad’s distinction between request-driven and event driven architectures is inapplicable. In claim 1, the event notifications only occur *after* a request for data. TTI-1001, cl.1. Likewise, in claim 6, new data is acquired before the mobile device requests data signal notifications; based on the reception of such a request, the capture device will provide a notification.

61. As another example, claim 1 of the ’121 patent recites that the processor of the capture device is configured to send an event notification and the acquired new data, where in the event notification corresponds to the acquired new data and comprises sending a signal to the cellular phone. TTI-1004, cl.1. The claims recite that the cellular phone is configured to “listen to the event notification and receive the new-data.” *Id*. There is nothing in claim 1 of the ’121 patent that limits the event notification beyond being a “signal.” *Id*. Similar reasoning applies to claims 4 and 7.

62. Claims 1 and 7 of the ’381 patent recite similar language to claims 1, 4, and 7 of the ’121 patent for this feature. The ’381 patent claims 1 and 7 only require that the cellphone’s software be configured to listen for an event notification, where the event notification corresponds to the acquired new data and comprises involves the capture device sending a signal to the cellular phone. Thus, they only require that the software “be configured” to listen for the event notification, which are the thumbnails of Hiroishi. A POSITA would not import into claims 1 and 7 the unnecessary distinctions between event-driven and request-driven architectures suggested by Mr. Gajendragad.

63. Regarding the ’766 patent, Mr. Gajendragad argues the claim “requires that the first time the phone received the new data object, that is the first time the phone knows/or becomes aware of this new data, because the signal to

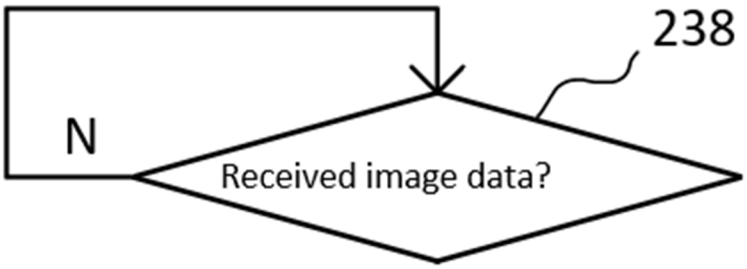
notify is embedded inside the new data Object." *Id*. This is incorrect. All that the claim "requires" is that the new-data object include the signal to notify, and that the signal to notify "corresponds to the acquiring of the new-data." Indeed, the claims suggests something different than what Mr. Gajendragad suggests. In claim 1, for example, the cellular phone is "configured to listen to the signal to notify and receive the new-data object." *E.g*., TTI-1003, cl.1. Nothing in this claim requires that the claimed signal to notify be the "first time the phone knows/or becomes aware" of something. The signal to notify of the claims, for example, may be used to notify the receiver that the data has arrived whereas an earlier signal alerted the phone that data would be forthcoming.

64. Mr. Gajendragad requires Kahn's camera to request GPS information from the host device. *E.g*., EX. 2001, ¶¶306-309. But, again, Mr. Gajendragad reads Kahn too narrowly. For example, Kahn teaches that location information can be automatically captured for individual photographic images. TTI-1017, [0051].

H. Bluetooth Polling Satisfies the Claims

65. Mr. Gajendragad is incorrect when he states that "Hiroishi and Khan and Bluetooth do NOT disclose Polling for new data". TTI-2001. ¶¶374, 452. The Hiroishi-Kahn-Endsley-Bluetooth and Hiroishi-Kahn-Bluetooth combinations clearly disclose polling as I discussed in my previous declaration. TTI-1030, ¶¶979-985, 1187-1188.

66. Hiroishi discloses that the mobile phone polls the camera in the flowchart loop for new data as represented when in "Step 238, [the phone] waits for the reception of the digital image data transmitted from the digital camera 50." TTI-1011, ¶107.



Step 238, TTI-1011, Fig. 6 (annotated)

67. A POSITA would understand that continually checking in step 238 for image data received until all of the selected images are received constitutes polling. Mr. Gajendragad alleges that "Hiroishi and Khan and Bluetooth do NOT disclose "Polling," citing the agreed construction, "["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.)]." TTI-2001, ¶¶130, 374. To the contrary, the polling that Hiroishi discloses in Step 238 meets the cited "polling" construction. The intervals for polling are predetermined at least by the loop time between Hiroishi's checks in step 238 for more image data received (until all of the selected images are received).

68. A POSITA would have recognized that the application layer programs cannot communicate directly with other application layer programs (or other lower layer programs) on another device using Bluetooth; the application layer relies on the lower layers to do so. Such is evident from the '121 patent, claim 9, where the capture device's processor is "configured to listen to a polling request from a single mobile client application." The '121 patent does not teach any application layer programs on the capture device. A POSITA would have understood that, in claim 9, the mobile client application would cause the lower layers of the mobile device to perform polling for new-data and that the lower layers of the capture device's protocol stack would perform listening. Thus, in '121 patent, a POSITA would have recognized that the application causes polling, but that polling occurs through other, lower layers of the stack.

69. Likewise, in the '381 patent, claim 12, a POSITA would have recognized that the mobile software application would have caused the cellular phone to perform polling, for the reasons I describe above. Also, in the '381 patent, polling includes **sending a request** to the Bluetooth enabled capture device to check if the acquired new-data is available for transfer. This limitation is met by Hiroishi's request for thumbnails.

70. Furthermore, polling is core to Bluetooth. Bluetooth operates using a master-slave model. The master controls the link and uses a polling mechanism to

communicate with slaves. This is described in the Bluetooth Baseband Core Specification. TTI-1019, 223-372. For example, see the "Polling in the active mode" section where a "master normally attempts to poll a slave … no less often than once every T_{poll} slots," where the predetermined interval of status checks is T_{poll}. TTI-1019, 332.

I. Signal to Notify Is Processed

71. Kahn teaches that the image manager has a database keyed for the timestamp (date/time) that was included in the image file. Kahn, [0098]. Thus, in Kahn, the timestamp must be "processed" so that the timestamp values are entered into the database and associated with the image. Additionally, a POSITA reading Kahn would have understood that the timestamp was processed so that "when" the image was taken could be associated with the image on the web server. Kahn, [0008].

J. Mischaracterizes Kahn+Bluetooth+Feinberg Combination

72. Mr. Gajendragad mischaracterizes the Kahn+Bluetooth+Feinberg combination by stating that "an extra" thumbnail is added to the new data object of Kahn. EX. 2001, ¶303. I stated simply that the thumbnail is not only sent first as a signal to notify but also for reasons given included with the uploaded photos, TTI-1030, 677-679. The combination shows that a POSITA would have found it obvious at that time to include a thumbnail version of the captured image in the file

header when transmitting the image file.

K. The Term "Handshaking Protocol" In The Specification Does Not Show Possession Of The Concept Of Messaging Support Supported Data Capability

73. Mr. Gajendragad argues that, despite no teaching in the specification of sending/receiving a message corresponding to supported data capability of the Bluetooth enabled data capture device, 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." EX. 2001, ¶199. This reliance on the Handshake Protocol briefly mentioned in the specification does not support Mr. Gajendragad's argument because 1) exchanging supported data capability is not inherent to handshake protocols, 2) the handshake protocol in the specification is not described as exchanging supported data capability, 3) the mention of handshake in the specification only applies to the "push mode," whereas claim 1 of the '030 patent does not claim push mode, 4) the handshake protocol described by the specification is an application-layer process, whereas the evidence cited by Mr. Gajendragad describes link layer processes, 5) Bluetooth does not use the term handshake, and 6) sending the supported data capability occurs after the devices have already been connected.

74. Mr. Gajendragad argues that the patent's reference to a "handshake protocol" necessarily discloses this feature. EX. 2001, ¶199. Mr. Gajendragad

(and Patent Owner) is essentially arguing that sending a message corresponding to supported data capability is inherent to a handshake protocol. However, Mr. Gajendragad is incorrect because handshake protocols do not always send supported data capability, as I understand that term. Mr. Gajendragad's own evidence proves that handshake protocols do not always include an exchange of supported data capability:

> A simple handshaking protocol might only involve the receiver sending a message meaning "I received your last message and I am ready for you to send me another one." A more complex handshaking protocol might allow the sender to ask the receiver if it is ready to receive or for the receiver to reply with a negative acknowledgement meaning "I did not receive your last message correctly, please resend it" (e.g., if the data was corrupted en route).[6]

EX. 2012, 1 (annotated)

As shown above, handshaking can include simply sending an acknowledgement message. EX. 2012, 1 (red underlining). Moreover, Mr. Gajendragad's evidence even states that more complex handshaking protocols do not require sending supported data capabilities. *Id*. (orange underlining). Mr. Gajendragad did not include this entry from EX. 2012 despite being on the same page as his snippet. EX. 2001, ¶204.

75. The specification has one mention of a "handshake protocol," and there are no details of the handshake protocol and no disclosure that it includes sending supported data capabilities. The actual language from the specification is reproduced below:

55 The transfer of the data, the multimedia content, and the associated files may also take place in a push mode of operation. In the push mode, the BT communication device **201***a* sends a signal to the client application **203** on creation of a new file. By implementation of a handshake protocol, the BT
60 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**. For some

TTI-1001, 57-62 (annotated)

More specifically, the handshake protocol described in the specification is not described as exchanging supported data capability. Instead, it is only described as configuring the BT communication device to "automatically transfer[] captured data, the multimedia content, and the associated files." TTI-1001, 57-62. There is no express mention here about sending supported data capability.

76. The brief mention of handshake in the specification only applies to the "push mode," where the Bluetooth communication device "automatically transfers captured data." TTI-1001, 57-62. Claim 1 of the '030 patent, however, does not claim the push mode. Specifically, in claim 1, after sending the supported data capability, the Bluetooth enabled data capture device **receives a request to get new data** from the Bluetooth enabled mobile device. TTI-1001, claim 1. This request for new data is in addition to receiving a request to enable event notifications. *Id*. Thus, at least in claim 1, the Bluetooth enabled capture device does not "automatically" transfer the data; instead, it transfers data after receiving a

request for new data. *Id*. Accordingly, the reference to the push-mode handshaking is inapplicable as support for sending a message corresponding to supported data capability.

77. The handshake protocol described by the specification is an application-layer process, whereas the evidence cited by Mr. Gajendragad describes link layer processes. Noteworthy is that Mr. Gajendragad never cites to Bluetooth for a message corresponding to supported data capability. And Bluetooth does not use the term handshake. Note that despite Bluetooth exchanging IO capabilities during the Secure Simple Pairing, this exchange is not the claimed supported data capability. For example, the IO capabilities exchange occurs *before* pairing, whereas in the claims, sending the supported data capability occurs *after* establishing the paired connection. TTI-1019, 135, 425, 1280. Recall that Secure Simple Pairing involves an exchange of IO capabilities to determine the whether the devices include a keyboard or display, so that the correct Association Model for Authentication is selected. TTI-1019, 135, 425, 1280.

78. In my original Declaration, I show that Bluetooth transmits supported data capability after pairing. *E.g.*, TTI-1030, 154, 181-183 (citing Bluetooth Core Specification V2.1, Section 4.3.4 (shown below)):

4.3.4 Supported features

The supported features may be requested at anytime following a successful baseband paging procedure by sending the LMP_features_req PDU. Upon reception of an LMP_features_req PDU, the receiving device shall return an LMP_features_res PDU.

The number of features bits required will in the future exceed the size of a single page of features. An extended features mask is therefore provided to allow support for more than 64 features. Support for the extended features mask is indicated by the presence of the appropriate bit in the LMP features mask. The LMP_features_req_ext and LMP_features_res_ext PDUs operate in precisely the same way as the LMP_features_req and LMP_features_res PDUs except that they allow the various pages of the extended features mask to be requested. The LMP_features_req_ext may be sent at any time following the exchange of the LMP_features_req and LMP_features_rsp PDUs.

The LMP_features_req_ext PDU contains a feature page index that specifies which page is requested and the contents of that page for the requesting device. Pages are numbered from 0-255 with page 0 corresponding to the normal features mask. Each page consists of 64 bits. If a device does not support any page number it shall return a mask with every bit set to 0. It also contains the maximum features page number containing any non-zero bit for this device. The recipient of an LMP_features_req_ext PDU shall respond with an LMP_features_res_ext PDU containing the same page number and the appropriate features page along with its own maximum features page number.

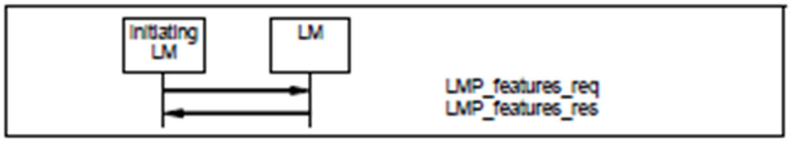
If the extended features request is not supported then all bits in all extended features pages for that device shall be assumed to be zero.

BLUETOOTH SPECIFICATION Version 2.1 + EDR [vol 2] page 272 of 906

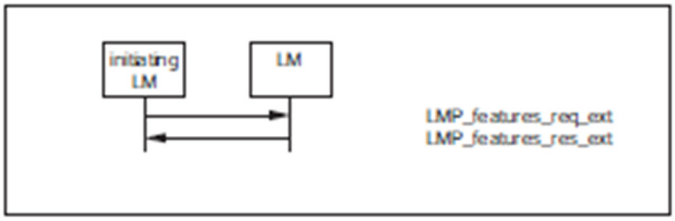
Link Manager Protocol

M/O	PDU	Contents
M	LMP_features_req	features
M	LMP_features_res	features
O(63)	LMP_features_req_ext	features page max supported page extended features
O(63)	LMP_features_res_ext	features page max supported page extended features

Table 4.22: PDUs used for features request.



Sequence 68: Request for supported features.



Sequence 69: Request for extended features.

E.g., Bluetooth Core Specification V2.1, 441-442.

79. As I explained in my original Declaration, examples of supported data capability that are exchanged by the Bluetooth devices are summarized in Section 3.2 of the Bluetooth specification. TI-1019, 384-389:

Feature	Definition
3 slot packets	This feature indicates whether the device supports the transmission and reception of both DM3 and DH3 packets for the transport of traffic on the ACL-U logical link.

5 slot packets	This feature indicates whether the device supports the transmission and reception of both DM5 and DH5 packets for the transport of traffic on the ACL-U logical link.
AFH capable slave	This feature indicates whether the device is able to support the Adaptive Frequency Hopping mechanism as a slave as defined in Baseband Specification, Section 2.3, on page 70 using the LMP sequences defined in Section 4.1.4 on page 227.
Encryption	This feature shall be supported. This feature indicates whether the device supports the encryption of packet contents using the sequence defined in Section 4.2.5 on page 248.
Enhanced Data Rate ACL 2 Mbps mode	This feature indicates whether the device supports the transmission and reception of 2-DH1 packets for the transport of traffic on the ACL-U logical link.
Enhanced Data Rate ACL 3 Mbps mode	This feature indicates whether the device supports the transmission and reception of 3-DH1 packets for the transport of traffic on the ACL-U logical link. This feature bit shall only be set if the “Enhanced Data Rate ACL 2 Mbps mode” feature bit is set.
3-slot Enhanced Data Rate ACL packets	This feature indicates whether the device supports the transmission and reception of three-slot Enhanced Data Rate packets on the ACLU logical link. This feature bit shall only be set if the “Enhanced Data Rate ACL 2 Mbps mode” feature bit is set.
5-slot Enhanced Data Rate ACL packets	This feature indicates whether the device supports the transmission and reception of 5-slot Enhanced Data Rate packets for the transport of traffic on the ACL-U logical link. This feature bit shall only be set if the “Enhanced Data Rate ACL 2 Mbps mode” feature bit is set.

Example Features Definition, Bluetooth Core Specification V2.1, 384-388 Cited by TTI-1030, 153-155.

80. As shown above, examples of features that are reported include number of slots per packet (3 slot packets or 5 slot packets), Adaptive Frequency Hopping (AFH) capability, Channel Quality Driven Data Rate, and Enhanced Data Rate (EDR). TTI-1019, 384-387 (explaining feature definitions).

81. In the claims, sending the supported data capability occurs after the devices have already been connected via a paired connection. For example, in claim 1 of the ’030 patent, sending a message to the Bluetooth enabled mobile device corresponding to supported data capability occurs only after the limitations that recites establishing a paired secure Bluetooth connection. TTI-1001, claim 1.

82. Thus, the handshake protocol being inapplicable to the supported data capability limitation, I reiterate that there is no support for sending / receiving a message corresponding to supported data capability in the specification. The Singh129 + Bluetooth combination and the Singh129 + Bluetooth + Takahashi combinations render the ’030 patent claims unpatentable.

L. Timestamp

83. Mr. Gajendragad points to OBEX for support for the new data object including a time-stamp attribute. EX. 2001, ¶211-212. This reasoning is flawed for two reasons: First, nothing from the cited portions of OBEX supports the time-

stamp being a data signal notification, as required by the claims. While it is true that a POSITA would have found it obvious from the prior art I discussed to use a time-stamp as a data signal notification, the cited portions of OBEX do not teach or suggest doing so. Second, the claims require that the time-stamp correspond to "a time when the new data was acquired." *E.g.*, TTI-1001, 13:20-21. None of these characteristics are met by the time header in OBEX.

84. In the cited portions of OBEX, the time header (which is only a recommendation) indicates the "time that the file was last modified." POR, 79 (citing EX. 2009, 73). When a data object "was last modified" says nothing about when that new data was acquired, or even whether new data was "acquired." Also, OBEX does not describe that the time entry in the header is used as a data signal notification.

M. Client application

85. Mr. Gajendragad's catalog of similarities between client application 202 from Singh906 and client application 203 from the present specification ignores their salient differences. *See*, POR, ¶490 (describing that the written description in both specifications is "almost" identical). Having reviewed Mr. Gajendragad's declaration, as well as CellSpin's PORs, I saw no response to the differences I identify. Thus, I stand by my original declaration and do not change my opinion that the client application 202 from Singh906 and client application

203 from the present specifications are different, and that CellSpin did not include any language in the present specifications that indicated that these two distinct embodiments could be combined.

III. CONCLUSION

I currently hold the opinions set expressed in this declaration. But my analysis may continue, and I may acquire additional information and/or attain supplemental insights that may result in added observations.

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

Dated: 04/04/2025 By: David B. Lett

David B. Lett