

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

In re Patent of: Singh et al.  
U.S. Patent No.: 8,898,260 Attorney Docket No.: 50048-0059IP1  
Issue Date: November 25, 2014  
App. Serial No.: 14/297,616  
Filing Date: June 5, 2014  
Title: AUTOMATIC MULTIMEDIA UPLOAD FOR  
PUBLISHING DATA AND MULTIMEDIA CONTENT

**Mail Stop Patent Board**

Patent Trial and Appeal Board  
U.S. Patent and Trademark Office  
P.O. Box 1450  
Alexandria, VA 22313-1450

**PETITION FOR *INTER PARTES* REVIEW OF UNITED STATES PATENT  
NO. 8,898,260 PURSUANT TO 35 U.S.C. §§ 311-319, 37 C.F.R. § 42**

## TABLE OF CONTENTS

|      |                                                                                                                            |    |
|------|----------------------------------------------------------------------------------------------------------------------------|----|
| I.   | REQUIREMENTS FOR IPR.....                                                                                                  | 1  |
| A.   | Standing.....                                                                                                              | 1  |
| B.   | Challenge and Relief Requested .....                                                                                       | 1  |
| II.  | THE '260 PATENT .....                                                                                                      | 2  |
| A.   | Brief Description.....                                                                                                     | 2  |
| B.   | Summary of the Prosecution History .....                                                                                   | 3  |
| C.   | Level of Ordinary Skill .....                                                                                              | 3  |
| D.   | Claim Construction .....                                                                                                   | 4  |
| III. | THE CHALLENGED CLAIMS ARE UNPATENTABLE .....                                                                               | 4  |
| A.   | GROUND 1— Aaltonen and Drescher render obvious claims 1-18.....                                                            | 4  |
| 1.   | Aaltonen.....                                                                                                              | 4  |
| 2.   | Drescher .....                                                                                                             | 5  |
| 3.   | A POSITA would have combined Aaltonen and Drescher (“Aaltonen<br>and Drescher”) .....                                      | 6  |
| 4.   | Analysis.....                                                                                                              | 9  |
| B.   | GROUND 2A—Takahashi and Na render obvious claims 1-26 GROUND<br>2B—Takahashi, Na, and Lind render obvious claims 1-26..... | 40 |
| 1.   | Takahashi .....                                                                                                            | 40 |
| 2.   | Na .....                                                                                                                   | 41 |
| 3.   | Lind .....                                                                                                                 | 42 |
| 4.   | A POSITA would have combined Takahashi and Na (“Takahashi-Na”) .....                                                       | 43 |
| 5.   | A POSITA would have combined Takahashi, Na, and Lind<br>(“Takahashi-Na-Lind”).....                                         | 45 |
| 6.   | Analysis.....                                                                                                              | 46 |
| IV.  | DISCRETIONARY CONSIDERATIONS.....                                                                                          | 94 |
| A.   | The <i>Fintiv</i> Factors Favor Institution—§314(a).....                                                                   | 94 |
| V.   | PAYMENT OF FEES – 37 C.F.R. §42.103 .....                                                                                  | 96 |
| VI.  | MANDATORY NOTICES UNDER 37 C.F.R § 42.8(a)(1) .....                                                                        | 96 |

|                                                                |    |
|----------------------------------------------------------------|----|
| A. Real Party-In-Interest Under 37 C.F.R. § 42.8(b)(1).....    | 96 |
| B. Related Matters Under 37 C.F.R. § 42.8(b)(2) .....          | 96 |
| C. Lead And Back-Up Counsel Under 37 C.F.R. § 42.8(b)(3) ..... | 97 |
| D. Service Information.....                                    | 97 |

**EXHIBITS**

- TTI-1001 U.S. Patent No. 8,898,260 to Singh et al. (“the ’260 patent”)
- TTI-1002 Excerpts from the Prosecution History of the ’260 patent
- TTI-1003 Declaration of Dr. Patrick Traynor
- TTI-1004 Complaint, *CellSpin Soft, Inc. v. ByteDance, Ltd. et al.*, Case No. 2:23-cv-496, E.D. Tex., filed Oct. 20, 2023
- TTI-1005 U.S. Patent Application Publication 2005/0209927 to Aaltonen et al. (“Aaltonen”)
- TTI-1006 U.S. Patent Application Publication 2006/0129631 to Na et al. (“Na”)
- TTI-1007 *HTTP: The Definitive Guide* (“Gourley”)
- TTI-1008 RESERVED
- TTI-1009 U.S. Patent Application Publication 2013/0315235 to Foo (“Foo”)
- TTI-1010 International Publication WO 2008/028508 to Drescher et al. (“Drescher”)
- TTI-1011 U.S. Patent Application Publication 2007/0174393 to Bosschaert et al. (“Bosschaert”)
- TTI-1012 Internet Archive of Mobic.com, “Ericsson Unveils the First Bluetooth Phone,” available at <https://web.archive.org/web/20001002110758/http://www.mobic.com:80/>, last accessed February 29, 2024.
- TTI-1013 U.S. Patent No. 8,862,757 to Singh et al. (“the ’757 patent”)
- TTI-1014 Excerpts from the Prosecution History of the ’757 patent
- TTI-1015 U.S. Patent No. 8,041,816 to Ozaki et al. (“Ozaki”)

- TTI-1016 Japanese Patent Application Publication No. 2005-303511 to Takahashi Susumu and Certified Translation (“Takahashi”)
- TTI-1017 International Publication No. WO 2005/109781 to Lind et al. (“Lind”)
- TTI-1018 U.S. Patent Application Publication No. 2008/0165982 to Hankey et al. (“Hankey”)
- TTI-1019 Hypertext Transfer Protocol -- HTTP/1.1, June 1999 (“RFC2616”)
- TTI-1020 U.S. Patent No. 7,181,207 to Chow et al. (“Chow”)
- TTI-1021 U.S. Patent No. 7,333,955 to Graves et al. (“Graves”)
- TTI-1022 U.S. Patent No. 8,095,463 to Hartmaier (“Hartmaier”)
- TTI-1023 WAP MMS Architecture Overview Version 25-April-2001 (“MMS\_1.0”)
- TTI-1024 U.S. Patent Application Publication No. 2009/0007196 to Ganesan (“Ganesan”)
- TTI-1025 Memorandum, Interim Procedure for Discretionary Denials in AIA Post-Grant Proceedings with Parallel District Court Litigation (USPTO June 21, 2022) (“Interim Guidance”)
- TTI-1026 TikTok Stipulation letter regarding IPR grounds in District court litigation.

## LISTING OF CHALLENGED CLAIMS

| Claim 1 |                                                                                                                                                                                                                                  |
|---------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1[pre]  | A method of segmented file transfer from a Bluetooth enabled mobile device to a web service, the method comprising:                                                                                                              |
| 1[a]    | providing a software module on the Bluetooth enabled mobile device, the software module, performing;                                                                                                                             |
| 1[b]    | determining size of a file to be transferred;                                                                                                                                                                                    |
| 1[c1]   | segmenting said file into data segments at an application layer level;                                                                                                                                                           |
| 1[c2]   | wherein data in said data segments comprises one or more of said file data and data associated with said file data;                                                                                                              |
| 1[d]    | applying a segment identifier to each of said data segments at the application layer level, wherein the segment identifier corresponds to the content range of said file data that is being transferred in a given data segment; |
| 1[e]    | applying a user identifier to each of said data segments for the destination web service at the application layer level, wherein said user identifier uniquely identifies a particular user of the web service;                  |
| 1[f]    | transferring a data segment along with said user identifier and said segment identifier to the web service,                                                                                                                      |
| 1[g]    | receiving a reply message from the web service, corresponding to successful completion of said data segment transfer;                                                                                                            |
| 1[h]    | repeating the step of data segment transfer and the step of receiving the reply message, until all the data segments have been transferred to the web service successfully,                                                      |
| 1[i]    | wherein if segmented file transfer is interrupted, the software module further performs:                                                                                                                                         |

|                |                                                                                                                                                                                                                                                                                               |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1[j]           | sending, a synchronization request to the web service from the application layer level, corresponding to asking the web service about the segmented file data the web service has successfully received, for said user identifier, after a network connection to the web service is restored; |
| 1[k]           | receiving, a synchronization reply back from the web service at the application layer level, corresponding to indicate segmented file data said web service has successfully received, for said user identifier; and                                                                          |
| 1[l]           | resuming, the segmented file transfer based on the synchronization reply received, for said user identifier.                                                                                                                                                                                  |
| 1[m]           | receiving, at the web service, all the data segments along with said user identifier and segment identifiers for said file transfer;                                                                                                                                                          |
| 1[n]           | storing, at the web service, the received file data segments and segment identifiers for said user identifier, and                                                                                                                                                                            |
| 1[o]           | making available, at the web service, the segmented data received from the Bluetooth enabled mobile device for public or private consumption over the internet.                                                                                                                               |
| <b>Claim 2</b> |                                                                                                                                                                                                                                                                                               |
| 2              | The method of claim 1, wherein the data associated with file data is at least one of name of the file, size of the file, and type of file.                                                                                                                                                    |
| <b>Claim 3</b> |                                                                                                                                                                                                                                                                                               |
| 3              | The method of claim 1, wherein said file data comprises one or more of audio data, video data, image data, text data, or digital data.                                                                                                                                                        |
| <b>Claim 4</b> |                                                                                                                                                                                                                                                                                               |
| 4              | The method of claim 1, wherein the user identifier comprises one or more elements from user information, and wherein the one or more elements of the user information comprise user-name, user-password, user-device-information, user settings, or user preferences.                         |

| Claim 5 |                                                                                                                                                                                                                                                      |
|---------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 5       | The method of claim 1, wherein the application layer is defined as one of following, the layer 7 in the OSI layer stack, or the application layer of the Internet protocol suite.                                                                    |
| Claim 6 |                                                                                                                                                                                                                                                      |
| 6[pre]  | A method of segmented file transfer from a Bluetooth enabled mobile device to a web service, the method comprising:                                                                                                                                  |
| 6[a]    | providing a software module on the Bluetooth enabled mobile device, the software module, performing;                                                                                                                                                 |
| 6[b]    | determining size of a file to be transferred;                                                                                                                                                                                                        |
| 6[c]    | segmenting said file into a first data segment and a second data segment at an application layer level;                                                                                                                                              |
| 6[d]    | wherein the second data segment comprises file data and the first data segment comprises data associated with file data, and wherein the data associated with file data is at least one of name of the file, size of the file, and type of the file; |
| 6[e1]   | applying a user identifier to said first data segment at the application layer level, wherein the user identifier uniquely identifies a particular user of a web service;                                                                            |
| 6[e2]   | transferring said first data segment along with said user identifier to the web service,                                                                                                                                                             |
| 6[e3]   | receiving a reply message from the web service, corresponding to successful completion of said first data segment transfer;                                                                                                                          |
| 6[f1]   | applying said user identifier to said second data segment at the application layer level;                                                                                                                                                            |
| 6[f2]   | transferring said second data segment along with said user identifier to the web service;                                                                                                                                                            |

|       |                                                                                                                                                                                                                                                         |
|-------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|       |                                                                                                                                                                                                                                                         |
| 6[f3] | receiving a reply message from the web service, corresponding to successful completion of said second data segment transfer,                                                                                                                            |
| 6[g]  | wherein if said second data segment transfer is interrupted, the software module further performs:                                                                                                                                                      |
| 6[h]  | sending, a synchronization request to the web service from the application layer level, for said user identifier, after a network connection to the web service is restored;                                                                            |
| 6[i]  | receiving, a synchronization reply back from the web service at the application layer level, corresponding to the successfully received data of said second data segment by said web service, for said user identifier; and                             |
| 6[j]  | resuming, the segmented file transfer by transferring remaining data of the second data segment based on the synchronization reply received, wherein resuming the segmented file transfer of remaining data comprises, the software module, performing: |
| 6[k]  | creating one or more data segments at the application layer level, wherein the one or more data segments comprise remaining data of the second data segment;                                                                                            |
| 6[l]  | applying a segment identifier to each of said one or more data segments at the application layer level, wherein the segment identifier corresponds to the content range of said file data that is being transferred in a given data segment;            |
| 6[m]  | applying said user identifier to each of said one or more data segments at the application layer level; and                                                                                                                                             |
| 6[n]  | transferring said one or more data segments along with said user identifier and said corresponding segment identifier to the web service,                                                                                                               |
| 6[o]  | receiving, at the web service, the data segments along with said user identifier and said segment identifiers for a given file transfer,                                                                                                                |

|                 |                                                                                                                                                                                                                                                                       |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 6[p]            | storing, at the web service, the received file data segments and segment identifiers for a user identifier; and                                                                                                                                                       |
| 6[q]            | making available, at the web service, the segmented data received from the Bluetooth enabled mobile device for public or private consumption over the internet,                                                                                                       |
| 6[r]            | wherein one or more portions of the data correspond to a particular user of the web services.                                                                                                                                                                         |
| <b>Claim 7</b>  |                                                                                                                                                                                                                                                                       |
| 7               | The method of claim 6, wherein said file data comprises one or more of audio data, video data, image data, text data, or digital data.                                                                                                                                |
| <b>Claim 8</b>  |                                                                                                                                                                                                                                                                       |
| 8               | The method of claim 6, wherein the user identifier comprises one or more elements from user information, and wherein the one or more elements of the user information comprise user-name, user-password, user-device-information, user settings, or user preferences. |
| <b>Claim 9</b>  |                                                                                                                                                                                                                                                                       |
| 9               | The method of claim 6, wherein the application layer is defined as one of following, the layer 7 in the OSI layer stack, or the application layer of the Internet protocol suite.                                                                                     |
| <b>Claim 10</b> |                                                                                                                                                                                                                                                                       |
| 10[pre]         | A method of segmented file transfer from a Bluetooth enabled mobile device to a web service, the method comprises:                                                                                                                                                    |
| 10[a]           | providing a software module on the Bluetooth enabled mobile device, the software module, performing;                                                                                                                                                                  |
| 10[b]           | determining size of a file to be transferred;                                                                                                                                                                                                                         |
| 10[c1]          | segmenting said file into data segments at an application layer level;                                                                                                                                                                                                |

|        |                                                                                                                                                                                                                                                                                               |
|--------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 10[c2] | wherein data in said data segments comprises one or more of said file data and data associated with said file data;                                                                                                                                                                           |
| 10[d]  | applying a segment identifier to each of said data segments at the application layer level, wherein the segment identifier corresponds to the content range of said file data that is being transferred in a given data segment;                                                              |
| 10[e]  | applying a user identifier to each of said data segments for the destination web service at the application layer level, wherein said user identifier uniquely identifies a particular user of the web service;                                                                               |
| 10[f]  | transferring a data segment along with said user identifier and said segment identifier to the web service,                                                                                                                                                                                   |
| 10[g]  | receiving a reply message from the web service, corresponding to successful completion of said data segment transfer;                                                                                                                                                                         |
| 10[h]  | repeating the step of data segment transfer and the step of receiving the reply message, until all the data segments have been transferred to the web service successfully,                                                                                                                   |
| 10[i]  | wherein if segmented file transfer is interrupted, the software module further performs:                                                                                                                                                                                                      |
| 10[j]  | sending, a synchronization request to the web service from the application layer level, corresponding to asking the web service about the segmented file data the web service has successfully received, for said user identifier, after a network connection to the web service is restored; |
| 10[k]  | receiving, a synchronization reply back from the web service at the application layer level, corresponding to indicate segmented file data said web service has successfully received, for said user identifier; and                                                                          |
| 10[l]  | resuming, the segmented file transfer based on the synchronization reply received, for said user identifier;                                                                                                                                                                                  |
| 10[m]  | receiving, at the web service, the data segments along with said user identifier and said segment identifiers for a given file transfer,                                                                                                                                                      |

|                 |                                                                                                                                                                                                                                                                        |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 10[n]           | reassembling the received file data segments automatically using segment identifiers and a user identifier, at the application layer level, at the web service; and                                                                                                    |
| 10[o]           | making available, at the web service, the reassembled data file, for public or private consumption over the internet.                                                                                                                                                  |
| <b>Claim 11</b> |                                                                                                                                                                                                                                                                        |
| 11              | The method of claim 10, wherein the data associated with file data is at least one of name of the file, size of the file, and type of file.                                                                                                                            |
| <b>Claim 12</b> |                                                                                                                                                                                                                                                                        |
| 12              | The method of claim 10, wherein said file data comprises one or more of audio data, video data, image data, text data, or digital data.                                                                                                                                |
| <b>Claim 13</b> |                                                                                                                                                                                                                                                                        |
| 13              | The method of claim 10, wherein the user identifier comprises one or more elements from user information, and wherein the one or more elements of the user information comprise user-name, user-password, user-device-information, user settings, or user preferences. |
| <b>Claim 14</b> |                                                                                                                                                                                                                                                                        |
| 14              | The method of claim 10, wherein the application layer is defined as one of following, the layer 7 in the OSI layer stack, or the application layer of the Internet protocol suite.                                                                                     |
| <b>Claim 15</b> |                                                                                                                                                                                                                                                                        |
| 15[pre]         | A method of segmented file transfer from a Bluetooth enabled mobile device to a web service, the method comprising:                                                                                                                                                    |
| 15[a]           | providing a software module on a Bluetooth enabled mobile device, the software module, performing;                                                                                                                                                                     |
| 15[b]           | determining size of file to be transferred;                                                                                                                                                                                                                            |

|        |                                                                                                                                                                                                                                                      |
|--------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 15[c]  | segmenting said file into a first data segment and a second data segment at an application layer level,                                                                                                                                              |
| 15[d]  | wherein the second data segment comprises file data and the first data segment comprises data associated with file data, and wherein the data associated with file data is at least one of name of the file, size of the file, and type of the file; |
| 15[e1] | applying a user identifier to said first data segment at the application layer level, wherein the user identifier uniquely identifies a particular user of a web service;                                                                            |
| 15[e2] | transferring said first data segment along with said user identifier to the web service,                                                                                                                                                             |
| 15[e3] | receiving a reply message from the web service, corresponding to successful completion of said first data segment transfer;                                                                                                                          |
| 15[f1] | applying said user identifier to said second data segment at the application layer level;                                                                                                                                                            |
| 15[f2] | transferring said second data segment along with said user identifier to the web service;                                                                                                                                                            |
| 15[f3] | receiving a reply message from the web service, corresponding to successful completion of said second data segment transfer,                                                                                                                         |
| 15[g]  | wherein if said second data segment transfer is interrupted, the software module further performs:                                                                                                                                                   |
| 15[h]  | sending, a synchronization request to the web service from the application layer level, for said user identifier, after a network connection to the web service is restored;                                                                         |
| 15[i]  | receiving, a synchronization reply back from the web service at the application layer level, corresponding to the successfully received data of said second data segment by said web service, for said user identifier; and                          |

|                 |                                                                                                                                                                                                                                                         |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 15[j]           | resuming, the segmented file transfer by transferring remaining data of the second data segment based on the synchronization reply received, wherein resuming the segmented file transfer of remaining data comprises, the software module, performing: |
| 15[k]           | creating one or more data segments at the application layer level, wherein the one or more data segments comprise remaining data of the second data segment;                                                                                            |
| 15[l]           | applying a segment identifier to each of said one or more data segments at the application layer level, wherein the segment identifier corresponds to the content range of said file data that is being transferred in a given data segment;            |
| 15[m]           | applying said user identifier to each of said one or more data segments at the application layer level; and                                                                                                                                             |
| 15[n]           | transferring said one or more data segments along with said user identifier and said corresponding segment identifier to the web service,                                                                                                               |
| 15[o]           | receiving, at the web service, the data segments along with said user identifier and said segment identifiers for a given file transfer,                                                                                                                |
| 15[p]           | reassembling the received file data segments automatically using segment identifiers and a user identifier, at the application layer level, at the web service; and                                                                                     |
| 15[q]           | making available, at the web service, the reassembled data file, for public or private consumption over the internet.                                                                                                                                   |
| <b>Claim 16</b> |                                                                                                                                                                                                                                                         |
| 16              | The method of claim 15, wherein said file data comprises one or more of audio data, video data, image data, text data, or digital data.                                                                                                                 |
| <b>Claim 17</b> |                                                                                                                                                                                                                                                         |
| 17              | The method of claim 15, wherein the user identifier comprises one or more elements from user information, and wherein the one or more                                                                                                                   |

|                 |                                                                                                                                                                                    |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                 | elements of the user information comprise user-name, user-password, user-device-information, user settings, or user preferences.                                                   |
| <b>Claim 18</b> |                                                                                                                                                                                    |
| 18              | The method of claim 15, wherein the application layer is defined as one of following, the layer 7 in the OSI layer stack, or the application layer of the Internet protocol suite. |

TikTok Inc. (“Petitioner”) petitions for *Inter Partes* Review of claims 1-18 (“the Challenged Claims”) of U.S. Patent No. 8,898,260 (“the ’260 patent”).

## **I. REQUIREMENTS FOR IPR**

### **A. Standing**

Petitioner certifies that the ’260 patent is available for IPR and Petitioner is not barred from requesting IPR.

### **B. Challenge and Relief Requested**

Petitioner requests IPR of the Challenged Claims on the grounds listed below. A declaration from Dr. Patrick Traynor (TTI-1003) supports this Petition.

| Ground | Claims | §103 Basis              |
|--------|--------|-------------------------|
| 1      | 1-18   | Aaltonen and Drescher   |
| 2A     | 1-18   | Takahashi and Na        |
| 2B     | 1-18   | Takahashi, Na, and Lind |

As shown below, each reference pre-dates the ’260 patent’s earliest proclaimed priority date (December 28, 2007; “Critical Date”), which Petitioner does not concede.

| Reference | Date                                              | Prior Art At Least Under pre-AIA |
|-----------|---------------------------------------------------|----------------------------------|
| Aaltonen  | Mar. 18, 2004 (published)                         | §102(b)                          |
| Drescher  | Sep. 7, 2006 (filed)<br>Mar. 13, 2008 (published) | §102(e)                          |
| Na        | Jun. 15, 2006 (published)                         | §102(b)                          |
| Takahashi | Oct. 27, 2005 (published)                         | §102(b)                          |
| Lind      | Nov. 17, 2005 (published)                         | §102(b)                          |

## **II. THE '260 PATENT**

### **A. Brief Description**

The '260 patent describes “a Bluetooth™ (BT) enabled mobile device for publishing data and multimedia content on one or more websites.” TTI-1001, 2:3-5, 1:36-40, 3:6-34, 5:18-64; TTI-1003, ¶¶24-25. FIGS. 2 and 4 show a mobile device 202 includes a client application 203. TTI-1001, 3:41-42. “In the case of a mobile device 202 with limited memory and processing capabilities, the client application 203 partitions the multimedia content of large files stored on the mobile device 202 into multiple data segments” and “generates segment identifiers and tags the data segments with the segment identifiers.” TTI-1001, 7:23-29, 7:52-62, 5:4-9, 8:62-9:3. “The tagged data segments are transferred from the client application 203 of the mobile device 202 to a publishing service 401 via a network 402 as illustrated in FIG. 4.” TTI-1001, 5:9-11, 8:65-9:3, 7:29-33, 7:62-67.

FIG. 4 shows the “publishing service 401” includes “a front end service 401a, a back end service 401b, and a database 401d.” TTI-1001, 8:1-6. “If the back end service 401b receives the multimedia content in the form of multiple data segments,” it “reassembles the data segments in a predetermined sequence using the segment identifiers” and “then creates a multimedia object from the transferred data and multimedia content,” which is “then published on the websites.” TTI-

1001, 8:13-22, 9:3-11.

### **B. Summary of the Prosecution History**

The '260 patent issued from U.S. Application No. 14/297,616, which was filed on June 5, 2014 and claims priority through continuation applications to Provisional Application No. 61/017,202 filed on December 28, 2007. TTI-1001, (21)-(22), (63), (60); TTI-1003, ¶26. In the sole office action, claims 1-22 were rejected based on nonstatutory obviousness-type double patenting over claims 1-5 of U.S. Application No. 14/109,972 and over claims 1-30 of U.S. Application No. 14/273,546. TTI-1002, 70-76. The examiner also rejected dependent claims under §112 for reciting new subject matter. *Id.* In response, the applicant filed a terminal disclaimer and argued that the specification provides support for the claims. TTI-1002, 97-158, 167-170. The claims were allowed with an examiner's amendment of independent claims 1, 7, 12, and 18 to include "wherein the segment identifier corresponds to the content range of said file data that is being transferred in a given data segment." TTI-1002, 182-202.

### **C. Level of Ordinary Skill**

A person of ordinary skill in the art at the time of the '260 patent ("POSITA") had at least a Bachelor of Science in electrical engineering, computer engineering, computer science, or a related field, with at least two years of relevant work experience in wireless communications and/or networking systems, or

equivalents thereof. TTI-1003, ¶21. Superior education could compensate for a deficiency in work experience, and vice-versa. *Id.*

#### **D. 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).

### **III. THE CHALLENGED CLAIMS ARE UNPATENTABLE**

#### **A. GROUND 1— Aaltonen and Drescher render obvious claims 1-18**

##### **1. Aaltonen**

Aaltonen discloses “uploading content” with “greater control over the uploading of content by allowing... entities to define instructions to which upload of the media content is subject.” TTI-1005, Abstract, ¶¶[0008]-[0009]; TTI-1003, ¶¶27-29. Aaltonen discloses “recipient,” “sender,” and “network instructions” that “define how content is uploaded to a recipient.” TTI-1005, ¶¶[0063]-[0082].

Aaltonen's FIG. 6<sup>1</sup> shows an example system.

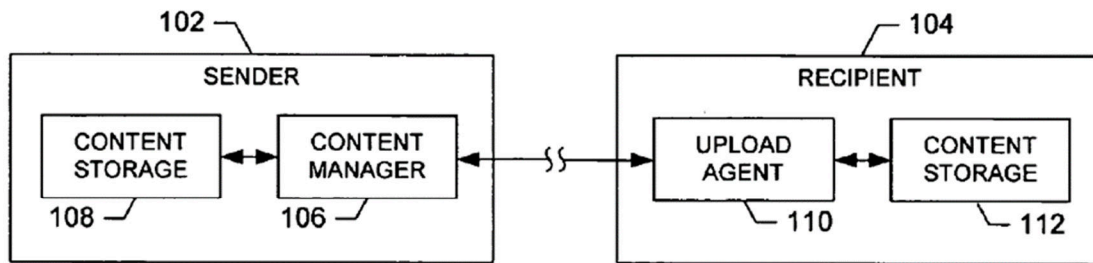


**FIG. 6.**

TTI-1005, FIG. 6.

Aaltonen's content manager "break[s] up the upload content into a plurality of portions" in response to instructions. TTI-1005, ¶[0012], [0069]. Aaltonen uses Hypertext Transfer Protocol ("HTTP") techniques that divide uploaded content into "a number of content data packets." TTI-1005, ¶¶[0101]-[0106]. Aaltonen discloses two specific HTTP techniques of uploading content – "HTTP POST" and "HTTP PUT." TTI-1005, ¶¶[0016], [0101], [0111]-[0113], FIGS. 9, 11.

## **2. Drescher**

Drescher discloses "sending large files or resources over the internet." TTI-

---

<sup>1</sup> Aaltonen refers to the "sender" and "content manager" interchangeably. TTI-1005, ¶¶[0060]-[0062], [0114]-[0115]. The same is true for the "recipient" and "upload agent." *Id.*

1010, Cover, 4:15-30<sup>2</sup>, 5:1-11, 10:14-11:7; TTI-1003, ¶¶30-31. Additionally, Drescher discloses a “HTTP PUT” technique (like Aaltonen) for uploading “chunks” of content where the server sends a “HTTP Response” to the sender’s PUT request that includes “the server’s log of progress of processing the PUT Request.” TTI-1010, 11:27-12:10, 15:23-16:16.

Drescher describes “ParHTTP,” a similar protocol to standard HTTP, for “communicating identities between the client 200 and the server 100.” TTI-1010, 9:16-21, 10:1-12, 12:12-20, 25:32-26:21, 34:13-35:4, 35:26-36:21.

### **3. A POSITA would have combined Aaltonen and Drescher (“Aaltonen and Drescher”)**

A POSITA found it obvious to combine Aaltonen’s and Drescher’s teachings to (1) use HTTP responses to acknowledge data blocks that are successfully received by the recipient via Aaltonen’s HTTP PUT/POST techniques, based on Drescher, and (2) add party-identifying information to Aaltonen’s HTTP PUT/POST messages, based on Drescher. TTI-1010, 14:10-15, 25:32-26:21, 34:13-35:4, 35:26-36:21; TTI-1005, ¶¶[0012], [0016], [0028], [0069], [0093]-[0094], [0096], [0101]-[0102], [0111], [0113], FIG. 10; TTI-1003, ¶¶32, 38; *supra* §III.A.1, §III.A.2. As Dr. Traynor explains, a POSITA would have combined the teachings of Aaltonen and Drescher for at least the reasons below.

---

<sup>2</sup> The citations to Drescher refer to the page number of the PDF document.

First, a POSITA recognized that, for segmented uploads, sending an acknowledgement (e.g., an HTTP response, based on Drescher) for each block indicated to the sender that the successfully received data can be deleted from the sender's content storage. TTI-1005, ¶¶[0093]-[0094], [0096], [0101]; TTI-1010, 14:10-15; TTI-1003, ¶33. A POSITA knew from Aaltonen that acknowledgements indicate to the sender that it can "delete or otherwise remove the upload content from the content storage," "thereby relieving at least a portion of the storage capacity burden on the sender, such as for subsequent storage of other content." TTI-1005, ¶¶[0093]-[0094], [0096], [0101].

Second, HTTP responses, such as those in Drescher, are another means for the sender to expediently detect service interruptions (e.g., a HTTP response indicating a failure alerts the sender that the recipient did not receive the most recently transmitted block). TTI-1005, ¶¶[0111]-[0114], FIG. 11; TTI-1010, 14:10-15, 15:23-16:16; TTI-1003, ¶34.

Third, a POSITA recognized that adding party-identifying information to data transfer messages improves the security of Aaltonen's data transfer techniques, as the recipient can identify when a message is received from a source that does not have "permission" to upload data to the recipient. TTI-1010, 25:4-26:21, 34:13-35:4, 35:26-36:21; TTI-1003, ¶35.

Additionally, such a combination merely involved (1) combining prior art

elements according to known methods to yield predictable results; and (2) use of known techniques to improve similar devices in the same way. *KSR Int'l Co. v.*

*Teleflex Inc.*, 550 U.S. 398, 415-421 (2007); MPEP §2143; TTI-1003, ¶36.

Aaltonen already describes acknowledging data transfers in a segmented data transfer protocol, and the combination of Drescher's teachings simply involved adding known HTTP responses, such as those in Drescher, to Aaltonen's similar HTTP techniques. *Id.*, TTI-1005, ¶¶[0093]-[0094], [0096], [0101]; TTI-1010, 14:10-15. Like Drescher, Aaltonen already discloses adding identifying information to HTTP messages. TTI-1010, 25:4-26:21, 34:13-35:4, 35:26-36:21; TTI-1005, ¶¶[0102], [0104]. Moreover, both Aaltonen and Drescher disclose segmented data transfer protocols implemented in the HTTP protocol, and thus the techniques of Aaltonen and Drescher are compatible, without requiring undue experimentation. TTI-1005, ¶¶[0016], [0056], [0098], [0101], [0111]-[0113]; TTI-1010, Cover, 10:14-15:9; TTI-1003, ¶37. For at least this reason, the Aaltonen-Drescher combination is predictable and foreseeable and would have been implemented with a reasonable expectation of success. *Id.*

#### 4. Analysis

##### (a) Claim 1

###### Element 1[pre].

If the preamble is limiting, Aaltonen discloses a method of “uploading content” (“*file transfer*”) from a “mobile station”/“terminal” (“*mobile device*”) to a “recipient” that can be an “origin server,” “digital broadcaster,” and/or “digital broadcast receiver” (“*a web service*”). TTI-1005, ¶¶[0002]-[0007], [0008], [0038]-[0046], [0052] (“mobile telephones”), [0056], [0097], [0111]-[0115], FIGS. 1, 5, 9-11; *cf.* TTI-1001, 9:37-60 (“mobile phone”); TTI-1003, ¶¶39-40. Aaltonen’s “content manager” “break[s] up... upload content into a plurality of portions” (“*segmented file transfer*”). TTI-1005, ¶[0069]; *infra* 1[c1]. Below is an example network in Aaltonen.



The '260 patent does not define a “*web service*,” but describes “publishing data and multimedia content on one or more websites” (an example of a “*web service*”). TTI-1001, 3:30-40, 5:65-6:8. Similarly, Aaltonen’s recipient is a “origin server,” “digital broadcaster,” “digital broadcast receiver,” and uploads occur over the “Internet” (a network hosting “*web service[s]*”). TTI-1005, ¶¶[0005], [0033], [0038]-[0046], [0056]-[0057]. Aaltonen’s mobile terminal uses “a conventional Web browser for communicating in accordance with... HTTP.” TTI-1005, ¶[0057]. Because the above services hosted “websites,” the recipient is a “*web service*.” *Id.*; TTI-1003, ¶42.

**Element 1[a].**

Aaltonen’s “content manager” application (“*a software module on the Bluetooth enabled mobile device*”) is “capable of uploading... content from a content storage... of the sender, to the recipient.” TTI-1005, ¶¶[0060]-[0062] (content manager is “software operated by the sender” (“*a software module*”)), [0115], FIG. 6. Aaltonen’s content manger formats and uploads content, deletes successfully uploaded content from the sender’s memory, and recovers from upload interruptions. TTI-1005, ¶¶[0060]-[0062], [0066], [0069], [0073], [0077]-[0080], [0084]-[0086], [0089]-[0094], [0111]-[0115]; TTI-1003, ¶43.

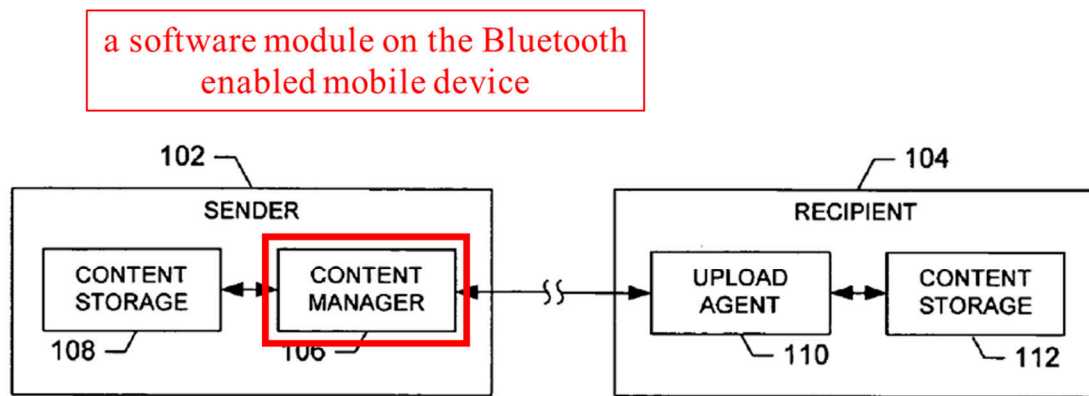


**FIG. 6.**

TTI-1005, FIG. 6.

**Element 1[b].**

Aaltonen discloses “instructions” for uploading content “such that the upload content is... processed based upon the... size... of the content” (“*size of a file to be transferred*”). TTI-1005, ¶¶[0012]-[0013], [0059], [0069], [0078].

Aaltonen’s sender’s (“*Bluetooth enabled mobile device*” per 1[pre]) content manager “*determin[ed]*” the size of the file to be uploaded when specifying instructions based on content size. TTI-1003, ¶¶44-45.

Aaltonen’s sender generates an “upload request”/“upload descriptor” that includes “information necessary to describe the content (e.g., size...)” (“*size of a file to be transferred*”). TTI-1005, ¶¶[0096], [0100]. Thus, the sender’s content manager “*determin[ed]*” the size of the file when providing the content size in the upload descriptor. TTI-1003, ¶46.

Further, Aaltonen’s sender’s content manager “select[s] or receiv[es] a

selection of one or more pieces of content to be uploaded to the recipient” and “select[s] piece(s) of content based upon the... size... of content” (“*size of a file to be transferred*”). TTI-1005, ¶[0084]. Thus, the content manager “*determin[ed]*” the size of the file when selecting uploads based on content size as this information is used to determine what content is uploaded from the sender’s content storage. TTI-1003, ¶47.

**Element 1[c1].**

Aaltonen discloses “breaking... content into a plurality of portions” in response to “instructions” (“*segmenting said file into data segments*”). TTI-1005, ¶¶[0012], [0069]; *supra* 1[pre]. Aaltonen’s FIG. 10 illustrates an upload with several content “groups or blocks” (“*data segments*”). TTI-1005, ¶¶[0028], [0102], FIG. 10. Dr. Traynor confirms that segmenting data for uploads was well known prior to the Critical Date. TTI-1003, ¶¶48-49; TTI-1010, 4:15-28, 9:31-12:24; TTI-1011, ¶¶[0037]-[0039].

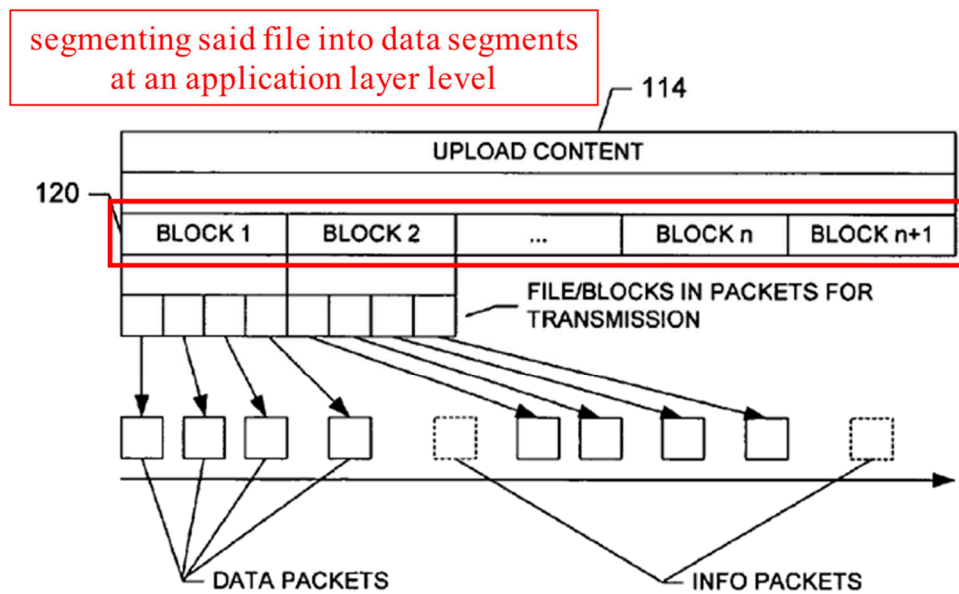


**FIG. 10.**

TTI-1005, FIG. 10.

Because Aaltonen-Drescher uses HTTP, a widely known application layer protocol, the segmentation of data is “*at an application layer level.*” TTI-1005, ¶¶[0016], [0056], [0098], [0101], [0111]-[0113]; TTI-1003, ¶48. Gourley corroborates this understanding and discloses that HTTP is part of the “application layer” (or “web application layer”). TTI-1007, 76, 249-251. Moreover, during the prosecution of the ’757 patent (which shares a specification with the ’260 patent), the applicant acknowledged that “[i]t is well known that both HTTP and XML-RPC are application layer protocols.” TTI-1014, 115.

**Element 1[c2]<sup>3</sup>.**

Aaltonen's uploaded data in the data blocks ("*data segments*") includes "graphic, audio and/or video content" and "text content" ("*file data*"). TTI-1005, ¶¶[0049], [0068], [0070], [0072], [0084]; *cf.* TTI-1001, 4:22-25 ("image files, audio files, video files, text files, or any combination"), 2:48-54, 5:55-59; TTI-1003, ¶¶50-51.

Aaltonen discloses "HTTP header information" included in all data blocks and HTTP messages. TTI-1005, ¶¶[0112]-[0113]. As Dr. Traynor explains, "HTTP header information" in HTTP PUT/POST messages included both the "name of the file" and "type of the file" being uploaded ("*data associated with said file data*"). TTI-1010, 32:25-28 ("content type specified in the 'Content-Type' header"); TTI-1003, ¶52. Indeed, Gourley corroborates Dr. Traynor's testimony and discloses that HTTP PUT messages include both the "name of the file" and "type of the file." TTI-1007, 54-55.

---

<sup>3</sup> The term "said file data" lacks antecedent basis. For the purposes of IPR, this feature has been interpreted as "file data."

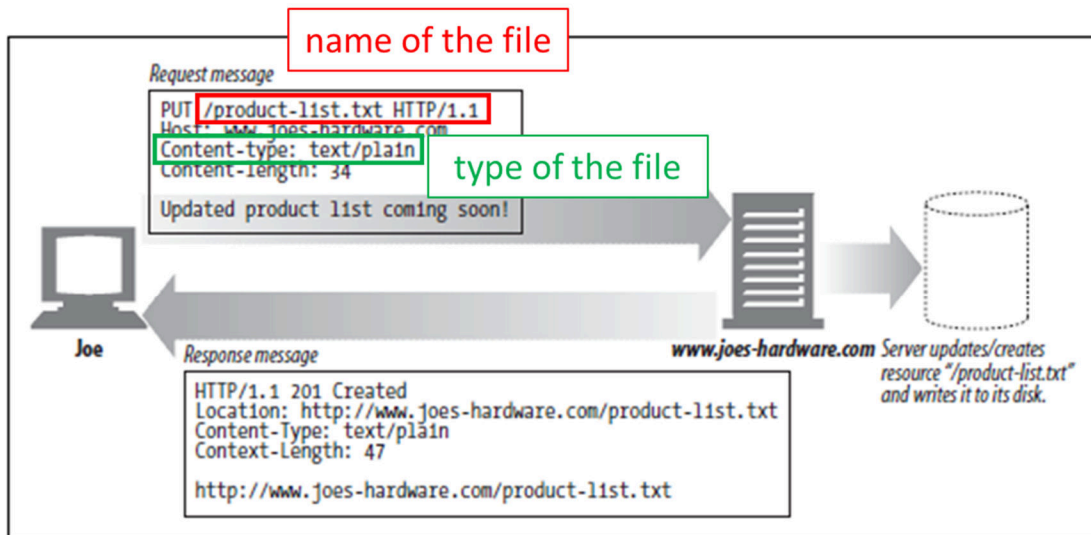


Figure 3-9. PUT example

TTI-1007, 55 (FIG. 3-9).

Additionally, Aaltonen discloses an “instruction” “that orders or prioritizes piece(s) or object(s) within the upload content, such as ordering graphic, audio and/or video content of the upload content” (thus indicating the “type of file” and “*data associated with said file data*”). TTI-1005, ¶[0072]. Indeed, Aaltonen’s recipient uploads and/or stores content based on file type. *Id.* (“a sports video clip with audio commentary” where “the sender could order video at a higher priority” and “a song clip with an audio track and a video of the singer” where “the sender may order the audio a higher priority”). Dr. Traynor explains that type of file is indicated in the HTTP message headers, which was well known before the Critical Date. TTI-1010, 32:25-28; TTI-1003, ¶53 (citing TTI-1007, 54-55).

#### Element 1[d].

Aaltonen discloses that “information packets... added to the packet stream”

“between groups or blocks 120 of one or more packets” or alternatively “data packets” within its HTTP data “blocks” (“*data segments*”) include “an identifier that identifies placement of the respective data packet in relation to the other data packets of the content,” “a number of packets in the block of packets,” “information... uniquely describing the packets,” and “a table uniquely identifying the block of packets before or after the respective information[/data] packet” (examples of “*applying a segment identifier to each of said data segments, wherein the segment identifier corresponds to the content range of said file data*”). TTI-1005, ¶¶[0101]-[0104], FIG. 10; TTI-1003, ¶54.

This information is included for or in “*each*” data block (“*each data segment*”) for the upload agent to identify the proper placement of blocks in the upload content as they are received. TTI-1003, ¶55. As Dr. Traynor explains, identifiers are preferentially added to “*each*” block as Aaltonen relies upon this information to “monitor the received... blocks of packet(s) or portion of the upload content,” determine if an interruption has been experienced, and “facilitate recovery of the upload session” following an interruption, and “neglecting to add identifying information to any one block would create the potential that a recipient would not be able to properly dispose of that data block.” *Id.*, TTI-1005, ¶¶[0106]-[0107].

Because Aaltonen-Drescher uses HTTP, these identifiers are applied to data

blocks “*at the application layer level.*” TTI-1003, ¶56; *supra* 1[b]; TTI-1007, 76, 249-251; TTI-1014, 115.

**Element 1[e]<sup>4</sup>.**

Aaltonen discloses that the information/data packets for its HTTP data “blocks” (“*data segments*”) include “identifiers identifying the sender.” TTI-1005, ¶¶[0101]-[0102], [0104]. Moreover, Aaltonen uploads content “in a number of content data packets” “to the recipient in accordance with an HTTP POST/PUT technique,” and Drescher’s HTTP PUT/POST messages have a user identifier disclosing “a name particularising the... [party] involved in the request” (“*applying a user identifier*” that “*uniquely identifies a particular user of the web service*”). TTI-1005, [0101], [0111]; TTI-1010, 25:32-26:21, 34:13-35:4, 35:26-36:21. Drescher discloses that this information is preferentially added “for each chunk, and... PUT” (each data packet of the Aaltonen-Drescher combination). *Id.* Also, because Drescher’s user identity is used to determine “permissions” to allow “access to resources at the server” (the recipient), this information is “*for the destination web service.*” TTI-1010, 25:4-25, 34:13-35:4, 35:26-36:21. Because each data packet within a data block includes this information, “*each*” data block

---

<sup>4</sup> The term “destination web service” lacks antecedent basis. For the purposes of IPR, this feature has been interpreted as “web service.”

(“*each data segment*”) includes this information. *Supra* 1[c1]-1[d]; TTI-1003, ¶57.

Additionally, these identifiers are applied to packets “*at the application layer level.*” TTI-1003, ¶58; *supra* 1[d]. Further, the “upload” is directed to the “*web service,*” and the applied “*user identifier*” is for “*each destination web service*” because Aaltonen’s “recipient” (“*web service*”) “utilize[s] the [identifiers within] information packets to monitor the received data packets” and “track the number and accuracy of the received packets, blocks of packet(s) or portion of the upload content.” TTI-1005, ¶[0104].

**Element 1[f].**

Aaltonen discloses content uploaded to a recipient in “groups or blocks 120 of one or more packets” (“*transferring a data segment... to the web service*”). TTI-1005, ¶¶[0101]-[0106], FIGS. 8-11. Aaltonen’s FIG. 10 depicts an example upload. TTI-1003, ¶59.

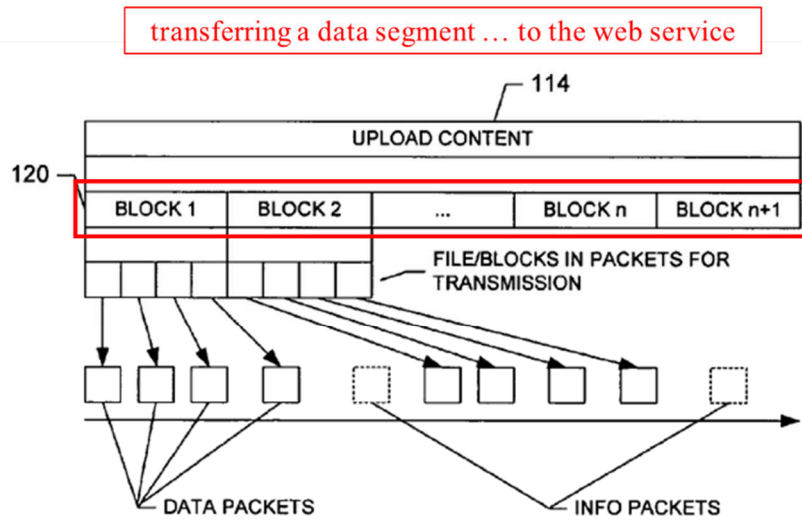


**FIG. 10.**

TTI-1005, FIG. 10.

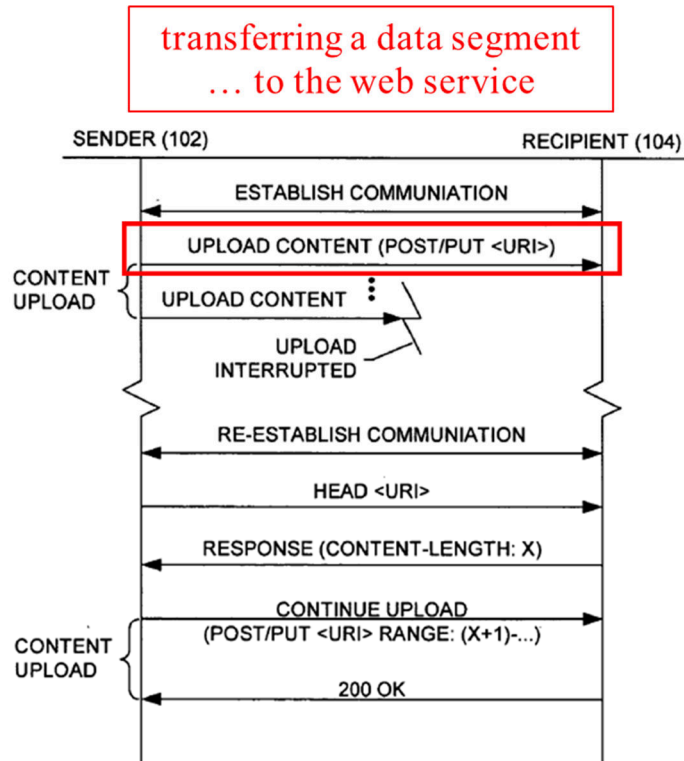


**FIG. 11.**

TTI-1005, FIG. 11.

As discussed above, each “block” of packets includes a “*user identifier*” and a “*segment identifier*.” *Supra* 1[d]-1[e]. These identifiers are transferred “*along with*” the block and used by the recipient for block identification and disposition (e.g., to “identify if, and when, an upload session has experienced an interruption”). TTI-1003, ¶60; TTI-1005, ¶¶[0102]-[0113].

**Element 1[g].**

During an upload, Aaltonen’s recipient “acknowledge[s] the upload” by sending “an acknowledgement to the terminal” (“*a reply message from the web service*”). TTI-1005, ¶¶[0093]-[0094], [0096], [0101]. For segmented uploads, the recipient sends an acknowledgement to the sender for each “*successful completion of said data segment transfer*” to indicate to the sender that a data “block” has been received successfully and can subsequently be deleted from the sender’s content storage. TTI-1003, ¶61; TTI-1005, ¶¶[0093]-[0094], [0096], [0101]. Indeed, Aaltonen discloses that acknowledgements indicate to the sender that it can “begin uploading content 114 to the recipient” and “delete or otherwise remove the upload content from the content storage.” *Id.*

If Aaltonen does not explicitly disclose that an acknowledgement is sent for each data segment transfer, Drescher renders this obvious. TTI-1010, 11:27-12:10, 15:23-16:16. Drescher discloses a “HTTP PUT” technique (like Aaltonen) for uploading “chunks” of content where the server sends a “HTTP Response” to the

sender's PUT request that includes "the server's log of progress of processing the PUT Request." TTI-1010, 11:27-12:10, 15:23-16:16. Drescher's "HTTP Response" is used by the recipient to "acknowledge" (via the "progress" indicated by a "HTTP Response") a successful transmission of a "block" sent using a "HTTP PUT" technique<sup>5</sup> ("*a synchronization message from the web service, corresponding to successful completion of said data segment transfer*"). TTI-1010, 11:27-12:10, 15:23-16:16; TTI-1005, ¶¶[0016], [0101], [0111], [0113]; TTI-1003, ¶62. Indeed, the HTTP response includes "the server's log of progress of processing the PUT Request" and thus indicates the "progress" of the upload, and accordingly, a *successful completion of said data segment transfer* (and vice-versa) as a failed upload is indicated in the server's processing log. *Id.*

Dr. Traynor confirms that sending acknowledgement messages for each portion of a segmented upload was well known. TTI-1003, ¶63; TTI-1009, ¶¶[0002]-[0004], [0009], [0097].

#### **Element 1[h].**

Aaltonen discloses that an upload continues until all data blocks are

---

<sup>5</sup> Dr. Traynor notes that "responses could also easily be sent in response to other HTTP techniques, for example, Aaltonen's "HTTP POST" technique. TTI-1003, ¶62 (citing TTI-1005, ¶¶[0016], [0101], [0111], [0113]).

transferred (“*repeating the step of data segment transfer and the step of receiving the reply message, until all the data segments have been transferred to the web service successfully*”). TTI-1005, ¶¶[0101]-[0114]; FIGS. 7, 9-11. Aaltonen’s FIG. 7 illustrates an upload successfully completed with no interruption. TTI-1003, ¶64.

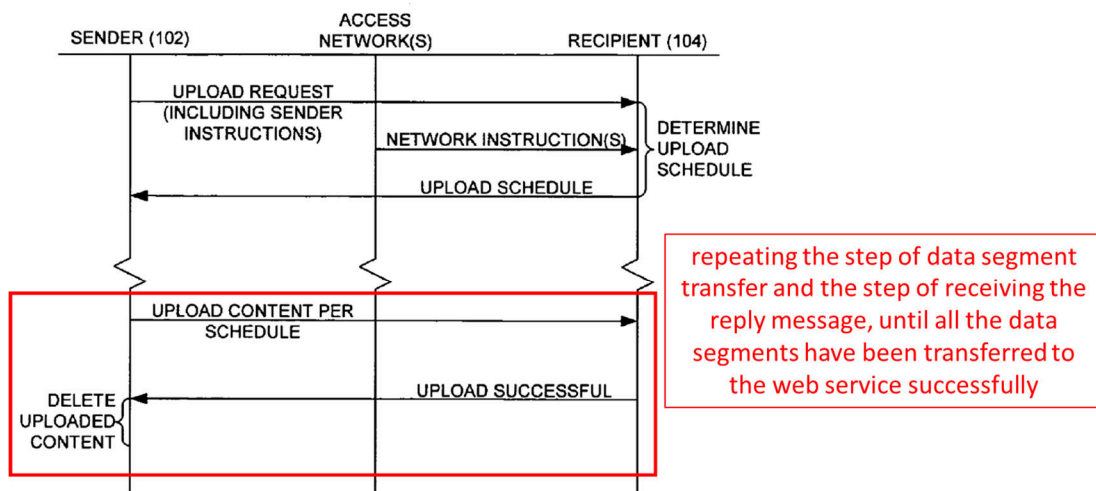


**FIG. 7.**

TTI-1005, FIG. 7.

#### Element 1[i].

Aaltonen discloses recovering from an “interruption” during an upload (“*segmented data transfer is interrupted*”). TTI-1005, ¶¶[0106]-[0114], FIGS. 9, 11. After an interruption, Aaltonen’s sender “request[s] the bit-length or size of the portion of the upload content 114 received” by the recipient and then “continue[s] uploading the content to the recipient” based on the response. TTI-1005, ¶¶[0111]-[0113], FIG. 11. The sender’s “content manager” (“*the software*

*module further performs*”) performs the above operations as Aaltonen discloses that the “sender 102 (e.g., terminal 10... generally operates under control of a computer program product (e.g., content manager...)” (“*software module*”). TTI-1003, ¶65; TTI-1005, ¶[0115].

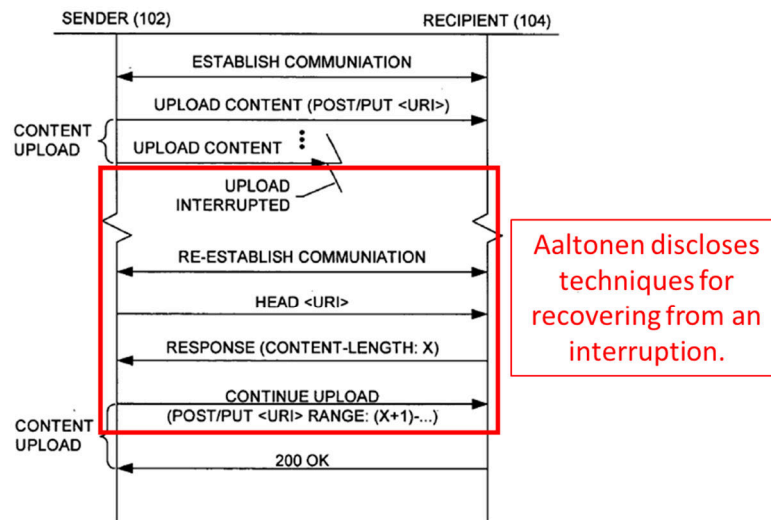


**FIG. 11.**

TTI-1005, FIG. 11.

### Element 1[j].

Upon experiencing an interruption, Aaltonen’s sender can “re-establish communication with the recipient” and then “request the bit-length or size of the portion of the upload content 114 received” by the recipient (“*sending, a synchronization request to the web service... corresponding to asking the web service about the segmented file data the web service has successfully received... after a network connection to the web service is restored*”). TTI-1005, ¶[0111]-

[0112]. The request from the sender is “*for said user identifier*” because the interrupted upload is associated with the user identifier. TTI-1005, ¶[0102], FIG. 11; *supra* 1[c]; TTI-1003, ¶66.

Additionally, Aaltonen’s “request” sent from the sender is in accordance with the HTTP HEAD technique” (“*from the application layer level*” – HTTP operates at the application layer). TTI-1005, ¶[0112]; *supra* 1[c1]; TTI-1003, ¶67.

Aaltonen’s FIG. 11 illustrates the sender re-establishing communication with the recipient and sending a synchronization message to the recipient following an interruption. TTI-1003, ¶68.

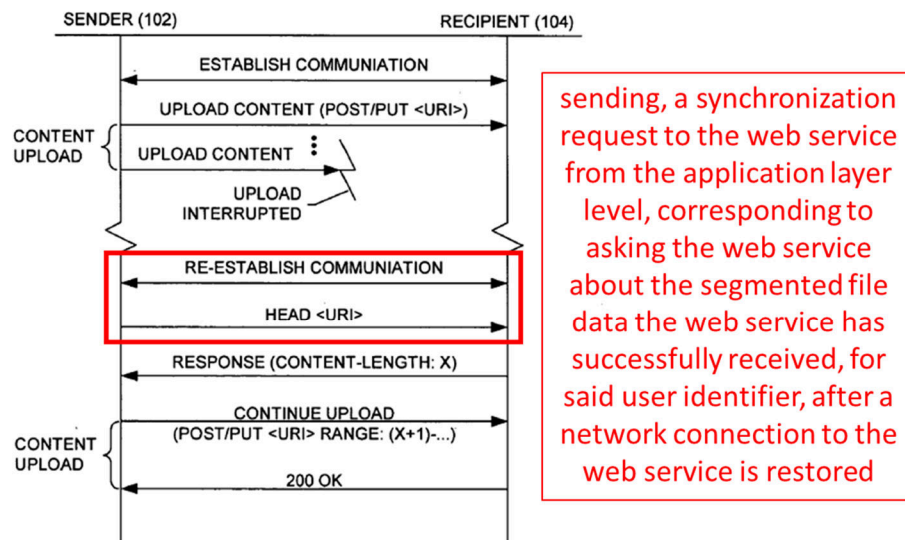


**FIG. 11.**

TTI-1005, FIG. 11.

### Element 1[k].

After receiving a synchronization request from a sender, Aaltonen’s

recipient “return[s] a response, such as a 200 OK message, that includes header information (e.g., content-length) specifying the size of the received portion of the upload content” (“*receiving, a synchronization reply back from the web service..., corresponding to indicate segmented file data said web service has successfully received*”). TTI-1005, ¶[0112]. The “200 OK message” is a HTTP message sent “*at the application layer level*” in response to a HTTP HEAD message. *Id.*; *supra* 1[c1]. The request from the sender is “*for said user identifier*” because the interrupted upload and content is associated with the user identifier identifying the sender of the upload content. TTI-1005, ¶[0102], FIG. 11; *supra* 1[e]. Aaltonen’s FIG. 11 illustrates the recipient sending a synchronization reply to the sender following a synchronization message. TTI-1003, ¶69.

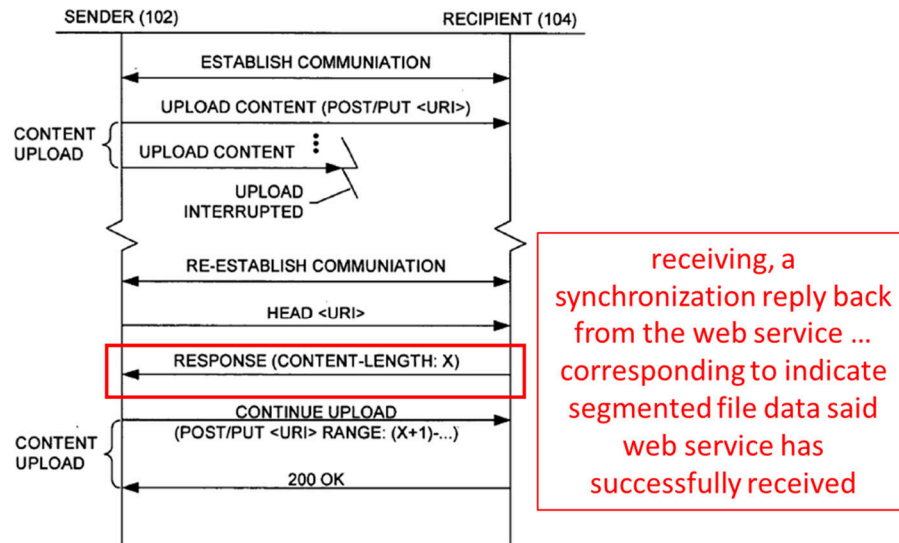


**FIG. 11.**

TTI-1005, FIG. 11.

**Element 1[l].**

Aaltonen discloses that “[u]pon receiving the length of the received portion of the upload content, the sender can continue uploading the content to the recipient 104 to complete the uploading session” (“*resuming, the segmented file transfer based on the synchronization reply received*”) and the sender uploads “only the remaining portion of the content” (“a range of ‘X+1’ through the last bit of the upload content”). TTI-1005, ¶¶[0113]-[0114], FIG. 11. The sender resuming the upload is “*for said user identifier*” because the upload is associated with the user identifier. TTI-1005, ¶[0102], FIG. 11; *supra* 1[e]; TTI-1003, ¶70.

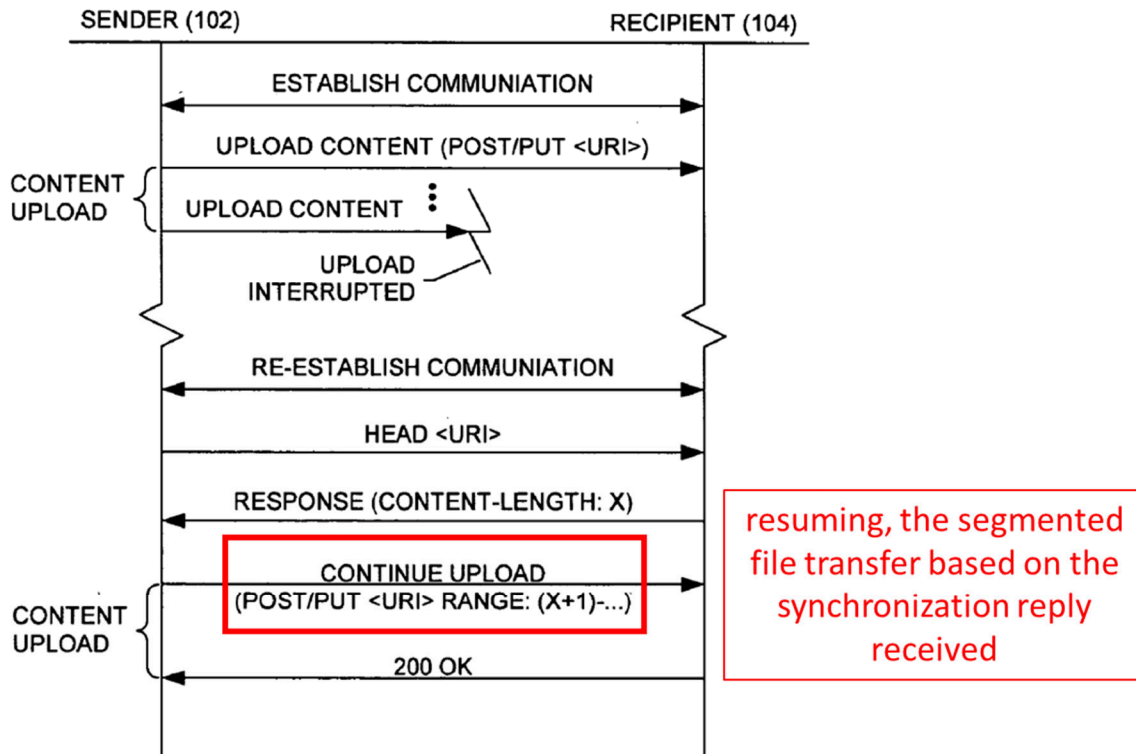


**FIG. 11.**

TTI-1005, FIG. 11.

**Element 1[m].**

In Aaltonen, the upload continues “until the upload session has concluded” (“*receiving, at the web service, all the data segments along with said user identifier and segment identifiers for said file transfer*”). TTI-1005, ¶¶[0102], [0114], FIG. 10; *supra* 1[h]; TTI-1003, ¶71.

**Element 1[n].**

Aaltonen’s recipient receives content and “store[s] the upload content... in content storage” (“*storing, at the web service, the received file data segments*”)

along with “each information[/data] packet” that contains the associated identifiers (*“storing, at the web service, segment identifiers for said user identifier”*). TTI-1005, ¶¶[0009], [0038], [0044], [0047], [0054], [0060]-[0061], [0093], [0098], [0102], [0110], FIGS. 6, 8. The upload content (including the *“received file data segments and segment identifiers”*) are *“for said user identifier”* as the user is associated with the upload session. TTI-1005, ¶¶[0102], [0104]; *supra* 1[e]; TTI-1003, ¶72.

**Element 1[o].**

Aaltonen’s “digital broadcast receiver” (*“the web service”*) includes “interface means for sharing... data from electronic devices” (*“making available, at the web service, the segmented data received from the Bluetooth enabled mobile device for public or private consumption”*). TTI-1005, ¶¶[0045]-[0046]. The digital broadcast receiver includes a “network interface means” for obtaining data from the “Internet” (*“over the internet”*). *Id.* Dr. Traynor explains that “digital broadcast receivers” share uploaded content (*“the segmented data received from the Bluetooth enabled mobile device”*) because Aaltonen discloses that these components are for “sharing and/or obtaining data from... a terminal” (e.g., the *“the Bluetooth enabled mobile device”* performing the upload). TTI-1003, ¶73. Dr. Traynor further explains that *“making [data] available”* (e.g., through a download) at an internet web service was well known before the Critical Date. *Id.*;

TTI-1010, 8:5-19, 9:23-33.

**(b) Claim 2**

The Aaltonen-Drescher combination renders obvious claim 2 for similar reasons as discussed above for 1[c2]. TTI-1003, ¶¶74-75.

**(c) Claim 3**

The Aaltonen-Drescher combination renders obvious claim 3 for similar reasons as discussed above for 1[c2]. Additionally, as Dr. Traynor explains, Aaltonen transferred “**digital data**” because Aaltonen discloses “MPEG-2 transport streams” – a common digital video coding standard. TTI-1003, ¶76; TTI-1005, ¶¶[0049]-[0050].

**(d) Claim 4**

Drescher’s user identifier includes “a username and password pair” (“*wherein the user identifier comprises... user-name, [or] user-password*”). TTI-1010, 25:32-26:21; *supra* 1[e]; TTI-1003, ¶77.

**(e) Claim 5**

Aaltonen-Drescher uses HTTP, which is an “**Internet protocol suite**” that segments data at the “**application layer**.” TTI-1005, ¶¶[0016], [0056], [0098], [0101], [0111]-[0113]; TTI-1007, 76, 249-251; *supra* 1[c1]; TTI-1003, ¶78. Moreover, Dr. Traynor explains it was well known before the Critical Date that the “**application layer**” was “**layer 7 in the OSI layer stack**.” *Id.*; TTI-1015, 2:58-4:40

(describing the “OSI model”).

**(f) Claims 6-9**

The Aaltonen-Drescher combination renders claims 6-9 obvious in the same manner as explained above for the corresponding claim elements listed in the following table. TTI-1003, ¶¶79-108.

| Claim Element | Corresponding Claim Element |
|---------------|-----------------------------|
| 6[pre]        | 1[pre]                      |
| 6[a]          | 1[a]                        |
| 6[b]          | 1[b]                        |
| 6[c]-6[g]     | See analysis below          |
| 6[h]          | 1[j]                        |
| 6[i]-6[k]     | See analysis below          |
| 6[l]          | 1[d]                        |
| 6[m]          | 1[e]                        |
| 6[n]          | 1[f]                        |
| 6[o]          | 1[m]                        |
| 6[p]          | 1[n]                        |
| 6[q]          | 1[o]                        |
| 6[r]          | See analysis below          |
| 7             | 3                           |
| 8             | 4                           |

|   |   |
|---|---|
| 9 | 5 |
|---|---|

**Element 6[c].**

Aaltonen's FIG. 10 shows the "upload content" is segmented into a plurality ("first" and "second") of data "blocks" ("*data segments*"). TTI-1005, ¶¶[0012], [0069], FIG. 10; TTI-1003, ¶82; *supra* 1[pre], 1[c1].

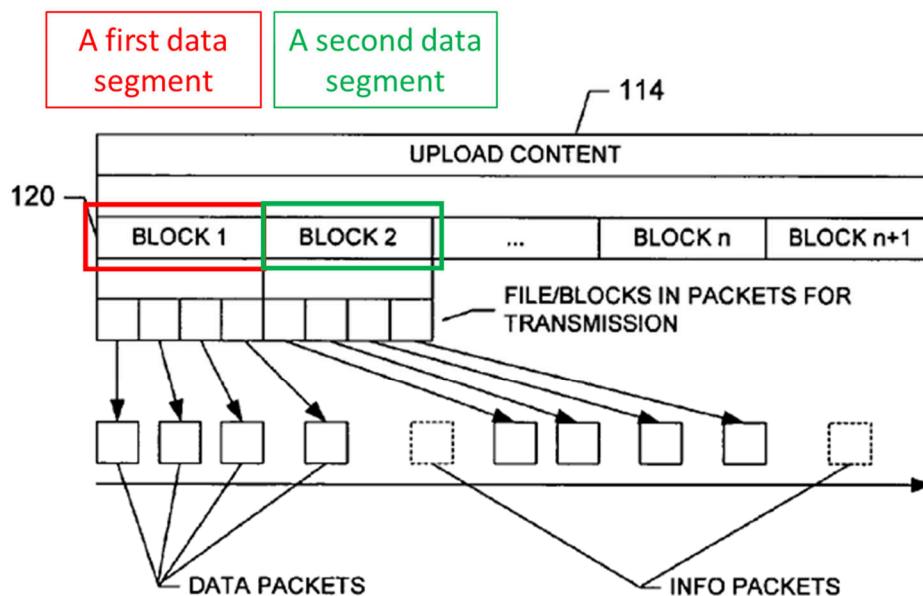


**FIG. 10.**

TTI-1005, FIG. 10.

Because Aaltonen-Drescher uses HTTP, data is segmented "*at an application layer level.*" TTI-1005, ¶¶[0016], [0056], [0098], [0101], [0111]-[0113]; TTI-1007, 76, 249-251; TTI-1003, ¶83.

**Element 6[d] <sup>6</sup>.**

Aaltonen's data blocks ("***data segments***") include uploaded data, e.g., "graphic, audio and/or video content" and "text content," ("***file data***") and "HTTP header information" that includes the "name of the file" and "type of the file" being uploaded ("***data associated with multimedia data***"). *Supra* 1[c2].

Accordingly, a first data block ("***first data segment***") includes "HTTP header information" that includes the "name of the file" and "type of the file" being uploaded ("***data associated with multimedia data***"), and a second data block ("***second data segment***") includes uploaded data, e.g., "graphic, audio and/or video content" and "text content," ("***file data***"). TTI-1003, ¶¶84-86.

---

<sup>6</sup> This claim element uses the open-ended term "comprises," which is well understood to mean "including but not limited to." *CIAS, Inc. v. All. Gaming Corp.*, 504 F.3d 1356, 1360-61 (Fed. Cir. 2007). Accordingly, the claimed data segments can comprise other data.

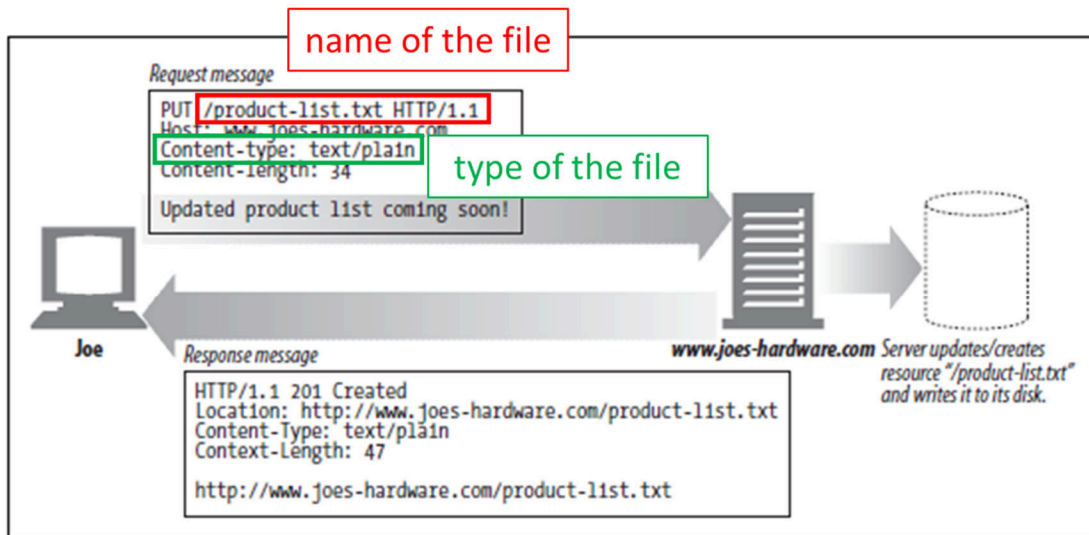


Figure 3-9. PUT example

TTI-1007, 55 (FIG. 3-9).

#### Elements 6[e1]-6[e3].

The reasons discussed above for 1[c1], 1[e], 1[f], 1[g], and 6[c] apply to each data segment and therefore apply to “*said first data segment*” of 6[e1]-6[e3].

TTI-1003, ¶¶87-89.

#### Element 6[f1]-6[f3].

The reasons discussed above for 1[c1], 1[e], 1[f], 1[g], and 6[c] apply to each data segment and therefore apply to “*said second data segment*” of 6[f1]-6[f3]. TTI-1003, ¶¶90-92.

#### Element 6[g].

The reasons discussed above for 1[i] and 6[c] apply to interruptions for “*said second data segment.*” TTI-1003, ¶93.

**Element 6[i].**

The reasons discussed above for 1[k] and 6[c] apply to “*said second data segment.*” TTI-1003, ¶95.

**Element 6[j].**

Aaltonen discloses that “[u]pon receiving the length of the received portion of the upload content, the sender can continue uploading the content to the recipient 104 to complete the uploading session” (“*resuming, the segmented file transfer by transferring remaining data of the second data segment based on the synchronization reply received*”) and uploads “only the remaining portion of the content.” TTI-1005, ¶¶[0113]-[0114]; *supra* 1[1]; TTI-1003, ¶96.

**Element 6[k].**

When Aaltonen’s sender recovers from an upload, it sends “only the remaining portion of the content” (thus, *creating one or more data segments at the application layer level* (per 15[c1]), *wherein the one or more data segments comprise remaining data of the second data segment*). TTI-1005, ¶¶[0113]-[0114]. Moreover, Dr. Traynor explains the *create[ed] one or more data segments* included the *remaining data of the second data segment* because retransmitting successfully received data is inefficient. TTI-1003, ¶97. Because this claim limitation recites creating “one or more” segments, transmitting the entirety of the “*remaining data of the second data segment*” as one data segment satisfies the

claim. *Id.*; TTI-1005, ¶¶[0113]-[0114].

Additionally, because Aaltonen-Drescher uses HTTP, data blocks are created “*at the application layer level.*” TTI-1003, ¶98 (citing TTI-1007, 76, 249-251).

**Element 6[r].**

Drescher’s HTTP PUT/POST messages include “a name particularising the... [party] involved in the request” (“*one or more portions of the data correspond to a particular user of the web services*”). TTI-1010, 25:32-26:21, 34:13-35:4, 35:26-36:21. Drescher discloses that this information is preferentially added “for each chunk, and... PUT” (each “*portions of the data*”). *Id.*; *supra* 1[c1], 1[e]; TTI-1003, ¶105.

**(g) Claims 10-14**

The Aaltonen-Drescher combination renders claims 10-14 obvious in the same manner as explained above for the corresponding claim elements listed in the following table. TTI-1003, ¶¶109-129.

| Claim Element | Corresponding Claim Element |
|---------------|-----------------------------|
| 10[pre]       | 1[pre]                      |
| 10[a]         | 1[a]                        |
| 10[b]         | 1[b]                        |
| 10[c1]        | 1[c1]                       |

|             |                    |
|-------------|--------------------|
| 10[c2]      | 1[c2]              |
| 10[d]       | 1[d]               |
| 10[e]       | 1[e]               |
| 10[f]       | 1[f]               |
| 10[g]       | 1[g]               |
| 10[h]       | 1[h]               |
| 10[i]       | 1[i]               |
| 10[j]       | 1[j]               |
| 10[k]       | 1[k]               |
| 10[l]       | 1[l]               |
| 10[m]       | 6[o]               |
| 10[n]-10[o] | See analysis below |
| 11          | 2                  |
| 12          | 3                  |
| 13          | 4                  |
| 14          | 5                  |

**Element 10[n].**

Aaltonen’s information/data packets include “a table uniquely identifying the block of packets before or after the respective information packet,... and the correct order of the packets” (“*segment identifiers*”), and Drescher’s identifiers include “a name particularising the... [party] involved in the request” (“*a user identifier*”). TTI-1005, ¶¶[0102]-[0104]; TTI-1010, 25:32-26:21, 34:13-35:4,

35:26-36:21. Dr. Traynor explains the recipient uses these identifiers, both the order of the data blocks and the user identifier, to reassemble received data blocks in the correct order when storing the content. TTI-1003, ¶124. Indeed, Aaltonen discloses that “[b]y directly or indirectly identifying the order of the data packets, the upload agent can subsequently order the packets in the same order that the [sender] uploaded the packets” (“*reassembling the received file data segments automatically using segment identifiers and a user identifier... at the web service*”). TTI-1005, ¶[0103]. Additionally, because the Aaltonen-Drescher combination uses HTTP, this process is performed “*at the application layer level.*” TTI-1003, ¶124.

**Element 10[o].**

Aaltonen’s “digital broadcast receiver” (“*the web service*”) includes “one or more interface means for sharing... data from electronic devices” (“*making available, at the web service, the reassembled data file, for public or private consumption over the internet*”). TTI-1005, ¶¶[0045]-[0046]; *supra* 1[o]. Dr. Traynor explains the data made available includes a “reassembled data file” because a reassembled file would be more desirable than its constituent segments taken individually (e.g., a complete video is more beneficial than a particular frame). TTI-1003, ¶125.

**(h) Claims 15-18**

The Aaltonen-Drescher combination renders claims 15-18 obvious in the same manner as explained above for the corresponding claim elements listed in the following table. TTI-1003, ¶¶130-154.

| Claim Element | Corresponding Claim Element |
|---------------|-----------------------------|
| 15[pre]       | 1[pre]                      |
| 15[a]         | 1[a]                        |
| 15[b]         | 1[b]                        |
| 15[c]         | 6[c]                        |
| 15[d]         | 6[d]                        |
| 15[e1]        | 6[e1]                       |
| 15[e2]        | 6[e2]                       |
| 15[e3]        | 6[e3]                       |
| 15[f1]        | 6[f1]                       |
| 15[f2]        | 6[f2]                       |
| 15[f3]        | 6[f3]                       |
| 15[g]         | 6[g]                        |
| 15[h]         | 1[j]                        |
| 15[i]         | 6[i]                        |
| 15[j]         | 6[j]                        |
| 15[k]         | 6[k]                        |

|       |       |
|-------|-------|
| 15[l] | 1[d]  |
| 15[m] | 1[e]  |
| 15[n] | 1[f]  |
| 15[o] | 1[m]  |
| 15[p] | 10[n] |
| 15[q] | 10[o] |
| 16    | 3     |
| 17    | 4     |
| 18    | 5     |

**B. GROUND 2A—Takahashi and Na render obvious claims 1-26**

**GROUND 2B—Takahashi, Na, and Lind render obvious claims 1-26**

**1. Takahashi**

Takahashi discloses a mobile phone that divides image data into a predetermined size and sequentially transmits the divided image data to a server via a network by an HTTP request. TTI-1016, ¶¶(0005)-(0008), (0020)-(0030), (0042)-(0043), (0049), (0057), (0080); TTI-1003, ¶¶155-156. Takahashi's mobile phone "adopts a sequential transmission rule for transmitting a divided file one at a time and transmitting the next divided file after waiting for the response of the reception completion" on the server side. TTI-1016, ¶(0058). The mobile phone repeats transmissions of divided image data until all the image data is transmitted.

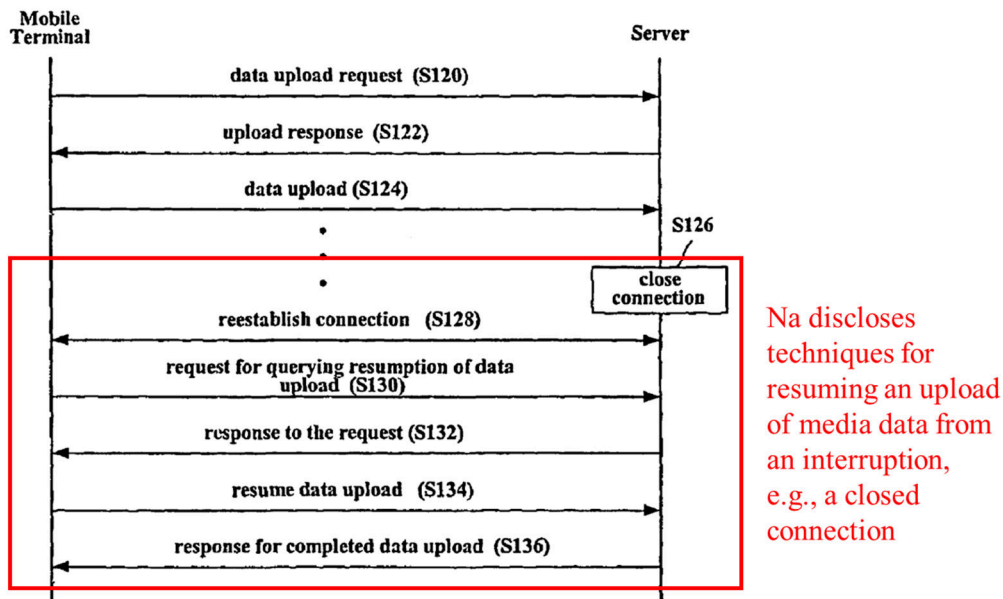
TTI-1016, ¶¶(0068), (0072), FIG. 8. The server receives the HTTP request from the mobile phone and “combin[es] the divided image data, if the image data has been divided, to be one image data... to be published to a web page.” TTI-1016, ¶¶(0061)-(0063), (0051), (0073), (0075)-(0079), (0089), FIG. 9.

## **2. Na**

Similar to Takahashi, Na discloses “controlling a media message upload” by exchanging “requests and responses... between a communication terminal and a server.” TTI-1006, Abstract, ¶[0001]; TTI-1003, ¶¶157-159. Na discloses that the HTTP protocol is used for the media data uploading. TTI-1006, ¶¶[0052], [0062]. Na explains that, per the HTTP 1.1 protocol, the media data is segmented; data segments are transmitted in request messages; and “response messages are repeatedly issued... for verification of the upload.” TTI-1006, ¶¶[0003]-[0005].

Na also discloses recovering from an “interruption” such as “an upload error” or “a closed connection” during an upload. TTI-1006, ¶¶[0054], [0070], [0084]-[0087], [0091]-[0094], [0113]-[0115], [0139]-[0157], FIGS. 11-12, 24-25. Na’s “FIG. 12 illustrates resuming an upload of media data when the communication connection is reestablished.” TTI-1006, ¶¶[0036], [0091]-[0097]. As another example, Na’s FIG. 25 illustrates “resuming an upload of media data stopped by a user.” TTI-1006, ¶¶[0049], [0148]-[0157].

FIG. 25



TTI-1006, FIG. 25.

### 3. Lind

Lind describes “electronic devices which are capable of being interconnected... by means of Bluetooth. Such devices include... mobile telephones.” TTI-1017, 7:27-8:3<sup>7</sup>; TTI-1003, ¶160. In Lind, “a first electronic device in the form of a mobile telephone, and a second electronic device in the form of a passive accessory, such as a headset or a hands free kit” is “provided with means for communicating by radio, such as a Bluetooth connection.” TTI-1017, 8:11-16.

<sup>7</sup> The citations to Lind refer to the page number of the PDF document.

**4. A POSITA would have combined Takahashi and Na (“Takahashi-Na”)**

Takahashi discloses transmitting divided media data using HTTP requests. TTI-1016, ¶¶(0005)-(0008), (0020)-(0030), (0042)-(0043), (0049). Takahashi does not appear to disclose details regarding the HTTP protocol and HTTP requests for transmitting divided media data. TTI-1003, ¶161. As such, a POSITA would have been motivated and found it obvious and straightforward to complement Takahashi with Na’s detailed disclosure of the HTTP protocol and HTTP requests used for media data uploading. TTI-1006, ¶¶[0052], [0062].

Further, Takahashi recognizes instances where a “communication is not completed normally.” TTI-1016, ¶(0071). Likewise, a POSITA recognized that, in mobile communications such as Takahashi’s, “a data upload error caused by various communication problems may occur during an upload of media data,” and “it is desirable to resume the upload in such a way that the upload is resumed from the point at which the error occurred.” TTI-1006, ¶¶[0006]-[0008]. A POSITA would have been motivated and found it obvious and straightforward to incorporate Na’s teachings in Takahashi’s system to beneficially allow “an upload of media data stopped by an upload error to be resumed from the point where the error occurred,” as taught by Na. TTI-1006, ¶¶[0008], [0010]; TTI-1003, ¶162. In doing so, a POSITA also would have been motivated and found it obvious and

straightforward to implement a mobile phone similar to the one disclosed by Takahashi, based on Na, to include a division/segment identifier with each divided media data to be uploaded so that the media data is “easily identified [by the server] when the data upload is stopped and resumed afterwards” and to receive from the server “a parameter about the range” of the divided media data “that has been successfully uploaded... when an upload error occurs” so that the mobile phone knows what part of the divided media data is to be uploaded additionally. TTI-1006, ¶¶[0012]-[0013], [0022]. In this way, in the combination, the mobile phone need not restart upload of the whole media data but rather “resumes the upload based on the sequence number of the segments, thereby improving upload efficiency and speed.” TTI-1006, ¶[0021].

Such a combination merely involved (1) combining prior art elements according to known methods to yield predictable results and (2) use of known techniques to improve similar devices in the same way. *KSR*, 550 U.S. at 415-421; MPEP §2143; TTI-1003, ¶163. Indeed, Takahashi already discloses transmitting divided media data using HTTP requests. TTI-1016, ¶¶(0005)-(0008), (0020)-(0030), (0042)-(0043), (0049). Na also describes data transfers using HTTP requests according to a segmented data transfer protocol, such as HTTP 1.1 protocol. TTI-1006, ¶¶[0003]-[0004], [0052]-[0053]. As such, the techniques of Takahashi and Na are compatible, without requiring undue experimentation, and

implementing a communication system similar to the one disclosed by Takahashi based on Na's teachings to also allow "an upload of media data stopped by an upload error to be resumed from the point where the error occurred," (TTI-1006, ¶¶[0008], [0010]), simply involved implementing a known feature in a similar system, making the Takahashi-Na combination predictable and foreseeable with a reasonable expectation of success. *Id.*; TTI-1003, ¶164.

**5. A POSITA would have combined Takahashi, Na, and Lind ("Takahashi-Na-Lind")**

Takahashi's techniques are implemented in a mobile phone. BD-1016, ¶¶(0049)-(0053), FIGS. 1-3. Similarly, Na's FIG. 1 shows a mobile phone as an example of "mobile communication terminal." TTI-1006, [0051]. A POSITA recognized and found obvious that one functionality of a mobile phone is communication via Bluetooth. TTI-1003, ¶¶165-167. As Dr. Traynor explains, by the Critical Date, Bluetooth was a common feature of mobile phones. *Id.*; TTI-1009, ¶¶[0016], [0030]-[0031], [0042]-[0043], [0049], [0054]; TTI-1011, ¶[0034], Claim 6; TTI-1012; TTI-1005, ¶¶[0056], [0036]-[0037], [0046]; TTI-1018, [0106]-[0107], [0165]; TTI-1017, 7:27-8:16.

A POSITA would have been motivated to include Bluetooth functionalities in the Takahashi-Na mobile phone to beneficially provide the mobile phone with capabilities for communicating with other Bluetooth accessories such as Bluetooth

headsets commonly available at the time of the '757 patent. *Id.*; TTI-1003, ¶168.

Indeed, a mobile phone “allows voice communications between the mobile communication terminal... and another mobile communication terminal,” as explicitly disclosed in Na (TTI-1006, ¶[0056]), for which a Bluetooth headset beneficially provides a wireless hands-free means of voice communication, as taught by Lind. TTI-1017, 6:9-10, 8:11-16, 9:12-16, 16:7-8. As such, a POSITA found it obvious to implement the Takahashi-Na mobile phone as a Bluetooth enabled mobile device that supported Bluetooth functionalities, based on Lind. *Id.*; TTI-1003, ¶168.

Additionally, implementing the Takahashi-Na mobile phone as a Bluetooth enabled mobile device was obvious at least because such a combination merely involved (1) combining prior art elements according to known methods to yield predictable results and (2) use of known techniques to improve similar devices in the same way. *KSR*, 550 U.S. at 415-421; MPEP §2143; TTI-1003, ¶169. Such a modification to the Takahashi-Na mobile phone simply involved implementing Bluetooth functionality into Takahashi-Na’s mobile phone, as taught by Lind, making the modification predictable and foreseeable with a reasonable expectation of success. *Id.*

## **6. Analysis**

Below, the Petition presents Ground 2A based on Takahashi-Na and Ground

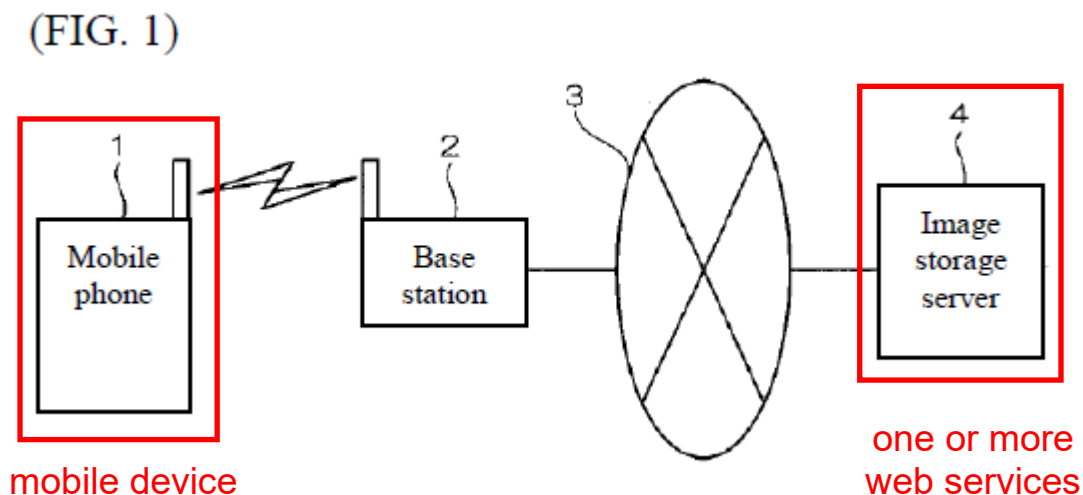
2B based on Takahashi-Na-Lind. Ground 2A explains why Takahashi's and Na's disclosures, when interpreted in view of the general knowledge of a POSITA, renders obvious claims 1-18. Ground 2B relies on the same disclosures of Takahashi and Na and the explanation of obviousness as presented in Ground 2A, but turns to Lind's disclosure for a single feature of claims 1-18 ("***a Bluetooth enabled mobile device***"). Although all features of claims 1-18 are rendered obvious by Takahashi-Na, Petitioner has advanced Takahashi-Na-Lind for claims 1-18 to the extent that the single feature of claims 1-18 is not rendered obvious by Takahashi-Na and the general knowledge of a POSITA. Because the use of Takahashi-Na and corresponding obviousness arguments are similar for both grounds, the Petition discusses these grounds together for ease of presentation. TTI-1003, ¶170.

**(a) Claim 1**

**Element 1[pre].**

If the preamble is limiting, Takahashi discloses a method of transmitting image data/file that are divided into "predetermined size [data]"/"divided file[s]" ("***segmented file transfer***") from a "mobile phone" ("***mobile device***") to "an image storage server." TTI-1016, Abstract, ¶¶(0005)-(0008), (0020)-(0030), (0042)-(0043), (0049), FIG. 1. Takahashi's "**image data**" is stored in a **file** with "a predetermined **file** name." TTI-1016, ¶(0056). Takahashi's mobile phone

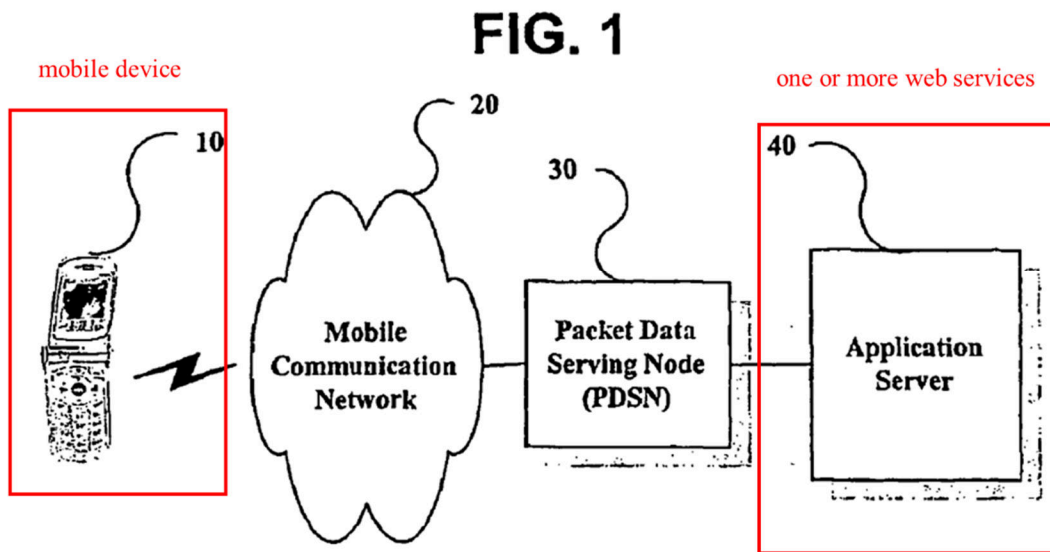
“carr[ies] out division and transmission of the **image file**” and “adopts a sequential transmission rule for **transmitting a divided file** one at a time”) (“segmented file transfer”). TTI-1016, ¶(0058). Takahashi discloses sequentially sending the divided image data/file “as an HTTP request.” TTI-1016, Abstract, ¶¶(0005)-(0008), (0020)-(0030), (0042)-(0043), (0049); TTI-1003, ¶171.



TTI-1016, FIG. 1.

Similarly, Na discloses a “method for controlling a media message upload” (“**file transfer**”) of “media data” from “a mobile communication terminal” (“**mobile device**”) to “an application server.” TTI-1006, Abstract, ¶¶[0051]-[0052], FIGS. 1, 10-14, 24-25. Na performs data upload “in accordance with the HTTP 1.1 protocol,” “performing successive uploads of the segments” (“segmented data transfer”) of media data. TTI-1006, ¶¶[0003]-[0004], [0022], [0052]-[0053], [0112]; *infra* 1[c1]. Na’s “media data upload” is a **file** transfer because, as Dr. Traynor explains, the data transfer over communication network is sometimes

referred to as “file” transfer because the data is represented in a data structure that is generally referred to as a “data file” or simply “file.” TTI-1003, ¶172; TTI-1024, [0005], [0012]. Moreover, Na corroborates this understanding and discloses that the **uploaded data** are stored as “a temporary **file**.” TTI-1006, Abstract, ¶[0068].



TTI-1006, FIG. 1.

For reasons discussed in §III.B.5, a POSITA found it obvious based on their general knowledge and would have been motivated to implement Takahashi-Na’s mobile phone to include Bluetooth functionality. TTI-1003, ¶173; TTI-1009, ¶¶[0016], [0030]-[0031], [0042]-[0043], [0049], [0054]; TTI-1011, ¶[0034], Claim 6; TTI-1012; TTI-1005, ¶¶[0056], [0036]-[0037], [0046]; TTI-1017, 3:4-22, 7:27-8:3; TTI-1018, [0106]-[0107], [0165]. But to the extent that Takahashi-Na in view of the general knowledge of POSITA does not render obvious a Bluetooth enabled

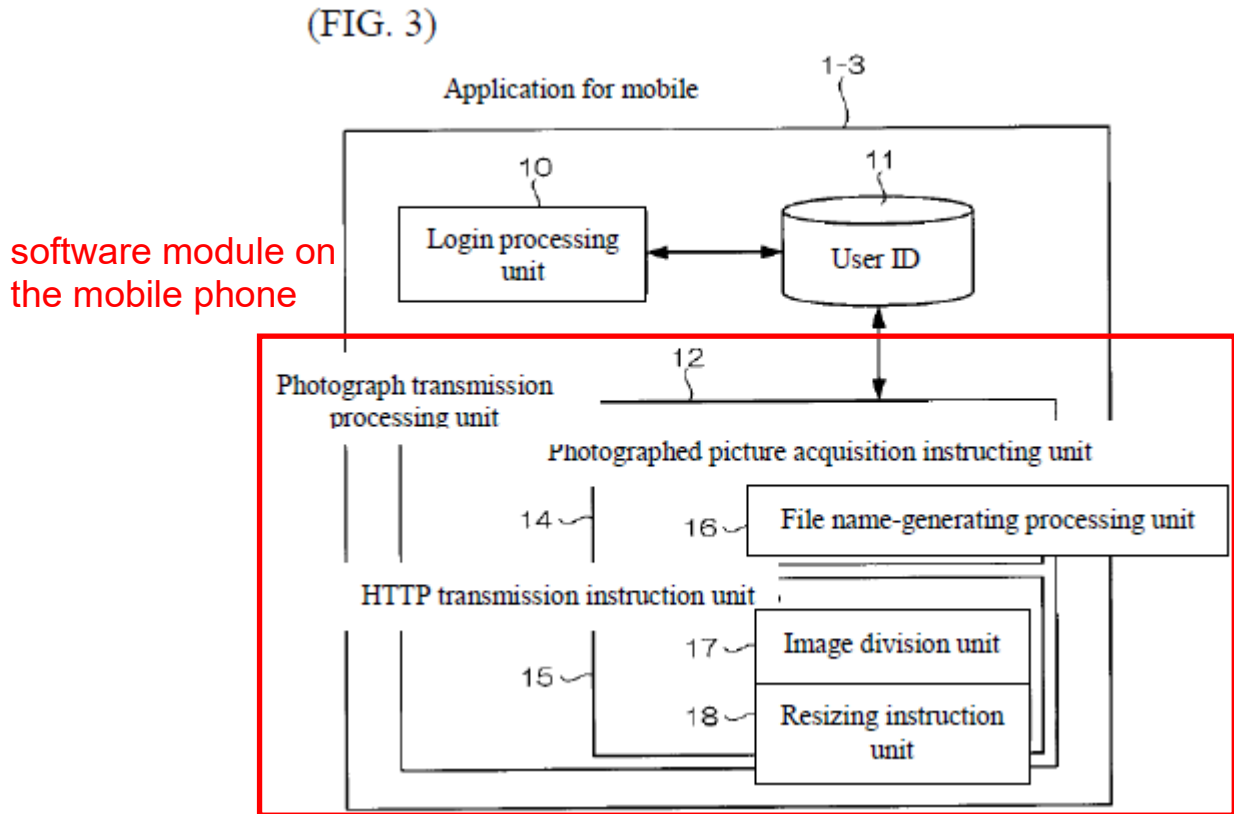
mobile device, Takahashi-Na-Lind renders this obvious as discussed in §III.B.3 and §III.B.5. In view of the general knowledge of a POSITA and Lind’s explicit disclosure, a POSITA found it obvious to implement the Takahashi-Na mobile phone as a **Bluetooth enabled mobile device** that supported Bluetooth functionalities. *Id.*; TTI-1003, ¶¶174-175.

Similar to the ’260 patent, Takahashi’s “image storage server” has “a WEB (World Wide Web) server” and publishes images “on a web page.” TTI-1016, ¶¶(0052), (0054), (0061)-(0062), FIG. 2; *cf.* TTI-1001, 3:30-40, 5:65-6:8 (“publishing data and multimedia content on one or more websites”). Takahashi and Na each discloses operating over the “Internet,” which was well known in the art to host websites. TTI-1016, ¶(0050); TTI-1006, ¶¶[0002]-[0003]; TTI-1003, ¶176. Takahashi’s server hosts a website, and accordingly, the Takahashi-Na server is a “**web service**.” *Id.*

#### **Element 1[a].**

Takahashi discloses a “software configuration of the mobile phone” that includes “an application for mobile [devices] for performing the transmission control of the image data.” TTI-1016, ¶(0053), FIGS. 2-3. Takahashi’s application software is a **software module**. TTI-1003, ¶177. Takahashi’s application software shown in FIG. 3 includes “a photograph transmission processing unit 12” having “a photographed picture acquisition instruction unit 14

and an HTTP transmission instruction unit 15.” TTI-1016, ¶¶(0055)-(0056).



TTI-1016, FIG. 3.

Further, Na discloses a “method” performed by the “mobile communication terminal.” TTI-1006, ¶¶[0016]-[0020]. It would have been obvious that Na’s mobile terminal included a *software module* to implement the disclosed method, for benefits such as flexibility and low cost of software implementations as opposed to hardware implementations. TTI-1003, ¶178.

#### Element 1[b].

Takahashi discloses, if “**the size of image data is larger than a predetermined size**, the mobile phone 1 automatically divides the captured image

data into predetermined size [data] and sequentially transmits the divided image data to a preset transmission destination (image storage server...) as an HTTP request.” TTI-1016, ¶¶(0049), (0057), (0080). The mobile phone application first determines the size of the image data (“*determining size of a file to be transferred*”) to determine whether to “automatically divide[] the captured image data into predetermined size [data].” *Id.*; TTI-1003, ¶179.

Moreover, Na discloses a request message header that includes “the number of total segments in the media data (X-Total-Seq...)” and “the length of one segment of the media data given in bytes (Content-Length...)” or “content length given in bytes (Contents-Length...).” TTI-1006, ¶¶[0062]-[0063], [0116]-[0117], [0122], FIGS. 2, 15. The mobile phone application first determines the size of the whole data in bytes (“*determining size of the data to be transferred*”) in order to insert the number of total segments to be transferred, the length of one segment, or the content length into the header of the request message when uploading the data. TTI-1003, ¶¶180-181.

#### **Element 1[c1].**

Takahashi’s “mobile phone 1 automatically divides the captured image data into predetermined size [data]” (“*segmenting said file into data segments*”). TTI-1016, ¶¶(0049), (0057), (0060), (0080); *supra* §§III.B.1-5, 1[pre]. Takahashi’s “divided image data” (“*data segments*”) are sequentially transmitted “by the

application” for the mobile phone “as an HTTP request.” *Id.*; TTI-1016, ¶¶(0053), (0055)-(0057), (0068), (0085), (0090), FIG. 3. Takahashi’s file is segmented “*at the application layer level*” because the file is segmented “by the application” for the mobile phone to be sequentially transmitted “as an HTTP request,” and HTTP is an application-level protocol. *Id.*; TTI-1003, ¶182; TTI-1007, 7, 76, 249-251; TTI-1002, 115.

There were multiple different ways to format Takahashi’s “divided image data” (“*data segments*”) “as an HTTP request.” TTI-1003, ¶¶183-186. For example, Na teaches: (1) a request message preceding each upload of divided image data (TTI-1006, ¶¶[0052]-[0053]; TTI-1003, ¶183) and (2) a request message including divided image data in the message body for each upload of divided image data (TTI-1006, ¶¶[0112]-[0113], [0039], [0116]-[0118], [0151], FIG. 15; TI-1003, ¶¶184-186).

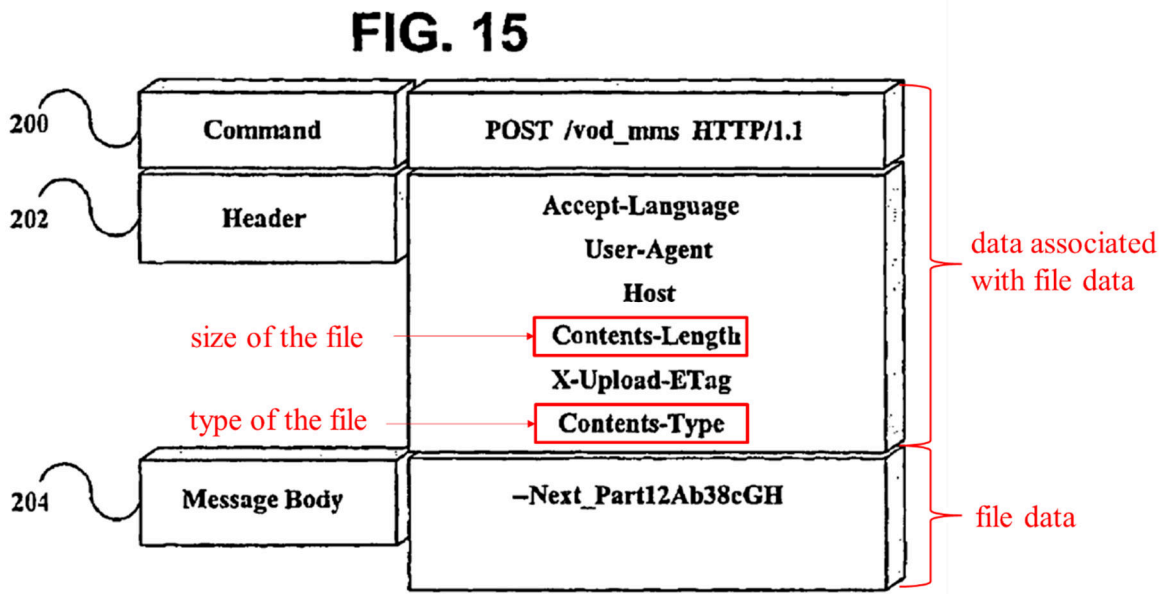
Because Na performs operations “in accordance with the **HTTP** 1.1 protocol,” data is segmented “*at the application layer level*” for upload with or in multiple “request messages” pursuant to the HTTP protocol. TTI-1006, ¶¶[0052], [0111], [0003]-[0004]; TTI-1003, ¶187; TTI-1019, 1; TTI-1007, 7, 76, 249-251; TTI-1002, 115.

#### **Element 1[c2].**

Takahashi’s mobile phone application divides the image data and transmits

each data division (“*file data*”) in a divided file (“*data segment*”) with a file name (“*data associated with said file data*”) so that “when the divided file (image data) with the same file name has already been received [by the image storage server], the divided file is written as an addition (combination) to the existing file.” TTI-1016, ¶¶(0010), (0013), (0032), (0035), (0066), (0070)-(0071), (0073), FIG. 5; TTI-1003, ¶188.

Na’s terminal transfers to the server request messages including a header with content type information and content length information (“*data associated with said file data*”), as shown in FIG. 2 and FIG. 15, along with data segments of the media data (“*file data*”). TTI-1006, ¶¶[0053], [0062]-[0063], [0112]-[0113], [0116]-[0118], FIG. 15; TTI-1003, ¶¶189-190.



TTI-1006, FIG. 15.

A POSITA would have been motivated to implement Takahashi's mobile phone application to transfer image data and data (e.g., a file name, content type, content length) associated with the image data, as suggested by Na, to facilitate the reception, storage, and publication of the image data at the server. TTI-1003, ¶191.

**Element 1[d].**

Takahashi discloses “[w]hen carrying out division and transmission of the image file from the mobile phone 1, the information showing the order of the file such as division number (Ex. 5/10, or the like)” (“*a segment identifier*”) “may be attached and sent, if needed” “in the [HTTP] transmission” that includes a division of the data (*applied to a data segment*). TTI-1016, ¶(0058). Takahashi's mobile

phone application transmits each data division “as an HTTP request.” TTI-1016, ¶¶(0049), (0053), (0055)-(0057), (0068), (0085), (0090), FIG. 3; *supra* 1[c1]. The division number is applied to each data division “*at the application layer level*” pursuant to the HTTP protocol. TTI-1003, ¶192; TTI-1006, ¶¶[0003]-[0004]; TTI-1019, 1; TTI-1007, 7, 76, 249-251; TTI-1002, 115.

There were multiple different ways to apply Takahashi’s division number (“*segment identifier*”) to each divided image data (“*data segment*”). TTI-1003, ¶193. For example, Na teaches a request message that includes the division number for each divided image data (“*segment*”) to be uploaded (“*applying a segment identifier to each data segment*”). TTI-1006, ¶¶[0012], [0022], [0053], [0062]-[0063], [0078], [0112]-[0113], [0116]-[0117], FIGS. 2, 15; TTI-1003, ¶¶193-198. In this way, in the combination, the mobile phone need not restart upload of the whole media data when the data upload is stopped but rather “resumes the upload based on the sequence number of the segments, thereby improving upload efficiency and speed.” TTI-1006, ¶[0021]; *supra* §III.B.4; TTI-1003, ¶198.

Na discloses performing such an operation “in accordance with... the **HTTP** 1.1 protocol,” and thus the division number included in the request message is applied to each divided image data “*at the application layer level*” pursuant to the HTTP protocol. TTI-1003, ¶199; TTI-1006, ¶¶[0003]-[0004]; TTI-1019, 1; TTI-

1007, 7, 76, 249-251; TTI-1002, 115; *supra* 1[c1]-1[c2].

**Element 1[e].**

Takahashi discloses that “a file name of the image data” contains at least “a user ID specific to a user from the server after logging-in” (“*user identifier... for each destination web service*” that “*uniquely identifies a particular user of the web service*”). TTI-1016, ¶¶(0010), (0013), (0032), (0035), (0064), (0066), (0070), FIG. 5. The file name is “in the format of user ID + YYYYMMDDHHMISS (year, month, day, hour, minute, and second).” TTI-1016, ¶¶(0056), (0084), (0087). Takahashi discloses that in “transmitting the divided image using HTTP, the generated file name” including the user ID “is also transmitted together” (“*applying a user identifier to each data segment*”) and “when the divided file (image data) with the same file name has already been received [by the image storage server], the divided file is written as an addition (combination) to the existing file.” TTI-1016, ¶(0071), (0073); TTI-1003, ¶200.

Additionally, Takahashi’s mobile phone application transmits each divided file “as an HTTP request.” TTI-1016, ¶¶(0049), (0053), (0055)-(0057), (0068), (0085), (0090), FIG. 3; *supra* 1[c1]. Therefore, the file name including the user ID is applied to each divided file “*at the application layer level*” pursuant to the HTTP protocol. TTI-1003, ¶201; TTI-1006, ¶¶[0003]-[0004]; TTI-1019, 1; TTI-1007, 7, 76, 249-251; TTI-1002, 115.

There were multiple different ways to apply Takahashi's file name including the user ID ("*user identifier*") to each divided image data ("*data segment*"). TTI-1003, ¶202. For example, Na teaches a request message that includes the user ID for each divided image data ("*segment*") to be uploaded ("*applying a user identifier to each data segment*"). TTI-1006, ¶¶[0053], [0062]-[0063], [0078], [0116]-[0117], FIGS. 2, 15; TTI-1003, ¶¶202-205; TTI-1020, 4:7-10, 4:17-19, 2:60-63, 9:10-13, 12:46-49; TTI-1021, 7:39-45; TTI-1022, 3:25-27; TTI-1019, 8, 89.

Additionally, as discussed above in 1[c1]-1[d], the identifier applied to the request message in accordance with HTTP 1.1 Protocol is "*at the application layer level.*" TTI-1003, ¶206; TTI-1006, ¶¶[0003]-[0004]; TTI-1019, 1; TTI-1007, 7, 76, 249-251; TTI-1002, 115.

#### **Element 1[f].**

Takahashi discloses that in "transmitting the divided image using HTTP, the generated file name" including the user ID "is also transmitted together" ("*transferring a data segment along with said user identifier*") and "when carrying out division and transmission of the image file from the mobile phone 1, the information showing the order of the file such as division number (Ex. 5/10, or the like)" ("*a segment identifier*") "may be attached and sent, if needed" "in the [HTTP] transmission" that includes a division of the data ("*transferring a data*

*segment along with... said segment identifier*"). TTI-1016, ¶¶(0071), (0073), (0058); *supra* 1[d]-1[e]. Takahashi's "mobile phone... sequentially transmits the divided image data to a preset transmission destination (image storage server...)" (*"transferring the data segment... to the web service"*) "as an HTTP request." TTI-1016, ¶¶(0049), (0005)-(0008), (0020)-(0030), (0042)-(0043); *supra* 1[b]; TTI-1003, ¶207.

Takahashi-Na's mobile phone application transmits the divided image data (*"transferring a data segment"*) together with the file name including the user ID (*"along with said user identifier"*) and a division number (*"said segment identifier"*) to the server (*"web service"* per 1[pre]) so the divided image data "can be easily identified [by the server] when the data upload is stopped and resumed afterwards." TTI-1006, ¶¶[0012]-[0013], [0022]. In this way, the mobile phone need not restart upload of the whole media data when the data upload is stopped but rather "resumes the upload based on the sequence number of the segments, thereby improving upload efficiency and speed." TTI-1006, ¶[0021]; *supra* §III.B.4; TTI-1003, ¶208.

There are multiple ways to transfer divided image data along with the file name including the user ID (*"user identifier"*) and division number "as an HTTP request." TTI-1003, ¶209. For example, Na teaches generating and sending a request message that includes the user ID (*"user identifier"*) and division number

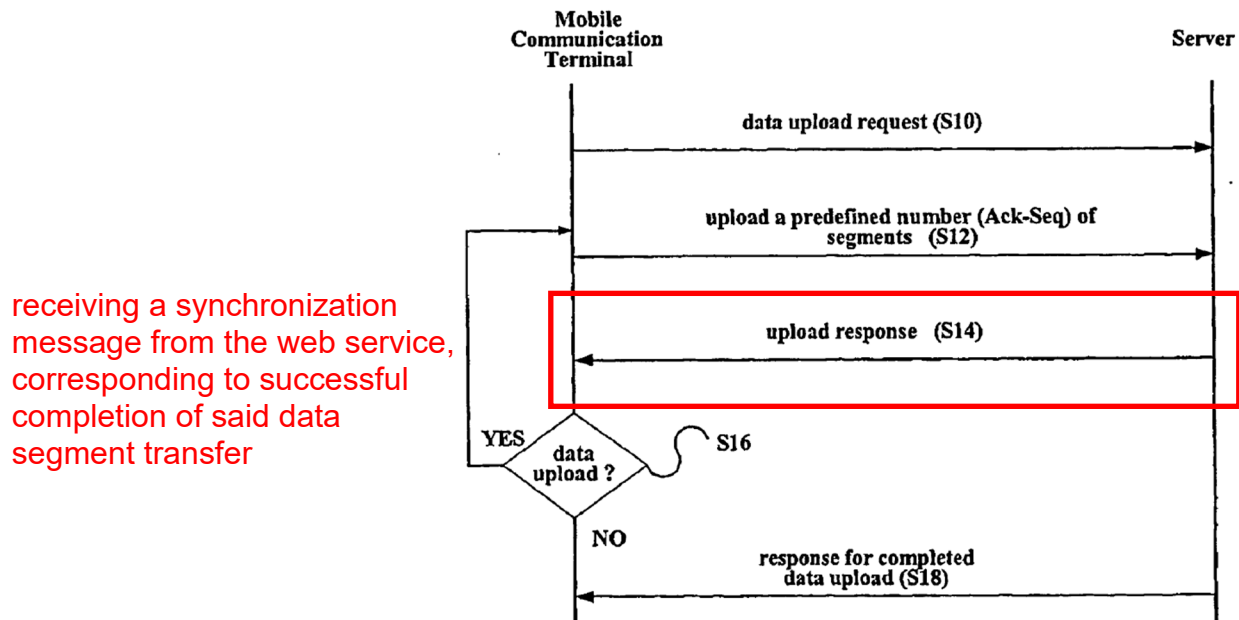
(“*segment identifier*”) for each divided image data (“*segment*”) to be uploaded along with or as part of the request message to the server (“*web service*”). TTI-1006, ¶¶[0053], [0062]-[0063], [0078]; [0112], [0116]-[0117], [0148]-[0149], FIGS. 2, 10, 15, 25; TTI-1003, ¶¶209-212.

**Element 1[g].**

Takahashi’s “mobile phone 1 adopts a sequential transmission rule for transmitting a divided file one at a time and transmitting the next divided file after waiting for the response of the reception completion on the image storage server 4 side” (“*receiving a reply message from the web service, corresponding to successful completion of said data segment transfer*”). TTI-1016, ¶(0058); TTI-1003, ¶213.

Additionally, Na discloses that in “the case where a mobile communication terminal uploads particular media data to a server on the wireless network, the mobile communication terminal issues a request message each time it uploads a segment of the media data in accordance with the HTTP 1.1 protocol and **the server issues a response message for each request message**, thereby performing successive uploads of the segments.” TTI-1006, ¶¶[0004], [0058], [0066]-[0067], [0080], FIGS. 3, 10; TTI-1003, ¶¶214-215; TTI-1019, 38. Takahashi-Na’s response message is *a reply message from the web service corresponding to successful completion of a data segment transfer*. TTI-1003, ¶216.

**FIG. 10**



BD-1006, FIG. 10.

**Element 1[h].**

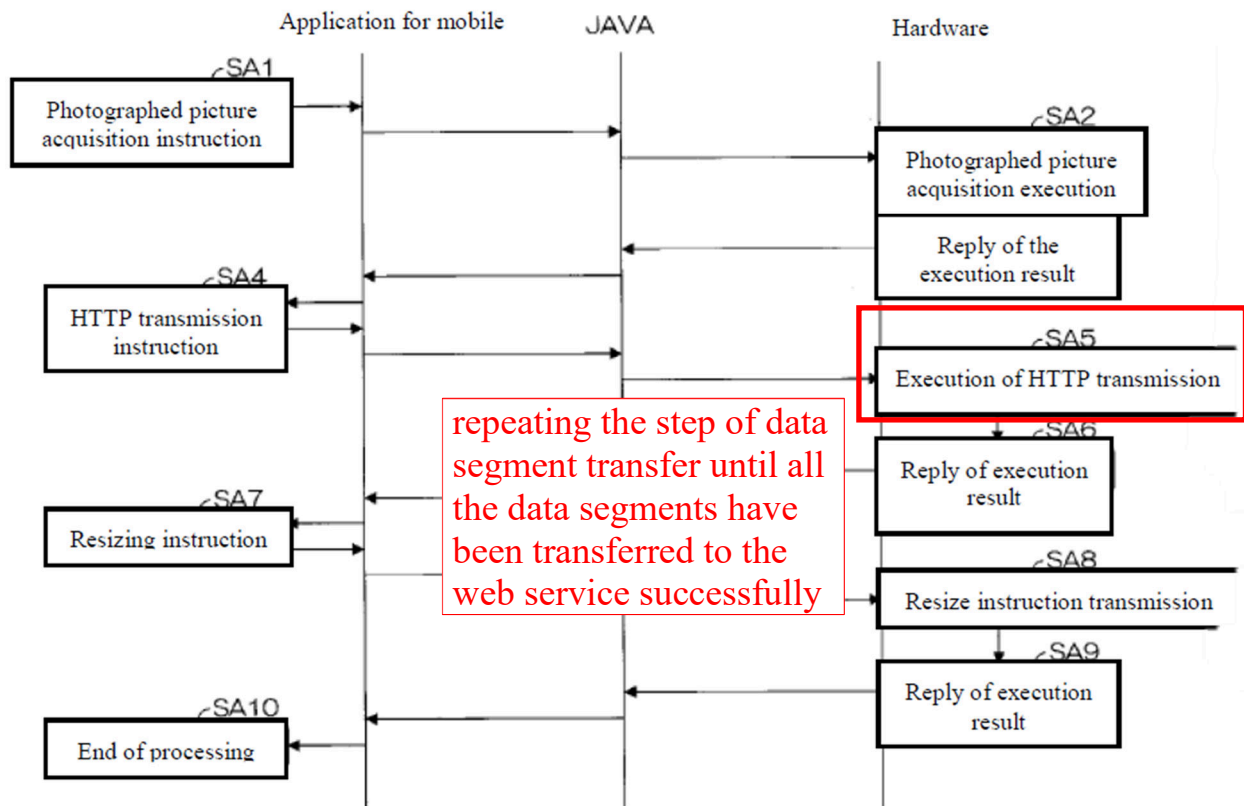
Takahashi discloses that when “image data is divided into several data at this time, step SA5 is repeated, and once the transmission of all the divided image data is completed, the executed result is returned to the application for mobile”

*(“repeating the step of data segment transfer... until all the data segments have been transferred to the web service successfully”)*, as shown in FIG. 6. TTI-1016,

¶(0068). Takahashi’s “mobile phone 1 adopts a sequential transmission rule for transmitting a divided file one at a time and transmitting the next divided file after waiting for the response of the reception completion on the image storage server 4 side” (*“the step of receiving the reply message”*). TTI-1016, ¶(0058); *supra* 1[g];

TTI-1003, ¶217.

(FIG. 6)

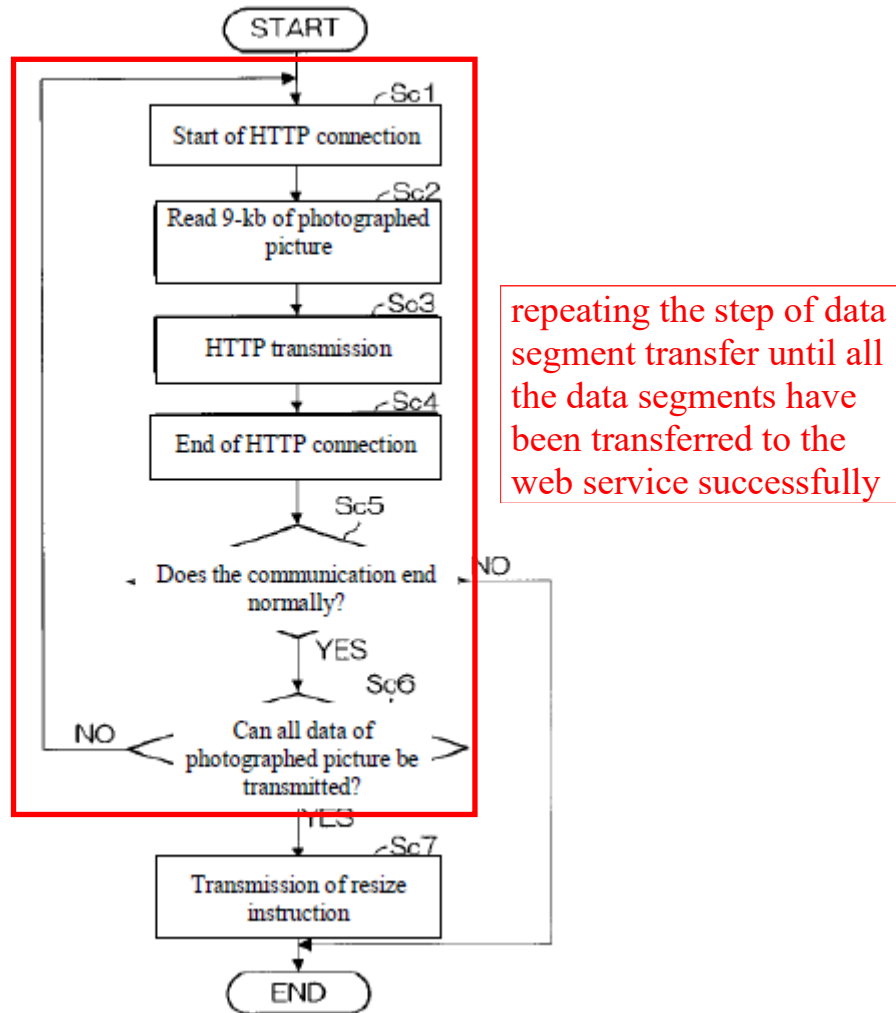


TTI-1016, FIG. 6.

Takahashi further explains that “a HTTP connection is initiated (Sc1), and 9-kb of photograph taken is read by the image dividing unit 17 ( Sc2). Next, once the read image data (9 kb) is transmitted to the image storage server 4 through HTTP (Sc3), and the transmission is completed, the HTTP connection is ended (Sc4).” TTI-1016, ¶(0071). If “all the data are not transmitted, the process returns to step Sc1, the processing described above is repeated, and the image data is transmitted 9 kb at a time to the image storage server 4 through HTTP.” TTI-1016, ¶(0072). As shown in FIG. 8, Takahashi repeats *the step of data segment transfer, until all*

*the data segments have been transferred to the web service successfully.* Here again, Takahashi's "mobile phone 1 adopts a sequential transmission rule for transmitting a divided file one at a time and transmitting the next divided file after waiting for the response of the reception completion on the image storage server 4 side" ("*the step of receiving the reply message*"). TTI-1016, ¶(0058); *supra* 1[g]; TTI-1003, ¶218.

(FIG. 8)

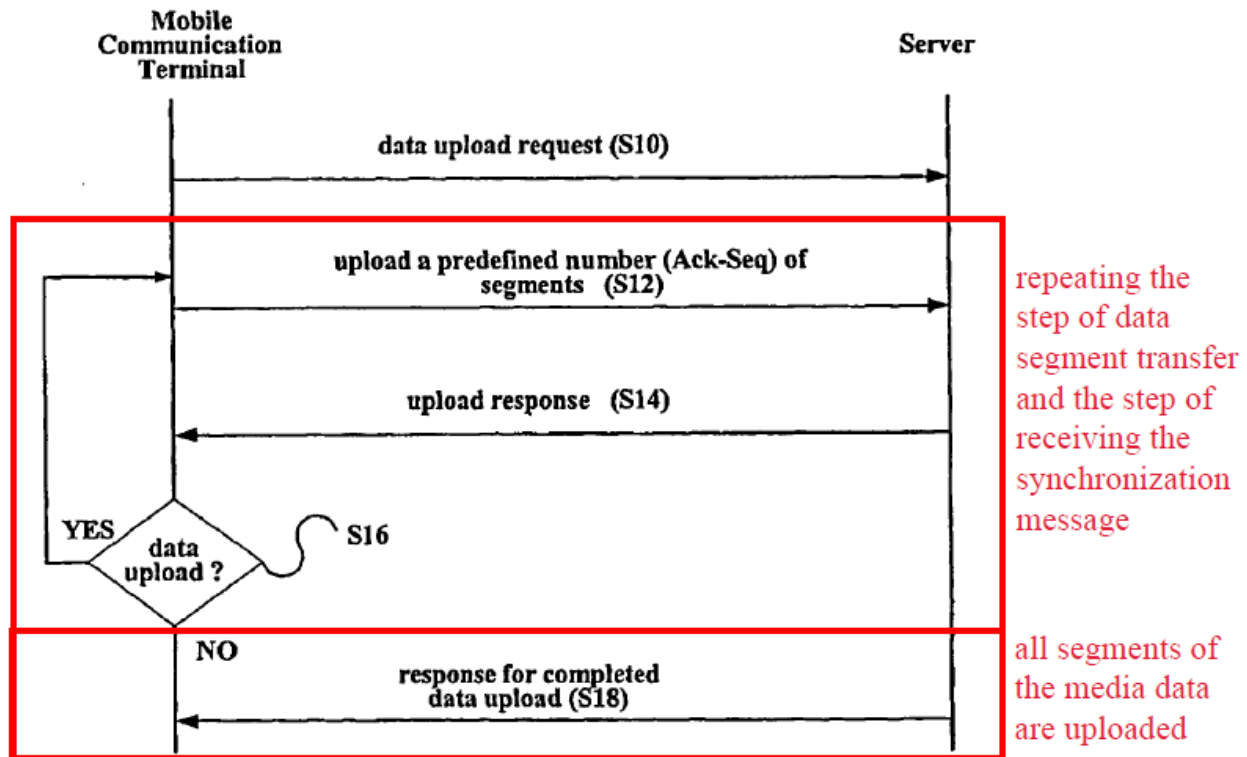


TTI-1016, FIG. 8.

Na confirms that transmitting data segments includes repeating the step of data segment transfer and the step of receiving the synchronization message, until all the data segments have been transferred to the web service successfully. TTI-1006, ¶¶[0079]-[0080], [0082]-[0083], [0097]. Na's FIG. 10 illustrates step S12 for uploading a group of K segments (*"the step of data segment transfer"* per 1[f1]) and S14 for receiving an acknowledgement response message (*"the step of*

*receiving the reply message*” per 1[g]) are repeated until all segments of the media data are uploaded. TTI-1003, ¶¶219-220.

**FIG. 10**



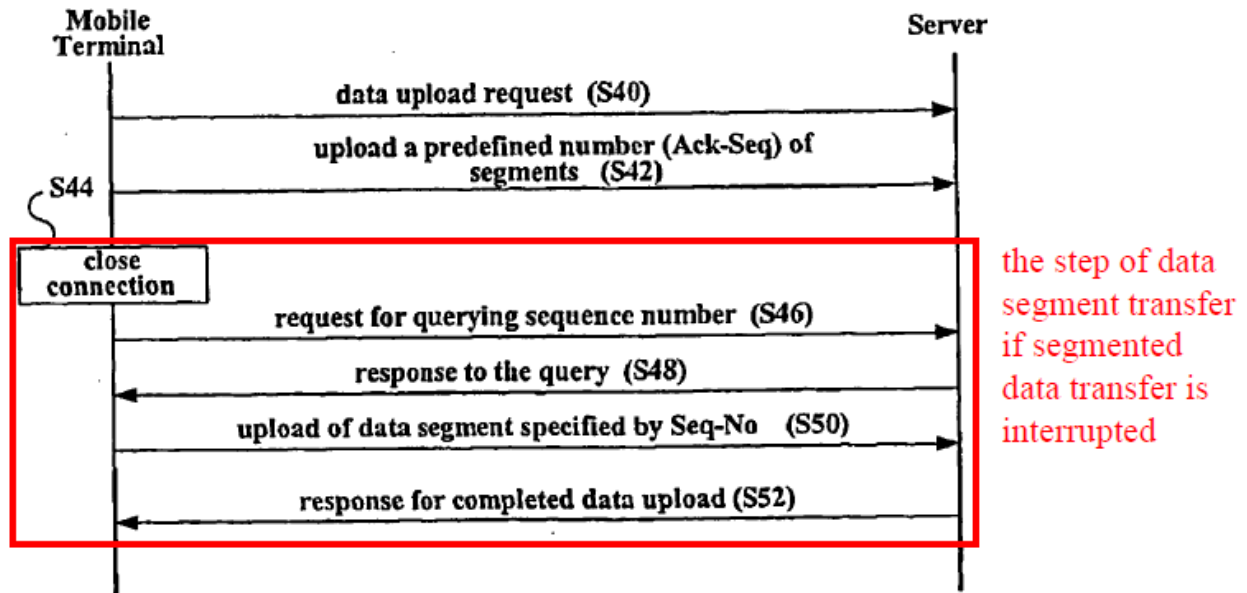
TTI-1006, FIG. 10.

**Element 1[i].**

As discussed above in §III.B.4, a POSITA would have been motivated and found it obvious and straightforward to implement Takahashi’s mobile phone application software (*see* 1[a]) based on Na’s teachings to beneficially allow “an upload of media data stopped by an upload error to be resumed from the point where the error occurred,” as taught by Na. TTI-1006, ¶¶[0006], [0008], [0010]; TTI-1016, ¶(0071); TTI-1003, ¶221.

Na discloses various techniques for “resuming an upload of media data” “stopped by an upload error,” “when the communication connection is reestablished,” or “stopped by a closed connection” (“*segmented file transfer is interrupted*”). TTI-1006, ¶¶[0084], [0091], [0139]-[0157], FIGS. 11-12, 24-25. For example, Na’s FIG. 12 shows “resuming an upload of media data when the communication connection is reestablished” if “the mobile communication terminal 10 gets disconnected from the application server 40 and as a result an upload error occurs” (“*if segmented file transfer is interrupted*”), which includes steps S46-S50 (“*the step of data segment transfer*”) as discussed in more detail below in 1[j]-1[l]. TTI-1006, ¶¶[0036], [0091]-[0097]; TTI-1003, ¶222.

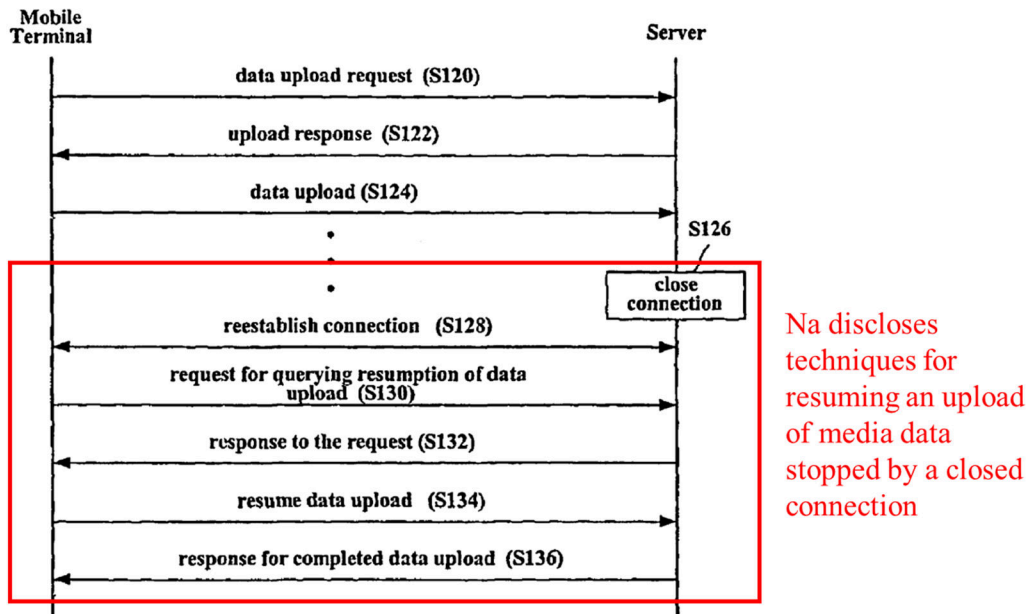
**FIG. 12**



TTI-1006, FIG. 12.

As another example, Na's FIG. 25 illustrates "resuming an upload of media data stopped by a user" ("*segmented file transfer is interrupted*"), which includes steps S128-S136 ("*the step of data segment transfer*"), as discussed in more detail below in elements 1[j]-1[l]. TTI-1006, ¶¶[0049], [0148]-[0157]; TTI-1003, ¶223.

FIG. 25



TTI-1006, FIG. 25.

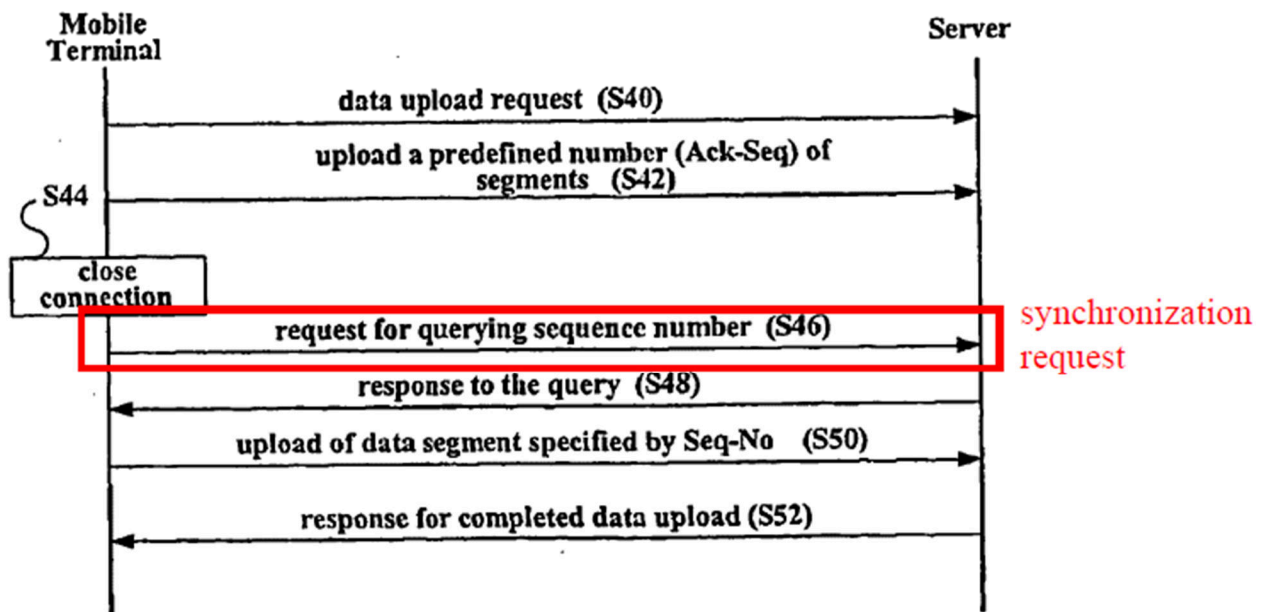
The above operations are performed by the mobile phone's software application ("*software module*"). TTI-1006, 2:12-16; TTI-1016, ¶¶(0053), (0055)-(0056), FIGS. 2-3; TTI-1003, ¶224; *supra* 1[a].

#### Element 1[j].

Na's FIG. 12 illustrates "resuming an upload of media data when the communication connection is reestablished" ("*after a network connection to the web service is restored*"). TTI-1006, ¶¶[0036], [0091]. Na's FIG. 12 step S46 shows that "the mobile communication terminal 10 **sends a request message querying the sequence number of the segment from which the upload should be resumed** to the application terminal," ("*a synchronization request to the web*

*service... corresponding to asking the web service about the segmented file data the web service has successfully received*") "the request message comprising the command 140 and the header 142 including a content type and a **message ID** as shown in FIG. 6." TTI-1006, ¶¶[0036], [0091], [0094]; TTI-1003, ¶225.

**FIG. 12**

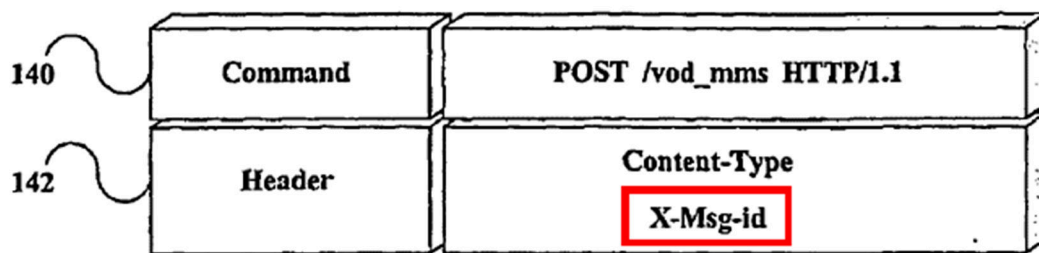


TTI-1006, FIG. 12.

As shown in FIG. 6, the message ID in the header 142 is "a message ID assigned to the media data to be uploaded (X-Msg-id, e.g., a **terminal number** and upload time information **in the form of MIN\_MMDDH HMMSS**)," and is in a similar format as Takahashi's file name that includes the user ID discussed above in 1[e]. TTI-1006, ¶¶[0063], [0078]; TTI-1016, ¶¶(0010), (0013), (0032), (0035), (0056), (0066), (0070)-(0071), (0073), (0084), (0087), FIG. 5; TTI-1003, ¶226;

TTI-1020, 4:7-10, 4:17-19, 2:60-63, 9:10-13, 12:46-49; TTI-1021, 7:39-45; TTI-1022, 3:25-27. The request message is “*for said user identifier*” because the request message is used to query the data previously uploaded that are associated with the “terminal number” or user ID (“*said user identifier*”) to determine “the sequence number of the segment from which the upload should be resumed” for the same user identifier. TTI-1006, ¶[0094]; TTI-1003, ¶226.

## FIG. 6

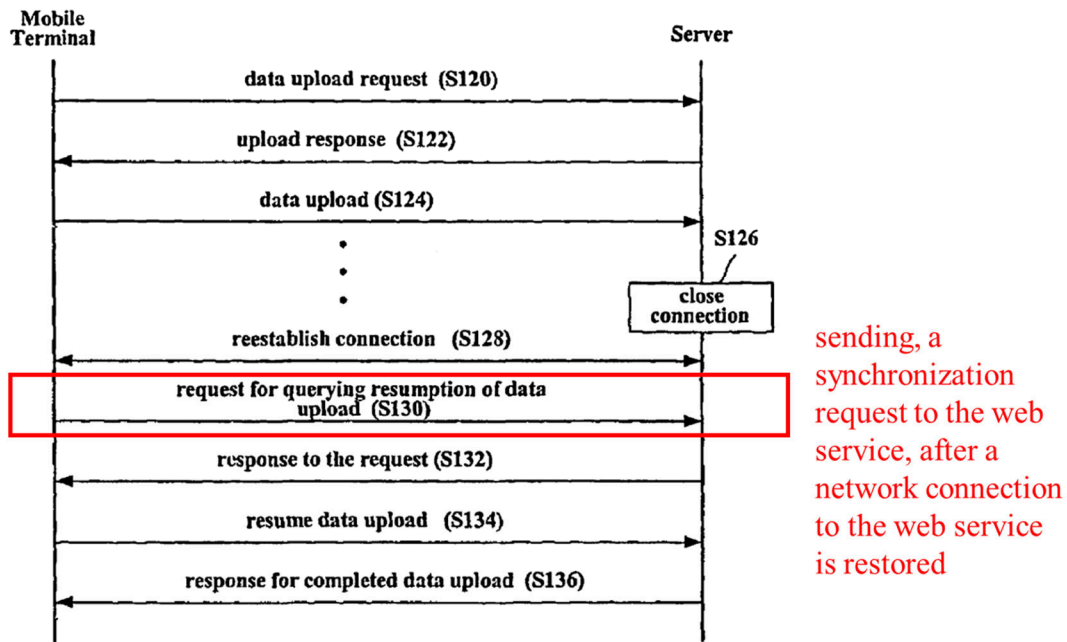


TTI-1006, FIG. 6.

As another example shown in FIG. 25, Na discloses that, after “the connection between them gets reestablished” in S128 (“*after a network connection to the web service is restored*”), “the mobile communication terminal 10 issues a request message for querying where to resume the upload” to the server in S130 (“*sending, a synchronization request to the web service*”). TTI-1006, ¶¶[0152]-[0153]. The “request message comprises a header including information **querying whether a portion of the message body that was previously uploaded exists in the application server**” (“*corresponding to asking the web service about the segmented file data the web service has successfully*

*received*”). TTI-1006, ¶[0153]; TTI-1003, ¶227.

**FIG. 25**

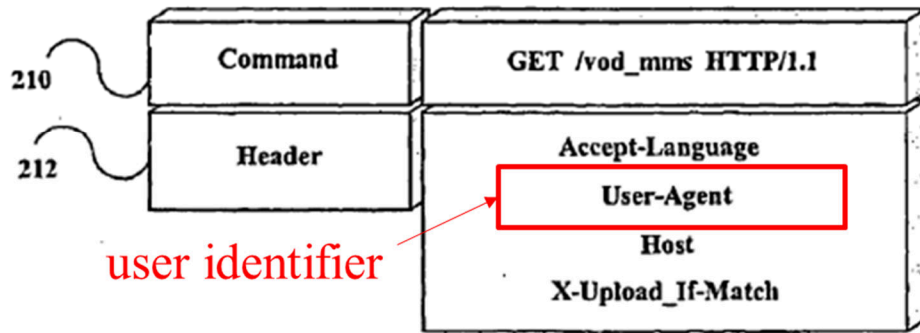


TTI-1006, FIG. 25.

The “request message for querying where to resume the upload” is “*for said user identifier*” because, “[a]s shown in FIG. 16, the request message comprises a header including information querying whether a portion of the message body that was previously uploaded exists in the application server 40 (S130)” such as the “user agent information (User-Agent, e.g., SKT/XXX200)” (another example of “*said user identifier*”) and the request message is used to query the data “previously uploaded” that are associated with the “user agent information” (“*said user identifier*”) to determine “where to resume the upload” the data for the same user identifier. TTI-1006, ¶[0153], [0116]-[0117], [0119], FIG. 16; TTI-1003,

¶¶228-229; TTI-1019, 8, 89.

**FIG. 16**



TTI-1006, FIG. 16.

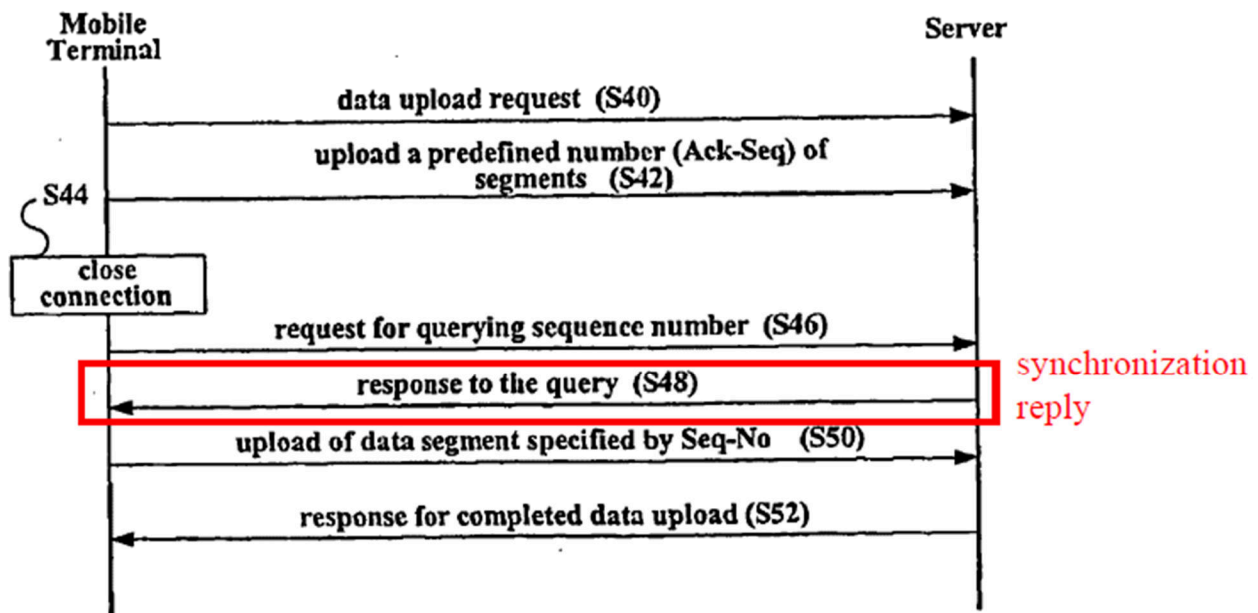
Na discloses performing operations “in accordance with... the **HTTP 1.1** protocol,” and thus the synchronization request is sent “*from the application layer level*” pursuant to the HTTP protocol. TTI-1003, ¶230; TTI-1006, ¶¶[0003]-[0004]; TTI-1019, 1; TTI-1007, 7, 76, 249-251; *supra* 1[c1]-1[c2]; TTI-1002, 115.

A POSITA would have been motivated and found it obvious and straightforward to implement Takahashi’s mobile phone to send to the server a synchronization request querying “the sequence number of the segment from which the upload should be resumed” or “the range of the message body that has been successfully uploaded” so that the mobile phone “is capable of knowing what part of the message body is to be uploaded additionally,” as taught by Na. TTI-1006, ¶¶[0012]-[0013], [0022], [0094]; TTI-1003, ¶231; *supra* §III.B.4.

**Element 1[k].**

Na discloses that receiving “the request message querying the sequence number” (“*synchronization request*” per 1[j]), “the application server 40 issues a response message for the query” (“*a synchronization reply back from the web service*”), shown as step S48 in Na’s FIG. 12. TTI-1006, ¶[0095]; TTI-1003, ¶232.

**FIG. 12**



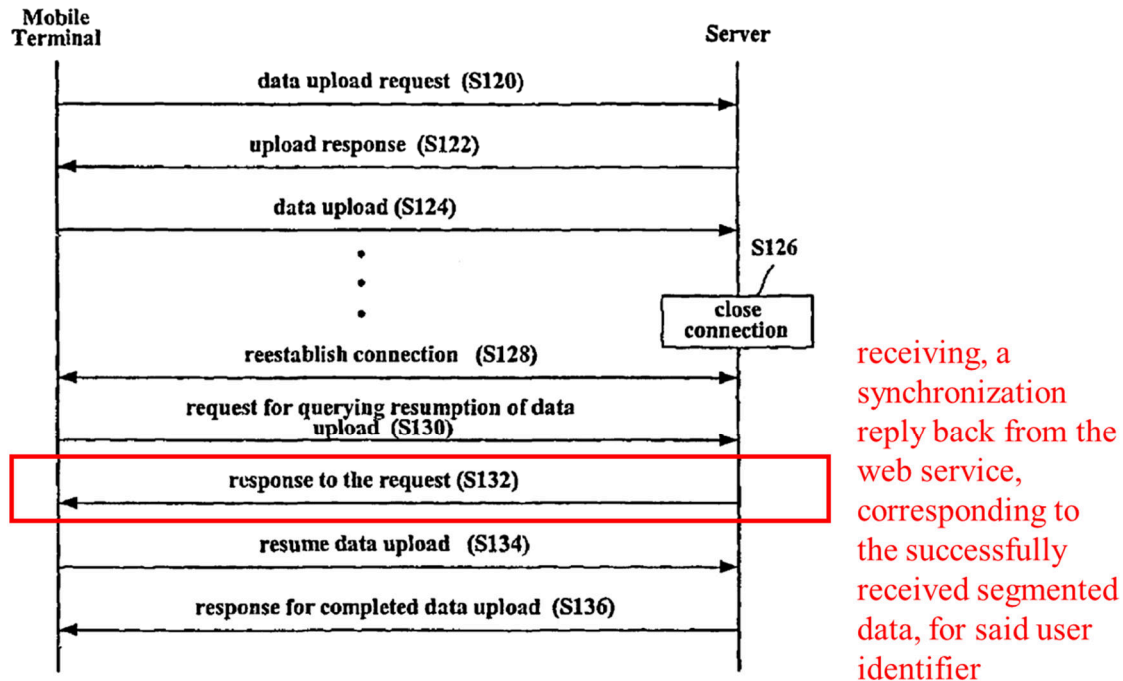
TTI-1006, FIG. 12.

Na further discloses that “the response message comprises the status line 150 and the header 152 including a connection status, a cache control, and the sequence number of the segment from which the upload should be resumed.” TTI-1006, ¶[0095]. The “sequence number of the segment from which the upload should be resumed” directly indicates (e.g., the last successfully received

segmented data) or indirectly indicates (e.g., the next data segment following the last successfully received segmented data) and thus corresponds to the successfully received segmented file data. TTI-1003, ¶233.

As another example, Na's terminal receives "a response message... as shown in FIG. 21 or FIG. 22" from the server ("***receiving, a synchronization reply back from the web service***"). TTI-1006, ¶[0154]. The response message includes "a message header including the range of data that has been uploaded already," such as "the range of the **content that has been uploaded successfully** before the upload error" ("***corresponding to the indicated segmented file data said web service has successfully received***"). TTI-1006, ¶[0154], [0132], [0134], FIGS. 21-22, 25; TTI-1003, ¶235.

FIG. 25



TTI-1006, FIG. 25.

The response message is “*for said user identifier*” because (1) the response message is directly in response to the request message that is “*for said user identifier*” (*supra* 1[j]); and (2) the response message itself identifies “the sequence number of the segment from which the upload should be resumed” or “the range of data that has been uploaded already” that are associated with the “user agent information” or user ID (“*said user identifier*”) so as to inform the terminal “where to resume the upload” for the user identifier. TTI-1006, ¶¶[0095], [0153]-[0155],[0119]; *supra* 1[e], [1j]; TTI-1003, ¶¶234, 236.

Additionally, Na discloses performing operations “in accordance with... the

HTTP 1.1 protocol,” and thus the synchronization reply is sent “*at the application layer level*” pursuant to the HTTP protocol. TTI-1003, ¶237; TTI-1006, ¶¶[0003]-[0004]; TTI-1019, 1; TTI-1007, 7, 76, 249-251; TTI-1002, 115; *supra* 1[c1]-1[c2].

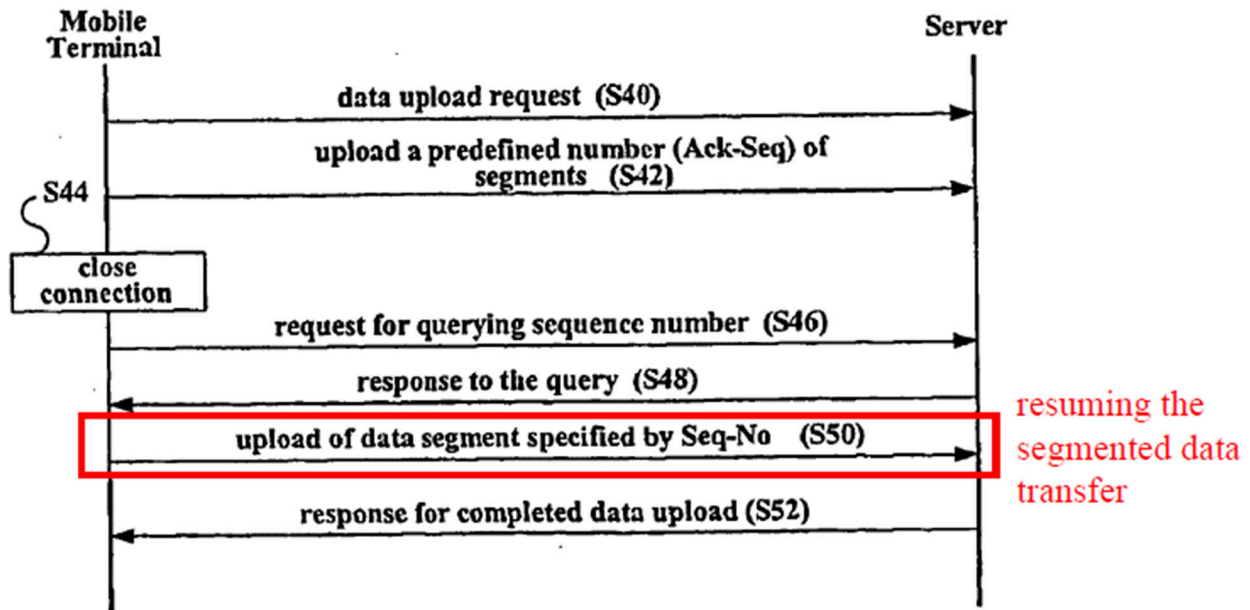
A POSITA would have been motivated and found it obvious and straightforward to implement Takahashi’s mobile phone to receive from the server a synchronization reply corresponding to “the sequence number of the segment from which the upload should be resumed” or “the range of the message body that has been successfully uploaded” so that the mobile phone “is capable of knowing what part of the message body is to be uploaded additionally,” as taught by Na.

TTI-1006, ¶¶[0012]-[0013], [0022], [0094]; *supra* §III.B.4, 1[j]; TTI-1003, ¶238.

**Element 1[i].**

Na’s terminal “**resumes the upload with reference to** the sequence number included in **the response message** from the application server” (“*resuming, the segmented file transfer based on the synchronization reply received*”), as shown in step S50 of FIG. 12. TTI-1006, ¶[0096]. The terminal resumes the upload from “the sequence number included in the response message from the application server” for the resumed media data upload to be completed normally. TTI-1006, ¶[0097]; TTI-1003, ¶239.

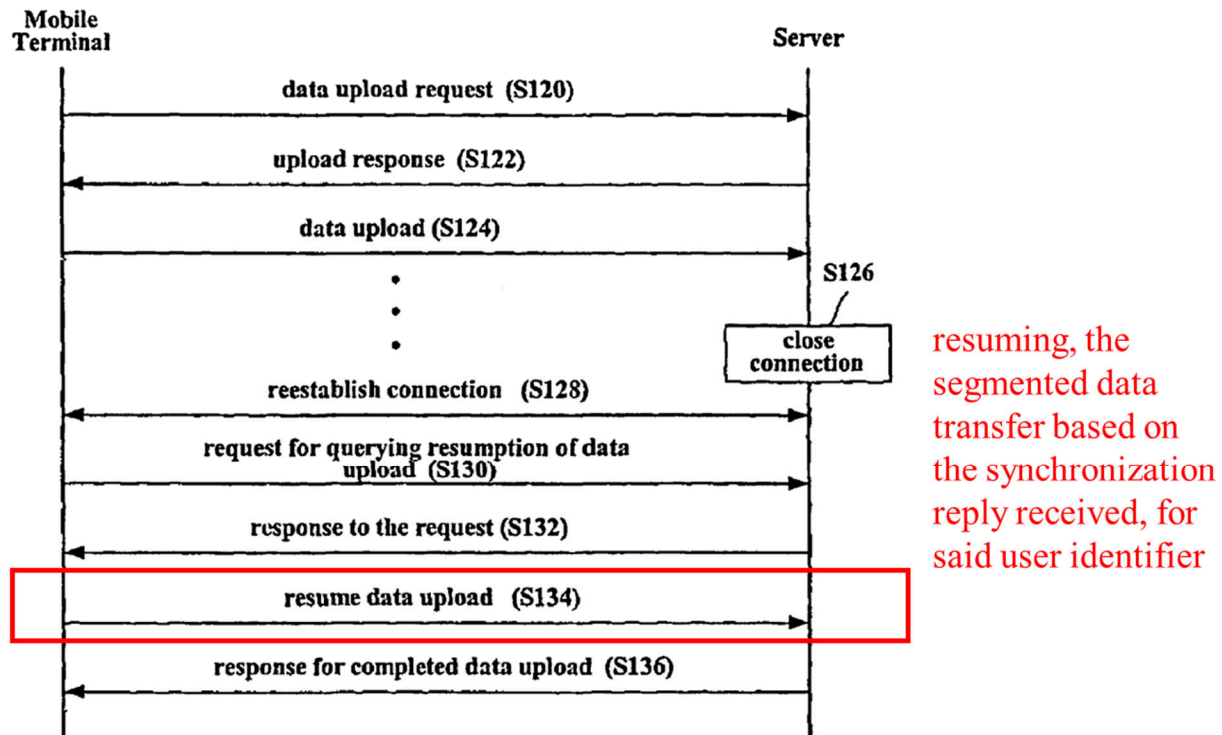
**FIG. 12**



TTI-1006, FIG. 12.

As another example, Na's terminal "begins **uploading a portion of the media data that needs to be uploaded additionally (S134)**" ("*resuming, the segmented file transfer*") in response to "receiv[ing] **the response message** from the application server" (*based on the synchronization reply received*"), as shown in FIG. 25. TTI-1006, ¶[0156]; TTI-1003, ¶[240].

FIG. 25



TTI-1006, FIG. 25.

The terminal resuming the upload is “*for said user identifier*” because the upload is associated with the user identifier, and the processes described with respect to FIGS. 12 and 25 are for resuming the upload of data for the user identifier. TTI-1006, ¶¶[0148], [0153]-[0155], [0119]; *supra* 1[e], [1j], 1[k]; TTI-1003, ¶¶239, 241.

As discussed in §III.B.4, a POSITA would have been motivated and found it obvious and straightforward to incorporate Na’s teachings into Takahashi’s system to beneficially allow “an upload of media data stopped by an upload error to be

resumed from the point where the error occurred,” as taught by Na. TTI-1006, ¶¶[0008], [0010]; TTI-1003, ¶242.

**Element 1[m].**

Takahashi discloses that in “transmitting the divided image using HTTP, the generated file name” including the user ID “is also transmitted together” (“*transferring a data segment along with said user identifier*”) and “when carrying out division and transmission of the image file from the mobile phone 1, the information showing the order of the file such as division number (Ex. 5/10, or the like)” (“*a segment identifier*”) “may be attached and sent, if needed” “in the HPPT [*sic*] transmission” that includes a division of the data (“*transferring a data segment along with... said segment identifier*”). TTI-1016, ¶¶(0071), (0073), (0058); *supra* 1[d]-1[f]. In Takahashi, “the image data is received through HTTP in the image storage server.” TTI-1016, ¶(0051), (0068). Takahashi repeats the transfer of divided image data along with the user ID and division number to the image storage server until “the transmission of all the divided image data is completed” (“*receiving, at the web service, all the data segments along with said user identifier and segment identifiers for said file transfer*”). TTI-1016, ¶¶(0068), (0071)-(0072); *supra* 1[h]; TTI-1003, ¶243.

Further, Na discloses transmitting the request messages (“*data segments along with said user identifier and segment identifiers for said file transfer*”)

“until all segments of the media data are uploaded” and then “the application server 40 sends a response message” to confirm that “the media data upload is completed normally” (“*receiving, at the web service, all the data segments along with said user identifier and segment identifiers for said file transfer*”). TTI-1006, ¶¶[0082]-[0083], FIG. 10; *supra* 1[h]; TTI-1003, ¶244.

In another example, Na discloses repeating the exchange of the request messages (“*data segments along with said user identifier and segment identifiers for said file transfer*”) and response messages until “the resumed upload of the media data is completed normally” (“*until all the data segments have been transferred to the web service successfully*”). TTI-1006, ¶¶[0124], [0157]; TTI-1003, ¶245. The server “sends a response message comprising .... the upload result report as shown in FIG. 23” to confirm “a resumed data upload is completed successfully” (“*receiving, at the web service, all the data segments along with said user identifier and segment identifiers for said file transfer*”). TTI-1006, ¶¶[0047] [0135], [0147], FIGS. 23, 25; *supra* 1[h].

#### **Element 1[n].**

Takahashi’s server “sequentially saves the divided image data to store them as one image data” and stores “the received image data in association with the user ID” (“*storing, at the web service, the received file data segments for said user identifier*”). TTI-1016, Abstract, ¶¶(0050)-(0053), (0060)-(0064), (0076)-(0082),

FIG. 1. Further, Na discloses that “upload[ed] particular media data” are “stored in a memory thereof to the application server” (“*storing, at the web service, the received file data segments for said user identifier*”). TTI-1006, ¶¶[0052], [0111]. Na discloses, during the upload, “the data being uploaded” is stored in “a temporary file.” TTI-1006, ¶¶[0068], [0098], [0102], [0132], [0136]; TTI-1003, ¶246.

A POSITA recognized or found obvious that *segment identifiers* (e.g., division/sequence numbers or unique IDs per 1[d]) is stored with the *received file data segments* because the segment identifiers are used “for the server to match the previously uploaded data and data to be newly uploaded” and to determine “whether a portion of the media data that has been already uploaded exists in the application server.” TTI-1006, ¶¶[0012]-[0113], [0008], [0010]; TTI-1016, ¶¶(0050)-(0053), (0060)-(0063), (0076)-(0082); *supra* 1[d]; TTI-1003, ¶247. As Dr. Traynor explains, it is important for the server to store such segment identifiers to perform the above-mentioned matching and determination to achieve Na’s goal of “improving upload efficiency and speed,” and to respond to the mobile phone’s “request message for querying where to resume the upload.” *Id.*; TTI-1006, ¶¶[0018], [0020]-[0021], [0072]-[0073], [0094]-[0095], [0131], [0133]; TTI-1016, ¶¶(0050)-(0053), (0060)-(0063), (0076)-(0082).

The received file data segments and segment identifiers are stored “*for said*

*user identifier*” because the interrupted upload is associated with said user identifier. *Supra* 1[d]-1[e]; TTI-1003, ¶248.

**Element 1[o].**

Takahashi’s server receives divided image data of an image from the mobile phone (“*the segmented data received from the Bluetooth enabled mobile device*”), “publish[es] the image on a web page,” and makes it available for later download (“*making available, at the web service,... for public or private consumption*”). TTI-1016, ¶¶(0050)-(0053), (0057), (0060)-(0063), (0075)-(0082), FIG. 1; TTI-1003, ¶249.

**(b) Claim 2**

The Takahashi-Na and Takahashi-Na-Lind combinations render obvious claim 2 for similar reasons as discussed above for 1[c2]. TTI-1003, ¶¶250-253.

**(c) Claim 3**

Takahashi’s mobile phone application transfers “**image data**” (“*said file data comprises... image data*”). TTI-1016, Abstract, ¶(0005). Takahashi’s image data can be captured with a **digital** camera. TTI-1016, ¶¶(0002)-(0003). When describing image size and data size, Takahashi uses the unit “KB”/“kb” which stands for kilobytes. TTI-1016, ¶¶(0003), (0051), (0057), (0071)-(0073), (0080); TTI-1003, ¶254.

Additionally, Na’s uploaded data (“*file data*”) is “media data,” which can be

transmitted in a message body of a request message, with a “content length given in **bytes** (Contents-Length, e.g., **500**),” and can be “defined within 64 **bytes** in accordance with the message body format of the MMS 1.0 specification.” TTI-1006, ¶¶[0001]-[0002], [0062]-[0063], [0116]-[0118], [0123]; TTI-1003, ¶255.

Takahashi-Na transferred “***digital data***” because it is common knowledge that “bytes” is a hallmark metric of “digital data,” and Takahashi and Na each discloses the media data size is measured in “bytes.” TTI-1003, ¶256.

**(d) Claim 4**

Based on reasons discussed above in 1[e] and Takahashi’s disclosure that “the user ID 30 is stored in association with a device number specific to the mobile phone” (“***user-device-information***”) and “the image storage server 4 searches a user ID based on the device number,” it would have been obvious that Takahashi’s user ID is either a “***user-name***” used to log in to the server or “***user-device-information***” stored in the image storage server. TTI-1016, ¶¶(0010), (0013), (0032), (0035), (0066), (0070), (0064), (0066), FIG. 5; TTI-1003, ¶257.

Moreover, Na’s request message header includes “a message ID assigned to the media data to be uploaded (X-Msg-id, e.g., **a terminal number** and upload time information **in the form of MIN\_MMDDH HMMSS**)” or “**user agent information (User-Agent, e.g., SKT/XXX200)**” (both examples of “***user identifiers***” that are “***user-device-information***”). TTI-1006, ¶¶[0063], [0078],

[0116]-[0117], FIG. 15; TTI-1020, 4:7-10, 4:17-19, 2:60-63, 9:10-13, 12:46-49;  
TTI-1021, 7:39-45; TTI-1022, 3:25-27; TTI-1019, 8, 89; TTI-1003, ¶¶258-259.

**(e) Claim 5**

Takahashi-Na uses HTTP (*supra* 1[c1]), which is “*the layer 7 in the OSI layer stack, or the application layer of the Internet protocol suite*” because, as Dr. Traynor explains, it was well known before the Critical Date that the “*application layer*” was “*layer 7 in the OSI layer stack,*” TTI-1016, Abstract, ¶¶(0005)-(0008), (0020)-(0030), (0042)-(0043), (0049); TTI-1006, ¶[0052]; TTI-1003, ¶260; TTI-1015, 2:58-4:40 (describing the “OSI model”); TTI-1007, 11-13, 76.

**(f) Claims 6-9**

The Takahashi-Na and the Takahashi-Na-Lind combinations render claims 6-9 obvious in the same manner as explained above for the corresponding claim elements listed in the following table. TTI-1003, ¶¶261-297.

| Claim Element | Corresponding Claim Element |
|---------------|-----------------------------|
| 6[pre]        | 1[pre]                      |
| 6[a]          | 1[a]                        |
| 6[b]          | 1[b]                        |
| 6[c]-6[g]     | See analysis below          |
| 6[h]          | 1[j]                        |
| 6[i]-6[k]     | See analysis below          |

|      |                    |
|------|--------------------|
| 6[l] | 1[d]               |
| 6[m] | 1[e]               |
| 6[n] | 1[f]               |
| 6[o] | 1[m]               |
| 6[p] | 1[n]               |
| 6[q] | 1[o]               |
| 6[r] | See analysis below |
| 7    | 3                  |
| 8    | 4                  |
| 9    | 5                  |

**Element 6[c].**

For reasons discussed above in 1[c1], Takahashi’s mobile phone application segments the media data into multiple divided files (“*segmenting said file into a first data segment and a second data segment*”) to transmit “all the divided image data.” TTI-1016, ¶¶(0068), (0058); TTI-1003, ¶264.

There were multiple different ways to format each “divided file” (“*data segment*”) “as an HTTP request.” TTI-1003, ¶265; *supra* 1[c1]. For example, Na teaches: (1) a request message preceding each upload of divided image data (TTI-1006, ¶¶[0052]-[0053]; TTI-1003, ¶265) and (2) a request message including divided image data in the message body for each upload of divided image data (TTI-1006, ¶¶[0112]-[0113], [0039], [0116]-[0118], [0151], FIG. 15; TTI-1003,

¶266).

Additionally, the image data (“*file*”) is segmented “*at an application layer level*” for upload with or in multiple “request messages” pursuant to the HTTP protocol. TTI-1003, ¶267; TTI-1006, ¶¶[0003]-[0004], [0052], [0111]; TTI-1019, 1; TTI-1007, 7, 76, 249-251; TTI-1002, 115; *supra* 1[c1].

**Element 6[d].**

Each of Takahashi’s divided image file (“*data segments*”) includes image data (“*file data*”) and a file name (“*data associated with multimedia data*”). TTI-1016, ¶¶(0010), (0013), (0032), (0035), (0066), (0070)-(0071), (0073), FIG. 5; TTI-1003, ¶268; *supra* 1[c3]. Further, Na transfers request messages including a header with content type information and content length information (“*data associated with said file data*”), as shown in FIG. 2 and FIG. 15, along with data segments of the media data (“*file data*”). TTI-1006, ¶¶[0053], [0062]-[0063], [0112]-[0113], [0116]-[0118], FIG. 15; TTI-1003, ¶¶269-271. Accordingly, a first data block (“*first data segment*”) includes a file name and a header with content type information and content length information (“*data associated with said file data*”), and a second data block (“*second data segment*”) includes image data (“*file data*”). *Id.*

**Elements 6[e1]-6[e3].**

The reasons discussed above for 1[c1], 1[e], 1[f], 1[g], and 6[c] apply to

each data segment and therefore apply to “*said first data segment*” of 6[e1]-6[e3].

TTI-1003, ¶¶272-274.

**Element 6[f1]-6[f3].**

The reasons discussed above for 1[c1], 1[e], 1[f], 1[g], and 6[c] apply to each data segment and therefore apply to “*said second data segment*” of 6[f1]-6[f3]. TTI-1003, ¶¶275-277.

**Element 6[g].**

The reasons discussed above for 1[i] and 6[c] apply to interruptions for “*said second data segment.*” TTI-1003, ¶278.

**Element 6[i].**

The reasons discussed above for 1[k] and 6[c] apply to “*said second data segment.*” TTI-1003, ¶280.

**Element 6[j].**

For reasons discussed above in 1[l] and 6[i], the mobile phone resumes the upload *by transferring remaining data of the second data segment* because Na’s “sequence number” indicates “from which the upload should be resumed,” therefore making it possible to “resume **the stopped upload from the point at which the previous upload was stopped**” and “the server to match the previously uploaded data and **data to be newly uploaded**” (“*remaining data of the second data segment*”), and preventing “the whole data to be uploaded from scratch” or

“[t]he upload of the whole media... be conducted again despite the fact that a portion of the media data has been already uploaded to the server.” TTI-1006, ¶¶[0095]-[0096], [0007]-[0008]. As another example, in FIG. 25, Na’s terminal “begins **uploading a portion of the media data that needs to be uploaded additionally**” (“*resuming, the segmented data transfer by transferring remaining data of the second data segment*”) in response to “receiv[ing] the response message from the application server” (“*based on the synchronization reply received*”). TTI-1006, ¶[0156]; TTI-1003, ¶¶281-283; *supra* §III.B.4. **Element 6[k].**

When resuming an interrupted upload, Na’s terminal sends “*remaining data of the second data segment*” (thus, *creating one or more data segments, wherein the one or more data segments comprise remaining data of the second data segment*). TTI-1006, ¶¶[0095], [0007]-[0008], [0156]; *supra* 15[j]. As Dr. Traynor explains, the *create[ed] one or more data segments* included the *remaining data of the second data segment* because retransmitting successfully received data is inefficient. TTI-1003, ¶¶284, 286; *supra* §III.B.4. Further, because this claim limitation recites creating “one or more” segments, simply transmitting the entirety of the “*remaining data of the second data segment*” as one data segment satisfies the claim. *Id.*

Na discloses performing operations “in accordance with... the **HTTP 1.1**

protocol,” and thus the one or more data segments are created *at the application layer level* because “*the second data segment*” and thus “*the remaining data of the second data segment*” are created *at the application layer level* per 15[c1]. TTI-1006, ¶[0052]; TTI-1003, ¶285.

**Element 6[r].**

Takahashi’s server “sequentially saves the divided image data to store them as one image data” and stores “the received image data in association with the user ID” specific to a user of the server (“*one or more portions of the data correspond to a particular user of the web services*”). TTI-1016, Abstract, ¶¶(0010), (0013), (0032), (0035), (0050)-(0053), (0060)-(0064), (0066), (0070), (0076)-(0082), FIG. 1; *supra* 1[e], 6[p]; TTI-1003, ¶293.

Further, Na discloses that a terminal number in the form of MIN (mobile identification number) or “user agent information (User-Agent, e.g., SKT/XXX200)” (“*user identifier*”) included in the request messages that carry one or more portions of the media data (“*one or more portions of the data*”) *corresponds to a particular user of the web services*” of a particular upload. TTI-1006, ¶¶[0063], [0078], [0116]-[0117], FIG. 15; *supra* 1[e], 6[p]; TTI-1003, ¶294.

**(g) Claims 10-14**

The Takahashi-Na and Takahashi-Na-Lind combinations render claims 10-14 obvious in the same manner as explained above for the corresponding claim

elements listed in the following table. TTI-1003, ¶¶298-321.

| Claim Element | Corresponding Claim Element |
|---------------|-----------------------------|
| 10[pre]       | 1[pre]                      |
| 10[a]         | 1[a]                        |
| 10[b]         | 1[b]                        |
| 10[c1]        | 1[c1]                       |
| 10[c2]        | 1[c2]                       |
| 10[d]         | 1[d]                        |
| 10[e]         | 1[e]                        |
| 10[f]         | 1[f]                        |
| 10[g]         | 1[g]                        |
| 10[h]         | 1[h]                        |
| 10[i]         | 1[i]                        |
| 10[j]         | 1[j]                        |
| 10[k]         | 1[k]                        |
| 10[l]         | 1[l]                        |
| 10[m]         | 6[o]                        |
| 10[n]-10[o]   | See analysis below          |
| 11            | 2                           |
| 12            | 3                           |
| 13            | 4                           |
| 14            | 5                           |

**Element 10[n].**

Takahashi's mobile phone divides the data and transmits each data division in a divided file ("*data segment*") with a division number ("*segment identifier*") and a file name including the user ID ("*user identifier*") so that "when the divided file (image data) with the same file name has already been received [by the image storage server], the divided file is written as an addition (combination) to the existing file" ("*reassembling the received file data segments automatically using... a user identifier... at the web service*"). TTI-1016, ¶¶(0058), (0071), (0073); *supra* 1[d]-1[f]; TTI-1003, ¶313.

A POSITA would have found it obvious to reassemble the divided image file using segment identifiers. For example, Na discloses that "by providing the sequence number of the segment to be uploaded," it is possible "for the server to match the previously uploaded data and data to be newly uploaded" and allow "an upload of media data stopped by an upload error to be resumed from the point where the error occurred." TTI-1006, ¶¶[0008], [0010]. The server uses identifiers, both the sequence number and the user identifier associated with the particular upload, to reassemble the received file data segments automatically in the correct order when storing the media data for the user identifier associated with the particular upload. TTI-1003, ¶314.

Na's unique ID (another example of “*segment identifier*”) and user agent information (another example of “*user identifier*”) “allow the application server 40 to match one portion of the message body that has been already uploaded and the other portion of the message body to be uploaded in the case where the upload of the media data is resumed.” TTI-1006, ¶[0112], [0012], [0022], [0116]-[0117]; *supra* 1[d]-1[e]. The server uses identifiers, both the sequence number and the user identifier associated with the particular upload, to reassemble the received data portions automatically in the correct order when storing the media data for the user identifier associated with the particular upload. TTI-1003, ¶315. Indeed, Na describes, using the segment identifiers, the server receives and reassembles portions of data in “the range of the message body that has been uploaded successfully before the upload error or upload cancellation” (e.g., “1-1000/1500”) and resumed remaining portions of the data in the range “1001-1500/1500,” “where 1500 is the size of the whole data,” for the user identifier associated with the particular upload. TTI-1006, ¶¶[0112], [0122], [0132].

Additionally, because Takahashi and Na both use HTTP, this process would have been performed “*at the application layer level.*” TTI-1003, ¶316; *supra* 1[c1].

**Element 10[o].**

In Takahashi-Na, given that the server has reassembled the received file data

segments per 10[n], a POSITA recognized and found obvious that some of the data made available would be a “reassembled data file” because a reassembled file would be more desirable than its constituent segments taken individually (e.g., a complete video is more beneficial than a particular frame and requires less latency in displaying), especially if a network connection with another device that requests downloading the file has a larger bandwidth than the one with the device that uploaded the file, which is often the case as corroborated by Ganesan. TTI-1024, ¶[0038] (disclosing “the downloading speed is at least k times faster than the uploading speed”); TTI-1003, ¶317; *supra* 1[o].

**(h) Claims 15-18**

The Takahashi-Na and Takahashi-Na-Lind combinations render claims 15-18 obvious in the same manner as explained above for the corresponding claim elements listed in the following table. TTI-1003, ¶¶322-346.

| Claim Element | Corresponding Claim Element |
|---------------|-----------------------------|
| 15[pre]       | 1[pre]                      |
| 15[a]         | 1[a]                        |
| 15[b]         | 1[b]                        |
| 15[c]         | 6[c]                        |
| 15[d]         | 6[d]                        |
| 15[e1]        | 6[e1]                       |

|        |       |
|--------|-------|
| 15[e2] | 6[e2] |
| 15[e3] | 6[e3] |
| 15[f1] | 6[f1] |
| 15[f2] | 6[f2] |
| 15[f3] | 6[f3] |
| 15[g]  | 6[g]  |
| 15[h]  | 1[j]  |
| 15[i]  | 6[i]  |
| 15[j]  | 6[j]  |
| 15[k]  | 6[k]  |
| 15[l]  | 1[d]  |
| 15[m]  | 1[e]  |
| 15[n]  | 1[f]  |
| 15[o]  | 1[m]  |
| 15[p]  | 10[n] |
| 15[q]  | 10[o] |
| 16     | 3     |
| 17     | 4     |
| 18     | 5     |

#### IV. DISCRETIONARY CONSIDERATIONS

##### A. The *Fintiv* Factors Favor Institution—§314(a)

The PTAB considers a set of non-exclusive factors (the *Fintiv* factors) in

determining whether to institute an IPR proceeding where there is parallel district court litigation. *Apple Inc. v. Fintiv, Inc.*, IPR2020-00019, Paper 11 (PTAB Mar. 20, 2020) (precedential). On June 21, 2022, Director Vidal issued “binding agency guidance” entitled “Interim Procedure for Discretionary Denials in AIA Post-Grant Proceedings with Parallel District Court Litigation.” TTI-1025 (hereinafter the “Interim Guidance”). This Interim Guidance presents “several clarifications... to the PTAB’s... application of *Fintiv* to discretionary institution where there is parallel litigation.” *Id.*, 2.

One clarification provided by the Interim Guidance specifies the effect of particular types of stipulations presented by petitioners on the Board’s application of the *Fintiv* framework. *Id.*, 3, 7-8. In particular, the Interim Guidance unequivocally states that “***the PTAB will not discretionarily deny institution*** of an IPR or PGR in view of parallel district court litigation where a petitioner stipulates not to pursue in a parallel district court proceeding the same grounds as in the petition or any grounds that could have reasonably been raised in the petition.” *Id.*, 7 (emphasis added) (citing *Sotera Wireless, Inc. v. Masimo Corp.*, IPR2020-01019, Paper 12 (PTAB Dec. 1, 2020)). In the present case, Petitioner presents a stipulation not to pursue in the parallel district court proceeding the same grounds as in the Petition or any grounds that could have reasonably been raised in the Petition upon institution. TTI-1026. Accordingly, based on the Interim Guidance,

institution of the Petition ***should not*** be discretionarily denied in view of the parallel district court proceeding. TTI-1025, 7.

**V. PAYMENT OF FEES – 37 C.F.R. §42.103**

Petitioner authorizes the Patent and Trademark Office to charge Deposit Account No. 06-1050.

**VI. MANDATORY NOTICES UNDER 37 C.F.R § 42.8(a)(1)**

**A. Real Party-In-Interest Under 37 C.F.R. § 42.8(b)(1)**

TikTok Inc., ByteDance, Inc., ByteDance, Ltd., ByteDance PTE, Ltd., and TikTok PTE, Ltd. are the real parties-in-interest.

**B. Related Matters Under 37 C.F.R. § 42.8(b)(2)**

Petitioner is not aware of any disclaimers, reexamination certificates, or petitions for *inter partes* review for the '260 patent. The '260 patent is the subject of a civil action *CellSpin Soft, Inc. v. ByteDance, Ltd. et al.*, Case No. 2:23-cv-496 (E.D. Tex.).

The '260 patent is in the same family as U.S. Patent Nos. 11,659,381 (“the '381 patent”), 11,234,121 (“the '121 patent”), 9,900,766 (“the '766 patent”), 8,756,336 (“the '336 patent”), 8,862,757 (“the '757 patent”), and 8,904,030 (“the '030 patent”). Petitioner is concurrently petitioning for *Inter Partes* Review of the '381, '121, '766, '336, '757, '260, and '030 patents.

U.S. Patent Application No. 18/193,686 is in the same patent family as the

'381, '121, '766, '336, '757, '260, and '030 patents and is currently pending before the Patent Office.

**C. Lead And Back-Up Counsel Under 37 C.F.R. § 42.8(b)(3)**

| Lead Counsel                                                                                                                                                                                                                         | Backup counsel                                                                                                                                                                                                                                            |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| W. Karl Renner, Reg. No. 41,265<br>Fish & Richardson P.C.<br>60 South Sixth Street<br>Minneapolis, MN 55402<br>Tel: 202-783-5070<br>Fax: 877-769-7945<br>Email: <a href="mailto:IPR50048-0059IP1@fr.com">IPR50048-0059IP1@fr.com</a> | Kim H. Leung, Reg. No. 64,399<br>Baile Xie, Reg. No. 78,380<br>Fish & Richardson P.C.<br>60 South Sixth Street<br>Minneapolis, MN 55402<br>Tel: 202-783-5070<br>Fax: 877-769-7945<br><a href="mailto:IPR50048-0059IP1@fr.com">IPR50048-0059IP1@fr.com</a> |

**D. Service Information**

Please address correspondence/service to the above-listed address.

Petitioner consents to email service at [IPR50048-0059IP1@fr.com](mailto:IPR50048-0059IP1@fr.com).

Respectfully submitted,

Dated: April 1, 2024

/W. Karl Renner/

W. Karl Renner, Reg. No. 41,265  
Kim H. Leung, Reg. No. 64,399  
Baile Xie, Reg. No. 78,380  
Fish & Richardson P.C.  
60 South Sixth Street  
Minneapolis, MN 55402  
T: 202-783-5070  
F: 877-769-7945

(Control No. IPR2024-00760)

Attorneys for Petitioners

**CERTIFICATION UNDER 37 CFR § 42.24**

Under the provisions of 37 CFR § 42.24(d), the undersigned hereby certifies that the word count for the foregoing Petition for *Inter Partes* Review totals 13,939 words, which is less than the 14,000 allowed under 37 CFR § 42.24.

Dated: April 1, 2024

/W. Karl Renner/

W. Karl Renner, Reg. No. 41,265

Kim H. Leung, Reg. No. 64,399

Baile Xie, Reg. No. 78,380

Fish & Richardson P.C.

60 South Sixth Street

Minneapolis, MN 55402

T: 202-783-5070

F: 877-769-7945

Attorneys for Petitioner

**CERTIFICATE OF SERVICE**

Pursuant to 37 CFR §§ 42.6(e)(4)(i) *et seq.* and 42.105(b), the undersigned certifies that on April 1, 2024, a complete and entire copy of this Petition for *Inter Partes* Review and all supporting exhibits were provided via Federal Express, to the Patent Owner, by serving the correspondence address of record as follows:

ASHOK TANKHA  
36 Greenleigh drive  
Sewell, NJ 08080  
(856) 266-5145

/Crena Pacheco/

Crena Pacheco  
Fish & Richardson P.C.  
60 South Sixth Street, Suite 3200  
Minneapolis, MN 55402  
[pacheco@fr.com](mailto:pacheco@fr.com)