Filed: February 10, 2017

Filed on behalf of:
Patent Owner Voip-Pal.com Inc.
By: Kerry Taylor
John M. Carson
KNOBBE, MARTENS, OLSON & BEAR, LLP
2040 Main Street, 14th Floor
Irvine, CA 92614
Tel.: (858) 707-4000
Fax: (858) 707-4001
Email: BoxDigifonica@knobbe.com

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

APPLE INC.

Petitioner,

v.

VOIP-PAL.COM, INC.,

Patent Owner

Case No. IPR2016-01198
U.S. Patent 9,179,005

PATENT OWNER RESPONSE TO PETITION

TABLE OF CONTENTS

Page No.

TABLE OF CONTENTS
(*cont'd*)

Page No.

<u>TABLE OF AUTHORITIES</u>

Page No(s).

Eaton v. Evans,
204 F.3d 1094 (Fed. Cir. 2000) 4, 48

In re Gordon,
733 F.2d 900 (Fed. Cir. 1984) 59

KSR Int'l. Co. v. Teleflex, Inc.,
550 U.S. 398 (2007)..69

In re NuVasive, Inc.
842 F.3d 1376 70

OTHER AUTHORITIES

35 U.S.C. § 102..3, 4, 47, 48

35 U.S.C. § 316..3

Patent Owner Voip-Pal.com, Inc. ("Voip-Pal") respectfully submits this Response to the Petition for *Inter Partes* Review of U.S. 9,179,005 (the '005 Patent) (Paper 1) by Apple Inc. ("Apple").

I. <u>INTRODUCTION</u>

Digifonica, a real party-in-interest to this proceeding and wholly owned subsidiary of Patent Owner Voip-Pal, starting in 2004 employed top professionals including three Ph.D.'s with various engineering backgrounds, to develop innovative software solutions for communications, which, by the mid-2000s it had implemented in four nodes across three geographic regions. Digifonica's R&D efforts led to several patents, including the '005 Patent.

Prior to the '005 Patent, private branch exchange (PBX) systems typically enabled users to call destinations internal to the PBX by dialing an extension (i.e., "private number") and destinations external to the PBX on the public switched telephone network (PSTN) by dialing a "public number." Such PBX systems relied on a *user-specified classification* of the dialed number to interpret the number and route the call. For example, a user placing a call to the public network dialed a predefined prefix such as "9" to indicate that subsequent digits were to be interpreted as a public PSTN number. If no prefix was dialed, the dialed digits were to be interpreted as a private PBX extension. The number alone, as dialed, dictated how the call was routed. Thus, the user made an affirmative decision when placing a call as to whether the call would be public or private.

Digifonica's system employed an approach fundamentally different from traditional PBX's: it did not rely on a caller-specified classification (e.g., prefix digit) to distinguish private calls from PSTN calls. Digifonica's system provided flexible, user-specific dialing features and could decouple the type of number being called from the manner in which the call would be handled. For example, even if a public PSTN number was dialed, Digifonica's system could determine that the call should be routed to a private network, thus allowing the advantages of private network calling even when callers were unaware that the call recipient ("callee") was a Digifonica system subscriber.

Petitioner represents that the claims would have been obvious over a combination of Chu '684 with Chu '366 or Chen. The cited references describe traditional PBX systems and PSTN number reformatting. Petitioner's proffered obviousness construction is unsupported by the teachings of the references. A fair reading of the combination of Chu '684 with either Chu '366 or Chen would lead to routing calls to a public or private network based solely on the caller's dialed number, which is distinct from classifying calls in the manner recited in the claims. Petitioner's attempt to combine Chu '684 with Chu '366 or Chen distorts the operation of Chu '684's system. These distortions are evident when one looks at the shifting explanations of Petitioner's Declarant in describing the proposed combinations during deposition. These distortions also undermine the ability of the combined system to accurately function for its intended purpose, i.e., call routing.

Therefore, Petitioner's arguments fail to carry its burden of proving that Claims 1, 24-26, 49-50, 73-79, 83-84, 88-89, 92, 94-96, 98, and 99 of the '005 Patent would have been obvious. 35 U.S.C. § 316(e). Petitioner fails to meet its burden for at least the following reasons:

1. Patent Owner submits herewith detailed evidence that the invention claimed in the '005 Patent was actually reduced to practice at least as early as June 6, 2005, prior to the filing dates of both Chu '366 and Chen, thus neither reference constitutes prior art under pre-AIA 35 U.S.C. § 102(e).

2. The proposed combinations fail to provide all claim elements, and thus fail to establish a *prima facie* obviousness case.

3. The Petition is premised on a fundamental misunderstanding of Chu '684, which, once properly understood, undermines Petitioner's proposed combinations.

For any one of the foregoing deficiencies, the Board should confirm the non-obviousness of Claims 1, 24-26, 49-50, 73-79, 83-84, 88-89, 92, 94-96, 98, and 99 of the ’005 Patent.

II. ARGUMENT

A. CHU ‘366 IS NOT PRIOR ART UNDER PRE-AIA 35 U.S.C. 102(e)

In Ground 1, Petitioner asserts that “U.S. Patent No. 8,036,366 to Chu (“Chu ’366”) was filed on Aug. 4, 2006 and therefore qualifies as prior art with regard to the ’005 Patent under 35 U.S.C. §102(e).” Petition at 10. However, pre-AIA 35 U.S.C. §102(e) establishes that a reference is prior art if it is “filed in the United States before the invention by the applicant” and Chu ‘366 was not filed before the invention by the inventors of the ‘005 Patent.

Prior invention can be established by an actual reduction to practice before the priority date. *Eaton v. Evans*, 204 F.3d 1094, 1097 (Fed. Cir. 2000). The inventors of the ‘005 Patent reduced the claimed subject matter to practice well before Chu ‘366’s filing date of August 4, 2006 and in fact had a system in operation that practiced the claims of the ‘005 Patent by at least as early as June 6, 2005. Evidence submitted with this response and discussed below includes computer source code, an outside technical review, reports, design documents and emails, as well as expert, inventor and employee testimony. This evidence establishes that before the filing date of Chu ‘366 the inventors of the ‘005 had reduced to practice the inventions of the challenged claims.

1. Digifonica’s RBR Software

The ‘005 Patent inventors started the company Digifonica in 2004 and developed a system that allowed calls to be placed between two IP phones and between an IP phone and traditional phones on the public switched telephone network (PSTN). **Ex. 2018** at ¶3. **Ex. 2012** at ¶3. **Ex. 2013** at ¶2. The system developed by the inventors utilized multiple geographically distributed “supernodes” that would handle routing and billing functions for a set of IP phones. By June 2005 Digifonica had deployed two supernodes, one in London, UK and one in Vancouver, Canada. **Ex. 2018** at ¶3. **Ex. 2012** at ¶3. **Ex. 2013** at ¶2.

One of the key components of the Digifonica supernodes was referred to as “RBR”, which was a software and hardware platform that received information related to the initiation of a call and responded with call routing messages. **Ex. 2012** at ¶4. **Ex. 2013** at ¶9. **Ex. 2018** at ¶5.

The Digifonica source code, including the RBR source code, was maintained in a source code control system known as “Subversion”, which maintains the complete history of all changes to the RBR source code. **Ex. 2012** at ¶4. **Ex. 2010** at ¶10. Mr. Pentti Huttunen, a former employee of Digifonica retained a portable hard drive that contained the Digifonica Subversion repository. **Ex. 2010** at ¶¶7-10. The Subversion repository was archived by Mr. Huttunen and it has remained in his possession unmodified. *Id.* at ¶11. Mr. Huttunen's disk drive was delivered to Mr. Ryan Purita, who analyzed the "svn.tar" file and computed electronic signatures for that file. **Ex. 2011** at ¶4. Dr. Mangione-Smith was provided with the "svn.tar" file and has verified it conforms to the signatures provided by Mr. Purita, ensuring that it is identical to the file that was safeguarded by Mr. Huttunen and analyzed by Mr. Purita. **Ex. 2016** at ¶20.

The Subversion source code repository contains the history of the files that make up the RBR software development including all versions and the changes that were made with each version. *Id*. The Subversion logs identify a version of the RBR software, Version 361, that was last modified on June 6, 2005 at 09:22:59AM. **Ex. 2016** at ¶21. Dr. Mangione-Smith generated a log file for the RBR software that includes the history of versions for the period up through November 2006 (**Ex. 2015**). **Ex. 2016** at ¶23. Dr. Mangione-Smith has generated a printout of Version 361 of the RBR software (**Ex. 2014**), has analyzed Version 361 of the RBR software, and has compared it to the challenged claims of the '005. **Ex. 2016** at pages **12-57**.

The RBR software implemented a call routing controller, which corresponds to the Routing Controller 16 illustrated in Fig. 1 of the '005 Patent and recited in the challenged claims. **Ex. 2013** at ¶12. **Ex. 2012** at ¶16. **Ex. 2016** at ¶24. The RBR software was implemented using a set of scripts in the programming language PHP, and the functionality that implements the features recited in the claims of the ‘005 patent can be found in four PHP files that ran on the RBR server: invite.php, call_routes.class.php, call_ttl.class.php and call_e164.class.php. **Ex. 2016** at ¶22.

The Table below illustrates how the Digifonica system running Version 361 of the RBR source code practices the challenged claims of the ‘005 Patent. Note the Table below numbers claim elements differently than the Petition in order to more clearly demonstrate how Version 361 meets all claim elements.

Patent 9,179,005	**RBR Source Code Version 361**
1. [1p] A process for producing a routing message for routing communications between a caller and a callee in a communication system, the process comprising:	The RBR server implements a call routing controller. Callers and callees, which may be internet (IP) phones or endpoints on the PSTN, are associated with nodes. Within a Digifonica supernode, the RBR server facilitates communication by responding to requests from the B2BUA server and providing routing messages back to the B2BUA server. **Ex. 2016 at 12-13**.[1]
[1a] using a caller	In response to a SIP INVITE message from an IP phone,

[1] References to Ex. 2016 throughout the claim chart refer to the page number in Ex. 2016 and not the paragraph number.

identifier associated with the caller to locate a caller dialing profile comprising a plurality of calling attributes associated with the caller;	the B2BUA server sends a USER_INVITE message to the RBR server, which then loads a calling profile using the "caller" parameter. **Ex. 2016 at 13-17**.
	The inviteClient() function is the entry point for the RBR software that receives this message. The inviteClient function receives five parameters in an XML_RPC_Message (**Ex. 2014 at 42**, "invite.php" line 27):
	`* This function is called by the XML-RPC Server expecting type (integer`
	` * like 4 is THIRD_PARTY_INVITE) ,username,callee,caller,callid (In that`
	` * order in the XML_RPC_Message)`
	A type value of 2 represents a "USER_INVITE" message (**Ex. 2014 at 42**, "invite.php" line 35):
	`$type_arr[2] = 'USER_INVITE';`
	Besides the type parameters, the USER_INVITE message contains four other parameters identified as "username", "callee", "caller" and "called" (**Ex. 2014 at 42**, "invite.php" lines 44-47):
	`$username = $params->getParam(1)->scalarval();`
	`$callee    = $params->getParam(2)->scalarval();`
	`$caller   = $params->getParam(3)->scalarval();`
	`$callid   = $params->getParam(4)->scalarval();`
	The "caller" variable corresponds to the caller identifier and the "callee" parameter corresponds to the callee identifier.
	The inviteClient() function first constructs a new call_routes() object (**Ex. 2014 at 43**, "invite.php" line 60):

```
$call = new
call_routes($caller,$caller_domain,$callee,$c
allee_domain,$type,$forced_cid);
```

The constructor for call_routes() is defined at lines 59-74 of call_routes.class.php (**Ex. 2014 at 19**):

```
/**
 * Constructor... the class must be called
with the defined parameters. Callee domain is
optional and the class will try to guess it.
 * Special is a flag (THIRD_PARTY_INVITE)
when set, the caller is not set and
concidered a incoming PSTN call.
 *
 * @param String $caller
 * @param String $domain
 * @param String $callee
 * @param String $callee_domain
 * @param String $special
 * @return call_routes
 */
function
call_routes($caller,$domain,$callee,$callee_d
omain='',$special='') {
    $this->special_request = $special;
    $this-
>call_ttl($caller,$domain,$callee,$callee_dom
ain);
    $this->genereate_routes();
}
```

The call_routes() constructor calls the call_ttl() function (**Ex. 2014 at 19**, “call_routes.class.php” line 72):

```
$this-
>call_ttl($caller,$domain,$callee,$callee_dom
ain);
```

The call_ttl() constructor is defined at lines 131-153 of call_ttl.class.php (**Ex. 2014 at 27**):

```
/**
```

```
 * Class constructor must be initiated with
at least caller, domain (callee_domain and
special flag are optional.)<br>
 * Once it is initiated, it will
automatically execute proper functions and do
its due diligence, to create TTL.
 * $this->ttl is where you can get the Total
time to live for this call. can be accessed
byt extending classes and <br>
 * through the function get_ttl().
 *
 * @param string $caller
 * @param string $domain
 * @param string $callee
 * @param string $callee_domain
 * @param string $special
 * @return call_ttl
 */
function
call_ttl($caller,$domain,$callee,$callee_doma
in='',$special='') {
    if(!$callee_domain){
        //try to guess
        if($temp_domain = $this-
>guess_user_domain($callee)) $this-
>callee_domain=$temp_domain;
    }
    $this-
>set_caller_data($caller,$domain,$special);

    $this->set_callee($callee,$this-
>callee_domain);
    $this->generate_ttl();
}
```

The call_ttl() function calls the set_caller_data() function (**Ex. 2014 at 27**, “call_ttl.class.php” line 149):

```
$this-
>set_caller_data($caller,$domain,$special);
```

The set_caller_data() function is defined at lines 187-189 of call_ttl.class.php and calls the set_caller() function (**Ex. 2014 at 28**, “call_ttl.class.php” line 188):

```
$this->set_caller($caller,$domain);
```

The set_caller() function is defined at lines 325-340 of call_e164.class.php (**Ex. 2014 at 7**):

```
/**
 * Takes care of sequence of execution to
create a caller profile
 *
 * @param String $caller
 * @param String $domain
 */
public function set_caller($caller,$domain){

    $this->caller = $caller;

    if(! ($this->special_request ==
'THIRD_PARTY_INVITE')){
        $this->caller_domain = $domain;
        $this->create_caller_profile();
    }
    $this->caller_is_set = true;
}
```

The set_caller() function calls the create_caller_profile() function (**Ex. 2014 at 7**, “call_e164.class.php” line 337):

```
$this->create_caller_profile();
```

The create_caller_profile() function is defined at lines 342-391 of call_e164.class.php (**Ex. 2014 at 7-8**) (lines 342-346 shown):

```
/**
 * All aspects of the caller Flags and data
needed to identify a caller is set here.
including overlapping numbers.
 *
 */
private function create_caller_profile(){
```

The create_caller_profile() function queries an SQL database to load a caller profile (**Ex. 2014 at 7**,

	“call_e164.class.php” line 348):
	`$user_data = mysql_fetch_array($this->sql_query("SELECT * FROM subscriber_dialing_profile WHERE username='$this->caller' and domain= '$this->caller_domain'"));`
	The data returned from the SQL query include the caller’s idd (international direct dial digits), ndd (national direct dial digits) and area code (**Ex. 2014 at 7**, “call_e164.class.php” lines 350-356):
	`$this->caller_idd = $user_data['idd'];`
	`$this->caller_ndd = $user_data['ndd'];`
	`$this->caller_areacode = $user_data['area_code'];`
	`$this->caller_countrycode = $user_data['country_code'];`
	`$temp = explode(",",$user_data['local_length']);`
	`$this->caller_min_local_length = $temp[0];`
	`$this->caller_max_local_length = $temp[1];`

[1b] when at least one of said calling attributes and at least a portion of a callee identifier associated with the callee meet private network classification criteria,	The RBR server matches attributes in the caller profile values against the callee identifier. The RBR server classifies the call as a public network call or a private network call. **Ex. 2016 at 17-20**.
	The call_ttl() constructor, after calling the set_caller_data() function calls the set_callee() function (**Ex. 2014 at 27**, “call_ttl.class.php” line 151):
	`$this->set_callee($callee,$this->callee_domain);`
	The set_callee() function is defined at lines 393-418 of call_e164.class.php (**Ex. 2014 at 8-9**) (lines 393-399 shown):
	`/**`
	` * Takes care of the sequence and logic`

```
needed to create a callee
 *
 * @param String $callee
 * @param String $callee_domain
 */
public function
set_callee($callee,$callee_domain=''){
```

The set_callee() function calls the create_callee_profile() function (**Ex. 2014 at 8**, "call_e164.class.php" line 407):

```
$this->create_callee_profile();
```

The create_callee_profile() function is defined at lines 450-539 of call_e164.class.php (**Ex. 2014 at 9-11**) (lines 450-454 shown):

```
/**
 * Creates a callee profile, all aspects of
the callee is set here including the e164
number is also set here. NOTE: the e164
number can change when call forwarding
occures.
 *
 */
private function create_callee_profile(){
```

The create_callee_profile() function matches values that were previously retrieved as part of the caller profile with the callee identifier. For example, a check is made to see if the callee identifier begins with the international dialing digits (idd) (**Ex. 2014 at 9-10**, "call_e164.class.php" lines 467-476):

```
case (isset($this->caller_idd) &&
preg_match("/^".$this->caller_idd."/",$this-
>callee)):
$idd_length = strlen($this->caller_idd);
$this->set_ml_id($idd_length);
if($this->callee_data['ml_id']){ //callee was
found and set
$this->callee_type = 1;
$this->check_callee_length($idd_length);
$this->set_e164_formed_number(substr($this-
>callee,$idd_length));
}else{
    throw new Exception(4);
}
```

Further, create_callee_profile() checks for matching national dialing digits (**Ex. 2014 at 10**, “call_e164.class.php” lines 479-494), checks for matching area codes (**Ex. 2014 at 10**, “call_e164.class.php” lines 495-501), checks for a matching local number (**Ex. 2014 at 10-11**, “call_e164.class.php” lines 506-513), and checks for a network number (**Ex. 2014 at 11**, “call_e164.class.php” lines 520-527).

The create_callee_profile() function classifies the call by setting the callee_type variable to a value from 1 to 7. In the case of a private classification, the callee_type variable is set to one of the values 4 or 7 (**Ex. 2014 at 11**, “call_e164.class.php” lines 523 and 525 respectively):

```
$this->callee_type = 4;
$this->callee_type = 7;
```

The set_callee() function calls the check_for_did() function. The check_for_did() function is defined at lines 420-441 of call_e164.class.php (**Ex. 2014 at 9**):

```
/**
 * Checks if the number being called is a
mapped DID in our system, if it is we rout it
internally
 *
 * @param String $user
 * @return Boolean or
DidMap(DigifonicaNumber#Domain)
 */
protected function check_for_did($user=''){
     $num =  $user?$user:$this->callee;
     $did_check = @mysql_fetch_array($this-
>sql_query("SELECT * FROM did_bank WHERE
did='$num'"));
```

<table>
<tr>
<td></td>
<td>

```
     if($did_check[0]){
          try{
               if($did_check['mapped_to'] and
$did_check['domain']){
                    return
$did_check['mapped_to']."#".$did_check['domai
n'];
               }else{
                    throw new Exception(8);
               }
          }catch (Exception $e){
               $this-
>e164_exception_handler($e->getMessage());
          }
     }
     return false;
}
```

The check_for_did() function queries an SQL database to determine if the reformatted callee number is contained in the "did_bank" table. This check determines if the callee is mapped to a Digifonica subscriber account. In the case that the check_for_did() function returns a value, the set_callee() function will make a call to itself at line 415 of call_e164.class.php (**Ex. 2014 at 8**). This will cause a second call to the create_callee_profile() function, which will cause the callee_type to be changed to 4 or 7 depending on whether the destination is associated with the same supernode or a different supernode as the caller.

</td>
</tr>
<tr>
<td>[1c] producing a private network routing message for receipt by a call controller, said private network routing message identifying an address, on the private network, associated with the callee; and</td>
<td>

The RBR server produces a private network routing message in the case of a private network call classification. The routing message identifies an address associated with the destination Digifonica IP phone. **Ex. 2016 at 20-21**.

The call_routes() constructor, after calling the call_ttl() function calls the genereate_routes() function (**Ex. 2014 at 19**, "call_routes.class.php" line 73):

```
$this->genereate_routes();
```

</td>
</tr>
</table>

	The genereate_routes() function is defined at lines 91-148 of call_routes.class.php (**Ex. 2014 at 19-21**) (lines 91-95 shown):
	`/**`
	` * Generates the routes and the SIP messages including VM data.`
	` *`
	` */`
	`protected function genereate_routes(){`
	The genereate_routes() function tests the value of callee_type and in the case of 4 and 7 sets the routes[] array (**Ex. 2014 at 20**, “call_routes.class.php” line 146):
	`/**`
	` * Generates the routes and the SIP messages including VM data.`
	` *`
	` */`
	`protected function genereate_routes(){`
	The routes[] array is returned to the inviteClient() function and stored into a “$response” variable (**Ex. 2014 at 43**, “invite.php” line 62):
	`$response = $call->get_routes();`
	The inviteClient() function returns the $response variable to the B2BUA server over the XML_RPC connection (**Ex. 2014 at 43**, “invite.php, line 70):
	`return new XML_RPC_Response(new XML_RPC_Value($response, "string"));`

[1d] when at least one of said calling attributes and at least a portion of said callee identifier meet a public network	The RBR server matches attributes in the caller profile values against the callee identifier. The RBR server classifies the call as a public network call or a private network call. **Ex. 2016 at 21-24.** The call_ttl() constructor, after calling the

classification criterion,

set_caller_data() function calls the set_callee() function (**Ex. 2014 at 27**, “call_ttl.class.php” line 151):

```
$this->set_callee($callee,$this-
>callee_domain);
```

The set_callee() function is defined at lines 393-418 of call_e164.class.php (**Ex. 2014 at 8-9**) (lines 393-399 shown):

```
/**
 * Takes care of the sequence and logic
needed to create a callee
 *
 * @param String $callee
 * @param String $callee_domain
 */
public function
set_callee($callee,$callee_domain=''){
```

The set_callee() function calls the create_callee_profile() function (**Ex. 2014 at 8**, “call_e164.class.php” line 407):

```
$this->create_callee_profile();
```

The create_callee_profile() function is defined at lines 450-539 of call_e164.class.php (**Ex. 2014 at 9-11**) (lines 450-454 shown):

```
/**
 * Creates a callee profile, all aspects of
the callee is set here including the e164
number is also set here. NOTE: the e164
number can change when call forwarding
occures.
 *
 */
private function create_callee_profile(){
```

The create_callee_profile() function matches values that were previously retrieved as part of the caller profile with the callee identifier. For example, a check is made to see if the callee identifier begins with the international dialing digits (idd) (**Ex. 2014 at 9-10**, “call_e164.class.php” lines 467-476):

```
case (isset($this->caller_idd) &&
preg_match("/^".$this->caller_idd."/",$this-
>callee)):
$idd_length = strlen($this->caller_idd);
$this->set_ml_id($idd_length);
if($this->callee_data['ml_id']){ //callee was
found and set
$this->callee_type = 1;
$this->check_callee_length($idd_length);
$this->set_e164_formed_number(substr($this-
>callee,$idd_length));
}else{
    throw new Exception(4);
}
```

Further, create_callee_profile() checks for matching national dialing digits (**Ex. 2014 at 10**, “call_e164.class.php” lines 479-494), checks for matching area codes (**Ex. 2014 at 10**, “call_e164.class.php” lines 495-501), checks for a matching local number (**Ex. 2014 at 10-11**, “call_e164.class.php” lines 506-513), and checks for a network number (**Ex. 2014 at 11**, “call_e164.class.php” lines 520-527).

The create_callee_profile() function classifies the call by setting the callee_type variable to a value from 1 to 7. In the case of a public classification, the callee_type variable is set to one of the values 1, 2, 3 or 6 (**Ex. 2014 at 10**, “call_e164.class.php” lines 471, 483, 485, 497 and 510):

```
$this->callee_type = 1;
$this->callee_type = 6;
$this->callee_type = 2;
$this->callee_type = 3;
$this->callee_type = 3;
```

The set_callee() function calls the check_for_did() function. The check_for_did() function is defined at lines

<table>
<tr>
<td></td>
<td>

420-441 of call_e164.class.php (**Ex. 2014 at 9**):

```
/**
 * Checks if the number being called is a mapped DID in our system, if it is we rout it internally
 *
 * @param String $user
 * @return Boolean or DidMap(DigifonicaNumber#Domain)
 */
protected function check_for_did($user=''){
     $num =  $user?$user:$this->callee;
     $did_check = @mysql_fetch_array($this->sql_query("SELECT * FROM did_bank WHERE did='$num'"));
     if($did_check[0]){
          try{
               if($did_check['mapped_to'] and $did_check['domain']){
                    return $did_check['mapped_to']."#".$did_check['domain'];
               }else{
                    throw new Exception(8);
               }
          }catch (Exception $e){
               $this->e164_exception_handler($e->getMessage());
          }
     }
     return false;
}
```

The check_for_did() function queries an SQL database to determine if the reformatted callee number is contained in the “did_bank” table. This check determines if the callee is mapped to a Digifonica subscriber account and will return false in the case that there is no value.

</td>
</tr>
<tr>
<td>[1e] producing a public network routing message for receipt by the call</td>
<td>The RBR server produces a public network routing message in the case of a public network call classification. The routing message identifies one or more addresses for PSTN gateways. <b>Ex. 2016 at 24-25</b>.</td>
</tr>
</table>

controller, said public network routing message identifying a gateway to the public network.	The genereate_routes() function tests the value of callee_type and in the case of 1, 2, 3 or 6 queries an SQL database to get 'supplier' addresses (**Ex. 2014 at 19**, "call_routes.class.php" line 99):
	`$routes_q = $this->sql_query("SELECT * FROM suppliers WHERE ml_id=".$this->callee_data['ml_id']." ORDER BY sup_price ASC");`
	The genereate_routes() function then processes the results from the SQL query and builds a routes[] array (**Ex. 2014 at 20**, "call_routes.class.php" line 133):
	`$this->routes[] = $c."h323-ivr-in = 'Routing:".$my_routes_info[$curr_route]['prepend'].$my_routes_info[$curr_route]['ndd_idd_replacement'].$this->e164_formed_nubmer."@".$curr_route.$my_routes_info[$curr_route]['port'].";credit-time=$this->ttl".$my_routes_info[$curr_route]['auth'].$my_routes_info[$curr_route]['expires'].";cli=".$caller_id."'";`
	The routes[] array is returned to the inviteClient() function and stored into a "$response" variable (**Ex. 2014 at 43**, "invite.php" line 62):
	`$response = $call->get_routes();`
	The inviteClient() function returns the $response variable to the B2BUA server over the XML_RPC connection (**Ex. 2014 at 43**, "invite.php" line 70):
	`return new XML_RPC_Response(new XML_RPC_Value($response, "string"));`

24. The process of claim 1, further	The RBR server returns the routing message to the B2BUA server, which then effects the routing of the call.

comprising causing the private network routing message or the public network routing message to be communicated to a call controller to effect routing of the call.	**Ex. 2016 at 25-26**. The inviteClient() function returns the $response variable to the B2BUA server over the XML_RPC connection (**Ex. 2014 at 43**, “invite.php” line 70): `return new XML_RPC_Response(new XML_RPC_Value($response, "string"));`
25. A non-transitory computer readable medium encoded with codes for directing a processor to execute the method of claim 1.	The RBR server is implemented on a computer with storage for the computer software executed by a processor. **Ex. 2016 at 26**.
26. [26p] A call routing controller apparatus for producing a routing message for routing communications between a caller and a callee in a communication system, the apparatus comprising:	See claim element [1p].
[26a] at least one processor operably configured to: use a caller identifier associated with the caller to locate a caller dialing profile comprising a plurality of calling attributes	See claim element [1a].

associated with the caller;	
[26b] when at least one of said calling attributes and at least a portion of a callee identifier associated with the callee meet private network classification criteria, produce a private network routing message for receipt by a call controller, said private network routing message identifying an address, on the private network, associated with the callee; and	See claim elements [1b] and [1c].
[26c] when at least one of said calling attributes and at least a portion of said callee identifier meet a public network classification criterion, produce a public network routing message for receipt by the call controller, said public network routing message identifying a gateway to the public network.	See claim elements [1d] and [1e].

49. The apparatus of claim 26, wherein said at least one processor is further operably configured to cause the private network routing message or the public network routing message to be communicated to a call controller to effect routing of the call.	See claim 24.
50. [50p] A call routing controller apparatus for producing a routing message for routing communications between a caller and a callee in a communication system, the apparatus comprising:	See claim element [1p].
[50a] means for using a caller identifier associated with the caller to locate a caller dialing profile comprising a plurality of calling attributes associated with the caller; and	In response to a SIP INVITE message from an IP phone, the B2BUA server sends a USER_INVITE message to the RBR server, which then loads a calling profile using the "caller" parameter. **Ex. 2016 at 28-32**. The inviteClient() function is the entry point for the RBR software that receives this message. The inviteClient function receives five parameters in an XML_RPC_Message (**Ex. 2014 at 42**, "invite.php" line 27). A type value of 2 represents a "USER_INVITE" message (**Ex. 2014 at 42**, "invite.php" line 35).

	Besides the type parameters, the USER_INVITE message contains four other parameters identified as “username”, “callee”, “caller” and “called” (**Ex. 2014 at 42**, “invite.php” lines 44-47). The “caller” variable corresponds to the caller identifier and the “callee” parameter corresponds to the callee identifier. The inviteClient() function first constructs a new call_routes() object (**Ex. 2014 at 43**, “invite.php” line 60). The constructor for call_routes() is defined at lines 59-74 of call_routes.class.php (**Ex. 2014 at 19)**. The call_routes() constructor calls the call_ttl() function (**Ex. 2014 at 19**, “call_routes.class.php” line 72). The call_ttl() constructor is defined at lines 131-153 of call_ttl.class.php (**Ex. 2014 at 27**). The call_ttl() function calls the set_caller_data() function (**Ex. 2014 at 27**, “call_ttl.class.php” line 149). The set_caller_data() function is defined at lines 187-189 of call_ttl.class.php and calls the set_caller() function (**Ex. 2014 at 28**, “call_ttl.class.php” line 188). The set_caller() function is defined at lines 325-340 of call_e164.class.php (**Ex. 2014 at 7**). The set_caller() function calls the create_caller_profile() function (**Ex. 2014 at 7**, “call_e164.class.php” line 337). The create_caller_profile() function is defined at lines 342-391 of call_e164.class.php (**Ex. 2014 at 7-8**). The create_caller_profile() function queries an SQL

	database to load a caller profile (**Ex. 2014 at 7**, "call_e164.class.php" line 348). The data returned from the SQL query include the caller's idd (international direct dial digits), ndd (national direct dial digits) and area code (**Ex. 2014 at 7**, "call_e164.class.php" lines 350-356).
[50b] means for, when at least one of said calling attributes and at least a portion of a callee identifier associated with the callee meet private network classification criteria, producing a private network routing message for receipt by a call controller, said private network routing message identifying an address, on the private network, associated with the callee; and	The RBR server matches attributes in the caller profile values against the callee identifier. The RBR server classifies the call as a public network call or a private network call. The RBR server produces a private network routing message in the case of a private network call classification. The routing message identifies an address associated with the destination Digifonica IP phone. **Ex. 2016 at 32-37**. The call_ttl() constructor, after calling the set_caller_data() function calls the set_callee() function (**Ex. 2014 at 27**, "call_ttl.class.php" line 151). The set_callee() function is defined at lines 393-418 of call_e164.class.php (**Ex. 2014 at 8-9**) . The set_callee() function calls the create_callee_profile() function (**Ex. 2014 at 8**, "call_e164.class.php" line 407). The create_callee_profile() function is defined at lines 450-539 of call_e164.class.php (**Ex. 2014 at 9-11**). The create_callee_profile() function matches values that were previously retrieved as part of the caller profile with the callee identifier. For example, a check is made to see if the callee identifier begins with the international dialing digits (idd) (**Ex. 2014 at 9-10**, "call_e164.class.php" lines 467-476). Further, create_callee_profile() checks for matching national dialing digits (**Ex. 2014 at 10**,

	"call_e164.class.php" lines 479-494), checks for matching area codes (**Ex. 2014 at 10**, "call_e164.class.php" lines 495-501), checks for a matching local number (**Ex. 2014 at 10-11**, "call_e164.class.php" lines 506-513), and checks for a network number (**Ex. 2014 at 11**, "call_e164.class.php" lines 520-527). The create_callee_profile() function classifies the call by setting the callee_type variable to a value from 1 to 7. In the case of a private classification, the callee_type variable is set to one of the values 4 or 7 (**Ex. 2014 at 11**, "call_e164.class.php" lines 523 and 525 respectively). The set_callee() function calls the check_for_did() function. The check_for_did() function is defined at lines 420-441 of call_e164.class.php (**Ex. 2014 at 9**). The check_for_did() function queries an SQL database to determine if the reformatted callee number is contained in the "did_bank" table. This check determines if the callee is mapped to a Digifonica subscriber account. In the case that the check_for_did() function returns a value, the set_callee() function will make a call to itself at line 415 of call_e164.class.php (**Ex. 2014 at 8**). This will cause a second call to the create_callee_profile() function, which will cause the callee_type to be changed to 4 or 7 depending on whether the destination is associated with the same supernode or a different supernode as the caller. The call_routes() constructor, after calling the call_ttl() function calls the genereate_routes() function (**Ex. 2014 at 19**, "call_routes.class.php" line 73). The genereate_routes() function is defined at lines 91-148 of call_routes.class.php (**Ex. 2014 at 19-21**). The genereate_routes() function tests the value of callee_type and in the case of 4 and 7 sets the routes[] array (**Ex. 2014 at 20**, "call_routes.class.php" line 146).

	The routes[] array is returned to the inviteClient() function and stored into a “$response” variable (**Ex. 2014 at 43**, “invite.php” line 62). The inviteClient() function returns the $response variable to the B2BUA server over the XML_RPC connection (**Ex. 2014 at 43**, “invite.php, line 70).
[50c] means for, when at least one of said calling attributes and at least a portion of said callee identifier meet a public network classification criterion, producing a public network routing message for receipt by the call controller, said public network routing message identifying a gateway to the public network.	The RBR server matches attributes in the caller profile values against the callee identifier. The RBR server classifies the call as a public network call or a private network call. The RBR server produces a public network routing message in the case of a public network call classification. The routing message identifies one or more addresses for PSTN gateways. **Ex. 2016 at 37-40**. The call_ttl() constructor, after calling the set_caller_data() function calls the set_callee() function (**Ex. 2014 at 27**, “call_ttl.class.php” line 151). The set_callee() function is defined at lines 393-418 of call_e164.class.php (**Ex. 2014 at 8-9**). The set_callee() function calls the create_callee_profile() function (**Ex. 2014 at 8**, “call_e164.class.php” line 407). The create_callee_profile() function is defined at lines 450-539 of call_e164.class.php (**Ex. 2014 at 9-11**). The create_callee_profile() function matches values that were previously retrieved as part of the caller profile with the callee identifier. For example, a check is made to see if the callee identifier begins with the international dialing digits (idd) (**Ex. 2014 at 9-10**, “call_e164.class.php” lines 467-476). Further, create_callee_profile() checks for matching national dialing digits (**Ex. 2014 at 10**,

	"call_e164.class.php" lines 479-494), checks for matching area codes (**Ex. 2014 at 10**, "call_e164.class.php" lines 495-501), checks for a matching local number (**Ex. 2014 at 10-11**, "call_e164.class.php" lines 506-513), and checks for a network number (**Ex. 2014 at 11**, "call_e164.class.php" lines 520-527). The create_callee_profile() function classifies the call by setting the callee_type variable to a value from 1 to 7. In the case of a public classification, the callee_type variable is set to one of the values 1, 2, 3 or 6 (**Ex. 2014 at 10**, "call_e164.class.php" lines 471, 483, 485, 497 and 510). The set_callee() function calls the check_for_did() function. The check_for_did() function is defined at lines 420-441 of call_e164.class.php (**Ex. 2014 at 9**). The check_for_did() function queries an SQL database to determine if the reformatted callee number is contained in the "did_bank" table. This check determines if the callee is mapped to a Digifonica subscriber account and will return false in the case that there is no value. The genereate_routes() function tests the value of callee_type and in the case of 1, 2, 3 or 6 queries an SQL database to get 'supplier' addresses (**Ex. 2014 at 19**, "call_routes.class.php" line 99). The genereate_routes() function then processes the results from the SQL query and builds a routes[] array (**Ex. 2014 at 20**, "call_routes.class.php" line 133). The routes[] array is returned to the inviteClient() function and stored into a "$response" variable (**Ex. 2014 at 43**, "invite.php" line 62). The inviteClient() function returns the $response variable to the B2BUA server over the XML_RPC connection (**Ex. 2014 at 43**, "invite.php" line 70).

73. The apparatus of claim 50, further comprising means for causing the private network routing message or the public network routing message to be communicated to a call controller to effect routing of the call.	The RBR server returns the routing message to the B2BUA server, which then effects the routing of the call. **Ex. 2016 at 41**. The inviteClient() function returns the $response variable to the B2BUA server over the XML_RPC connection (**Ex. 2014 at 43**, "invite.php" line 70).
74. [74p] A method of routing communications in a packet switched network in which a first participant identifier is associated with a first participant and a second participant identifier is associated with a second participant in a communication, the method comprising:	See claim element [1p].
[74a] after the first participant has accessed the packet switched network to initiate the communication, using the first participant identifier to locate a first participant profile comprising a plurality	See claim element [1a].

of attributes associated with the first participant;	
[74b] when at least one of the first participant attributes and at least a portion of the second participant identifier meet a first network classification criterion,	The RBR server matches attributes in the caller profile values against the callee identifier. The RBR server classifies the call as a public network call or a private network call. **Ex. 2016 at 42-45**. The call_ttl() constructor, after calling the set_caller_data() function calls the set_callee() function (**Ex. 2014 at 27**, “call_ttl.class.php” line 151). The set_callee() function is defined at lines 393-418 of call_e164.class.php (**Ex. 2014 at 8-9**). The set_callee() function calls the create_callee_profile() function (**Ex. 2014 at 8**, “call_e164.class.php” line 407). The create_callee_profile() function is defined at lines 450-539 of call_e164.class.php (**Ex. 2014 at 9-11**). The create_callee_profile() function matches values that were previously retrieved as part of the caller profile with the callee identifier. For example, a check is made to see if the callee identifier begins with the international dialing digits (idd) (**Ex. 2014 at 9-10**, “call_e164.class.php” lines 467-476). Further, create_callee_profile() checks for matching national dialing digits (**Ex. 2014 at 10**, “call_e164.class.php” lines 479-494), checks for matching area codes (**Ex. 2014 at 10**, “call_e164.class.php” lines 495-501), checks for a matching local number (**Ex. 2014 at 10-11**, “call_e164.class.php” lines 506-513), and checks for a network number (**Ex. 2014 at 11**, “call_e164.class.php” lines 520-527). The create_callee_profile() function classifies the call by

	setting the callee_type variable to a value from 1 to 7. In the case of a private classification, the callee_type variable is set to one of the values 4 or 7 (**Ex. 2014 at 11**, "call_e164.class.php" lines 523 and 525 respectively). The set_callee() function calls the check_for_did() function. The check_for_did() function is defined at lines 420-441 of call_e164.class.php (**Ex. 2014 at 9**). The check_for_did() function queries an SQL database to determine if the reformatted callee number is contained in the "did_bank" table. This check determines if the callee is mapped to a Digifonica subscriber account. In the case that the check_for_did() function returns a value, the set_callee() function will make a call to itself at line 415 of call_e164.class.php (**Ex. 2014 at 8**). This will cause a second call to the create_callee_profile() function, which will cause the callee_type to be changed to 4 or 7 depending on whether the destination is associated with the same supernode or a different supernode as the caller.
[74c] producing a first network routing message for receipt by a controller, the first network routing message identifying an address in a first portion of the packet switched network, the address being associated with the second participant, the first portion being controlled by an entity; and	The RBR server produces a private network routing message in the case of a private network call classification. The routing message identifies an address associated with the destination Digifonica IP phone. **Ex. 2016 at 45-46**. The call_routes() constructor, after calling the call_ttl() function calls the genereate_routes() function (**Ex. 2014 at 19**, "call_routes.class.php" line 73). The genereate_routes() function is defined at lines 91-148 of call_routes.class.php (**Ex. 2014 at 19-21**). The genereate_routes() function tests the value of callee_type and in the case of 4 and 7 sets the routes[] array (**Ex. 2014 at 20**, "call_routes.class.php" line 146). The routes[] array is returned to the inviteClient() function

	and stored into a "$response" variable (**Ex. 2014 at 43**, "invite.php" line 62). The inviteClient() function returns the $response variable to the B2BUA server over the XML_RPC connection (**Ex. 2014 at 43**, "invite.php, line 70).
[74d] when at least one of the first participant attributes and at least a portion of the second participant identifier meet a second network classification criterion,	The RBR server matches attributes in the caller profile values against the callee identifier. The RBR server classifies the call as a public network call or a private network call. **Ex. 2016 at 46-49**. The call_ttl() constructor, after calling the set_caller_data() function calls the set_callee() function (**Ex. 2014 at 27**, "call_ttl.class.php" line 151). The set_callee() function is defined at lines 393-418 of call_e164.class.php (**Ex. 2014 at 8-9**). The set_callee() function calls the create_callee_profile() function (**Ex. 2014 at 8**, "call_e164.class.php" line 407). The create_callee_profile() function is defined at lines 450-539 of call_e164.class.php (**Ex. 2014 at 9-11**). The create_callee_profile() function matches values that were previously retrieved as part of the caller profile with the callee identifier. For example, a check is made to see if the callee identifier begins with the international dialing digits (idd) (**Ex. 2014 at 9-10**, "call_e164.class.php" lines 467-476). Further, create_callee_profile() checks for matching national dialing digits (**Ex. 2014 at 10**, "call_e164.class.php" lines 479-494), checks for matching area codes (**Ex. 2014 at 10**, "call_e164.class.php" lines 495-501), checks for a matching local number (**Ex. 2014 at 10-11**, "call_e164.class.php" lines 506-513), and checks for a network number (**Ex. 2014 at 11**,

	"call_e164.class.php" lines 520-527). The create_callee_profile() function classifies the call by setting the callee_type variable to a value from 1 to 7. In the case of a public classification, the callee_type variable is set to one of the values 1, 2, 3 or 6 (**Ex. 2014 at 10**, "call_e164.class.php" lines 471, 483, 485, 497 and 510). The set_callee() function calls the check_for_did() function. The check_for_did() function is defined at lines 420-441 of call_e164.class.php (**Ex. 2014 at 9**). The check_for_did() function queries an SQL database to determine if the reformatted callee number is contained in the "did_bank" table. This check determines if the callee is mapped to a Digifonica subscriber account and will return false in the case that there is no value.
[74e] producing a second network routing message for receipt by the controller, the second network routing message identifying an address in a second portion of the packet switched network, the second portion not controlled by the entity.	The RBR server produces a public network routing message in the case of a public network call classification. The routing message identifies one or more addresses for PSTN gateways. **Ex. 2016 at 49-50**. The genereate_routes() function tests the value of callee_type and in the case of 1, 2, 3 or 6 queries an SQL database to get 'supplier' addresses (**Ex. 2014 at 19**, "call_routes.class.php" line 99). The genereate_routes() function then processes the results from the SQL query and builds a routes[] array (**Ex. 2014 at 20**, "call_routes.class.php" line 133). The routes[] array is returned to the inviteClient() function and stored into a "$response" variable (**Ex. 2014 at 43**, "invite.php" line 62). The inviteClient() function returns the $response variable to the B2BUA server over the XML_RPC connection (**Ex. 2014 at 43**, "invite.php" line 70).

75. The method of claim 74, wherein the packet switched network comprises the Internet.	The Digifonica supernodes were coupled to the Internet and the IP phone communicated with the supernodes over the Internet. **Ex. 2016 at 50**.
76. The method of claim 74, wherein the first participant identifier comprises a first participant telephone number or username.	The SIP_INVITE message contains the phone number of the caller, and this information is passed to the RBR server as the “caller” parameter. **Ex. 2016 at 50**.
77. The method of claim 74, wherein the second participant identifier comprises a second participant telephone number or username.	The SIP_INVITE message contains the dialed digits of the user, and this is passed to the RBR server as the “callee” parameter, which may be a telephone number or a Digifonica username. **Ex. 2016 at 50**.
78. The method of claim 74, wherein the communication comprises a voice-over-IP communication.	The Digifonica system was a voice-over-IP communication system. **Ex. 2016 at 50**.
79. The method of claim 74, wherein the packet switched network is accessed via an Internet service provider.	The Digifonica supernodes were coupled to the Internet through an Internet service provider. **Ex. 2016 at 51**.
83. The method of claim 74, wherein the	The RBR server classifies a call as a private call when the callee is determined to be a Digifonica subscriber. **Ex.**

first network classification criterion is satisfied when an address associated with the first participant and the address associated with the second participant are both in the first portion of the packet switched network.	**2016 at 51**.
84. The method of claim 74, wherein the address in the first portion is accessible through the first participant's Internet service provider.	The address associated with the destination IP phone is accessible over the Internet through an Internet service provider. **Ex. 2016 at 51**.
88. The method of claim 74, wherein the entity is an entity supplying communication services for the first portion.	Digifonica supplied communication services to its subscribers. **Ex. 2016 at 51**.
89. The method of claim 74, wherein the second network classification criterion is satisfied when access to the second participant requires routing through a portion of the packet switched network	In the case that a call is classified as a public call, then access to the callee requires routing through a PSTN gateway coupled to the Internet operated by a PSTN suppler. **Ex. 2016 at 51-52**.

operated by a communication service supplier.	
92. The method of claim 74, wherein the address in the second portion of the packet switched network comprises an address accessed by a communication service supplier.	The one or more PSTN gateways identified in a public network routing message are accessed by a PSTN supplier. **Ex. 2016 at 52**.
94. [94p] A system for routing communications in a packet switched network in which a first participant in a communication has an associated first participant identifier and a second participant in the communication has an associated second participant identifier, the system comprising:	See claim element [1p].
[94a] a controller comprising: a processor operably configured to access a memory, wherein the processor is configured to: after the first participant	See claim element [1a].

has accessed the packet switched network to initiate the communication, locate a first participant profile in the memory using the first participant identifier, the first participant profile comprising a plurality of attributes associated with the first participant	
[94b] produce a first network routing message when at least one of the first participant attributes and at least a portion of the second participant identifier meet a first network classification criterion, the first network routing message identifying an address in a first portion of the packet switched network, the address being associated with the second participant, the first portion being controlled by an entity; and	See claim elements [1b] and [1c].
[94c] produce a	See claim elements [1d] and [1e].

second network routing message when at least one of the first participant attributes and at least a portion of the second participant identifier meet a second network classification criterion, the second network routing message identifying an address in a second portion of the packet switched network, the second portion not controlled by the entity.	
95. The system of claim 94, wherein the communication comprises a voice-over-IP communication.	See claim 78.
96. The system of claim 94, wherein the packet switched network is accessed via an Internet service provider.	See claim 79.
98. The system of claim 94, wherein the second network classification criterion is satisfied when access to the second	See claim 89.

participant requires routing through a portion of the packet switched network operated by a communication service supplier.	
99. [99p] A non-transitory computer readable medium comprising instructions that when executed cause a processor to perform a method of routing communications in a packet switched network in which a first participant identifier is associated with a first participant and a second participant identifier is associated with a second participant in a communication, the method comprising:	See claim element [1p].
[99a] after the first participant has accessed the packet switched network to initiate the communication, using the first participant identifier to locate a first participant profile comprising a plurality	See claim element [1a].

of attributes associated with the first participant;	
[99b] when at least one of the first participant attributes and at least a portion of the second participant identifier meet a first network classification criterion, producing a first network routing message for receipt by a controller, the first network routing message identifying an address in a first portion of the packet switched network, the address being associated with the second participant, the first portion being controlled by an entity; and	See claim elements [1b] and [1c].
[99c] when at least one of the first participant attributes and at least a portion of the second participant identifier meet a second network classification criterion, producing a second network routing message for	See claim elements [1d] and [1e].

receipt by the controller, the second network routing message identifying an address in a second portion of the packet switched network, the second portion not controlled by the entity.	

2. Digifonica Release of RBR Version 361

Digifonica would periodically deploy or "roll out" a newer version of the RBR software into production. Version 361 of the RBR software was deployed on June 6, 2005. Ex. 2018 at ¶7. Ex. 2013 at ¶10. Ex. 2012 at ¶11. David Terry sent an email dated: "Mon, Jun 6, 2005 at 9:33 AM" (**Ex. 2026**) indicating that RBR Version 361 had been installed on the package server, and later that day, Emil Bjorsell sent an email dated: "Mon, Jun 6, 2005 at 11:33 AM" (**Ex. 2027**) that was received by both David Terry and Clay Perreault indicating that RBR Version 361 had been deployed to the Vancouver and London supernodes. Ex. 2018 at ¶7. Ex. 2013 at ¶10. Ex. 2012 at ¶11.

Subsequent roll-out and release emails through August, 2005 illustrate the continued operation of the Digifonica system (**Ex. 2030 – 2036, 2042, and 2019**). **Ex. 2018** at ¶¶8-10. **Ex. 2012** at ¶¶12-14. **Ex. 2013** at ¶11. The Subversion repository also indicates continued development of the RBR source code through November 2006. **Ex. 2012** at ¶22. **Ex. 2016** at ¶23.

3. The Smart 421 Engagement

In June 2005 Digifonica engaged Smart 421, a company headquartered in Ipswitch, England, to perform a technical review and appraisal of the Digifonica VoIP application software and development processes. **Ex. 2008** at ¶2. **Ex. 2009** at ¶2. **Ex. 2013** at ¶5. On June 6, 2005 Clay Perreault of Digifonica sent an email to Steve Nicholson and others within Digifonica announcing that "a contract with Smart 421 had been signed …" (**Ex. 2004)**. **Ex. 2013** at ¶5.

The Smart 421 engagement started with document review by Smart 421. On June 6, 2005 and June 15, 2006 Clay Perreault sent emails to Smart 421 employee John Rutter indicating that documents had been uploaded to Smart 421 servers to facilitate Smart 421's review (**Ex. 2005, Ex 2006)**. **Ex. 2013** at ¶6. Mr. Rutter and Stuart Gare, both Smart 421 employees at that time, have reviewed these emails and indicated that they believe them to be accurate. **Ex. 2008** at ¶3. **Ex. 2009** at ¶3.

John Rutter and Stuart Gare visited Digifonica in June 2005 and witnessed the Digifonica system in operation. They saw that the system was able to place phone calls between two SIP phone devices, on the same or different nodes, and between a SIP phone device and the PSTN network. **Ex. 2008** at ¶4. **Ex. 2009** at ¶4. **Ex. 2013** at ¶7.

John Rutter and Stuart Gare prepared a 35-page report entitled “TECHNICAL REVIEW OF DIGIFONICA VOIP SYSTEM” dated July 5, 2005 (“Smart 421 Report”) (**Ex. 2003)**. The Smart 421 Report (**Ex. 2003**) was emailed in PDF form by John Rutter to Clay Perreault on July 5, 2006, with Stuart Gare also copied on the email, and Clay Perreault subsequently emailed the report to others including Emil Bjorsell (**Ex. 2007)**. **Ex. 2008** at ¶6. **Ex. 2009** at ¶6. **Ex. 2013** at ¶8. **Ex. 2012** at ¶7.

The Smart 421 report (**Ex. 2003**) confirms in numerous places that Digifonica had at that time a working, or “live” system running what is referred to as “Version 1” and was developing “Version 2”.

> “The core code appears to be very well written and **has been tested in live operation** and destructive testing by developers over a period of time.” [**Ex. 2003** at 5, emphasis added]
>
> “**Version 1 is the historical development path leading to the current live system**, and Version 2 is a newer development path that has been implemented in recent months to include more formal measures against software deliveries.” [**Ex. 2003** at 9, emphasis added]

> "This also provides the opportunity for further documentation and other quality control measures to be applied, without the overhead of enforcing this additional work on **the phase 1 system that is currently in operation**." [**Ex. 2003** at 15, emphasis added]

> "This approach was partly down to issues of resources and costs, **getting a very functional system operational** with a strong understanding of the underlying technology and network integration issues of a VoIP solution." [**Ex. 2003** at 20, emphasis added]

> "**With a live system in operation**, the need for strict release controls and quality assurance is increased to avoid potential service disruption. In recent weeks, Digifonica have filled further positions in delivery management and these issues are under control for the Version 2 development, as well as for any **maintenance releases required against the live Version 1 software**." [**Ex. 2003** at 21, emphasis added]

The authors of the Smart 421 Report (**Ex. 2003**), John Rutter and Stuart Gare, stated in attached declarations that the quoted passaged above accurately reflected their review of the Digifonica system at the time the report was written. **Ex. 2008** at ¶7. **Ex. 2009** at ¶7.

Because Version 361 had been deployed to the production system on June 6, 2005, before Smart 421's visit, it is certain that the system that Mr. Rutter and Mr.

Gare saw demonstrated contained all of the features analyzed above in connection with Version 361 of the RBR software. **Ex. 2012** at ¶11. **Ex. 2013** at ¶10.

4. Inventor and Employee Testimony

Further corroborating the fact that the Digifonica system running RBR Version 361 was working for its intended purpose is inventor and other employee testimony. David Terry, a software engineer employed by Digifonica at the time has explained the features of the Digifonica system. **Ex. 2018** at **¶¶3**-4. Specifically, Mr. Terry describes the operation of the Digifonica system as follows:

> "The Digifonica system functioning in June 2005 included hardware and software that established a user-specific profile for each user containing attributes such as international dialing digits (IDD), national dialing digits (NDD) and area code. When a call was placed, the caller identifier was used to locate the profile associated with the caller containing these attributes from a database. The caller attributes were then matched against the callee identifier (the dialed digits) to create a reformatted callee identifier. Based on the reformatted callee identifier and a lookup of that number in a database of Digifonica subscribers, a call was classified as a private, or on-net call if the destination was another Digifonica subscriber, and classified as a public, or off-net call if the destination was the PSTN. The Digifonica system was capable of classifying a call as an on-net call after a user had dialed a PSTN number by first matching the dialing string according to the caller's profile, and then checking to see if the destination number was mapped to a Digifonica IP phone.

> Once a call was classified, appropriate routing messages were generated so that a call controller could direct the call according to the classification of the call. All of these features were incorporated into the Digifonica system that was deployed and fully operational by June 2005." **Ex. 2018** at ¶4.

Mr. Terry also states that: "I'm certain that Version 361 of the RBR software was in operation on the production system on June 6, 2005 and successfully performed the call routing functions described above in paragraph 4 on that date." **Ex. 2018** at ¶7. Dr. Mangione-Smith has reviewed the Declaration of Mr. Terry and states that this description confirms that the RBR software was deployed in a system that operated consistent with his analysis of Version 361 and this further confirms the fact that Version 361 was functional and performing all limitations recited in the challenged claims of the '005 Patent. **Ex. 2016** at ¶28-29.

Two of the inventors of the '005 Patent, Mr. Johan Emil Viktor Bjorsell and Mr. Clay Perreault also confirm that the Digifonica system was in operation at least by June 6, 2005. **Ex. 2013** at ¶13. **Ex. 2012** at ¶21. Mr. Bjorsell was responsible for software development, systems engineering, testing and deployment at Digifonica from July 2004 through October 2008 and has personal knowledge of the Smart 421 report, the RBR software releases and the use of the source code repository. **Ex. 2012** at ¶¶4, 7, 22. Mr. Perreault was a Founder and Chief Technology Officer of Digifonica through 2005 and has personal knowledge of the visit by Smart 421, the Smart 421 report, and the operation of the RBR server. **Ex. 2013** at ¶¶7, 10, 11, 13, 14. The testimony of these inventors confirms that Digifonica had a live system in operation in June 2005, and that Version 361 of the RBR software was released and was in operation on June 6, 2005. **Ex. 2013** at ¶13. **Ex. 2012** at ¶21. Thus, the Bjorsell and Perreault Declarations confirm that Version 361 was working for its intended purpose and performing all limitations recited in the challenged claims of the '005 Patent.

As demonstrated above, the claims of the '005 Patent challenged in the Petition were practiced by the system that included the RBR call routing platform that was operating at least as early as June 6, 2005. Thus, the inventor's actual reduction to practice preceded the filing date of Chu '366 of August 4, 2006. Accordingly, Chu '366 is not prior art under 35 U.S.C. § 102(e) and Ground 1 must be rejected.

B. CHEN IS NOT PRIOR ART UNDER PRE-AIA 35 U.S.C. 102(e)

In Ground 2, Petitioner asserts that "U.S. Patent Publication No. 2007/0064919 to Chen at al. ("Chen") was filed on Sept. 14, 2005 and therefore qualifies as prior art with regard to the '005 Patent under 35 U.S.C. §102(e)." Petition at 36. However, pre-AIA 35 U.S.C. §102(e) establishes that a reference is prior art if it is "filed in the United States before the invention by the applicant" and Chen was not filed before the invention by the inventors of the '005 Patent.

Prior invention can be established by an actual reduction to practice before the priority date. *Eaton v. Evans*, 204 F.3d 1094, 1097 (Fed. Cir. 2000). The inventors of the ‘005 Patent reduced the claimed subject matter to practice before Chen’s filing date of September 14, 2005.

As demonstrated above, the claims of the ‘005 Patent challenged in the Petition were practiced by the system that included the RBR call routing platform that was operating at least as early as June 6, 2005. Thus, the inventor’s actual reduction to practice preceded the filing date of Chen of September 14, 2005. Accordingly, Chen is not prior art under 35 U.S.C. § 102(e) and Ground 2 must be rejected.

C. PETITIONER FURTHER FAILS TO ESTABLISH THAT THE CHALLENGED CLAIMS ARE OBVIOUS

Even if available as prior art, no combination of the cited references renders the challenged claims obvious.

1. Claimed Subject Matter

Exemplary Claim 1 is provided below. Note this Response numbers claim elements differently than the Petition in order to conform to the higher level of specificity of the antedating tables *supra* at II(A)(1) and in the Mangione-Smith Declaration (**Ex. 2016**). Antedating table numbering is denoted by **square brackets [ ]**; Petition numbering is shown in **parentheses ( )**.

> 1. **(1p)[1p]** A process for producing a routing message for routing communications between a caller and a callee in a communication system, the process comprising:
>
> **(1a)[1a]** using a caller identifier associated with the caller to locate a caller dialing profile comprising a plurality of calling attributes associated with the caller;
>
> **(1b)[1b]** when at least one of said calling attributes and at least a portion of a callee identifier associated with the callee meet private network classification criteria, **[1c]** producing a private network routing message for receipt by a call controller, said private network routing message identifying an address, on the private network, associated with the callee; and
>
> **(1c)[1d]** when at least one of said calling attributes and at least a portion of said callee identifier meet a public network classification criterion, **[1e]** producing a public network routing

> message for receipt by the call controller, said public network routing message identifying a gateway to the public network.

The method of Claim 1 establishes private and public network classification criteria and produces routing messages to route a call to a "public network" or a "private network" based on at least one calling attribute and at least a portion of the callee identifier. For example, when a caller initiates a call to a callee, the call may be routed to the PSTN or over the Internet, based on a calling attribute and at least a portion of callee information. The method does not evaluate the callee identifier in isolation, but establishes criteria based on at least one attribute in the caller's dialing profile, to make a network classification decision, e.g., PSTN or Internet routing.

Petitioner directed its analysis almost entirely to Claim 1, therefore the Patent Owner's arguments focus on Claim 1, but will apply to other claims *mutatis mutandis*. Where additional arguments are applicable to the distinct subject-matter of the other claims, this is explained separately.

2. Overview Of Cited Art

a. Overview of Chu '684

Chu '684 discloses an architecture for providing voice-over-IP virtual private network (VoIP VPN) services to an organization/enterprise ("subscriber") with multiple IP-PBXs, and a method of connecting the organization's IP-PBXs into a single logical network. Chu ’684 at 1:44-46, 3:52-56. The enterprise subscribes to data and voice services from the service provider (SP). *Id.* at 5:3-6. FIG. 2 illustrates a subscribing customer’s IP-PBX (i.e., multiple phones and a server 110 located at the subscribing customer’s premises 105), which is configured to communicate with a soft-switch 220 and packet switch 210 located at the SP’s central office 205:

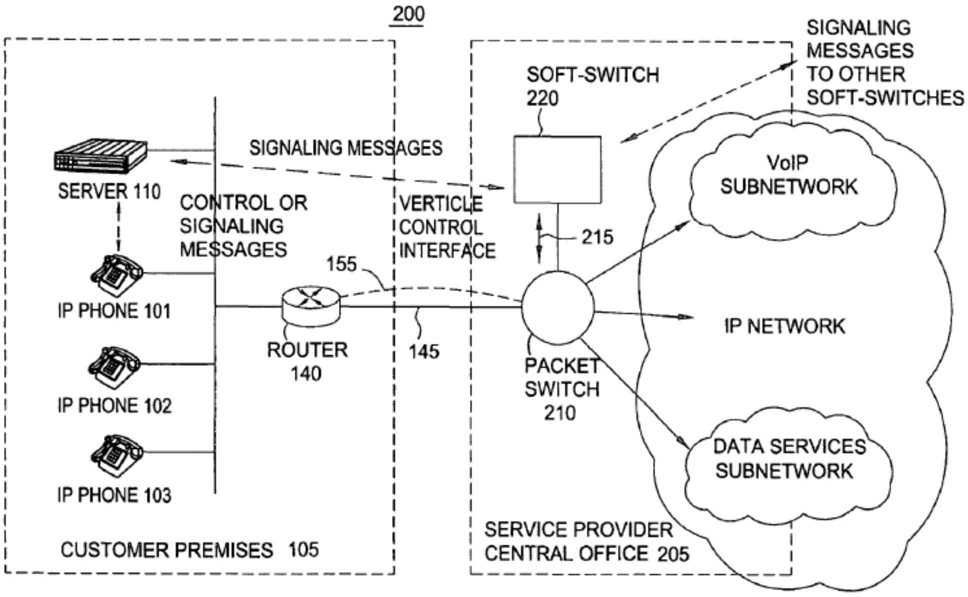


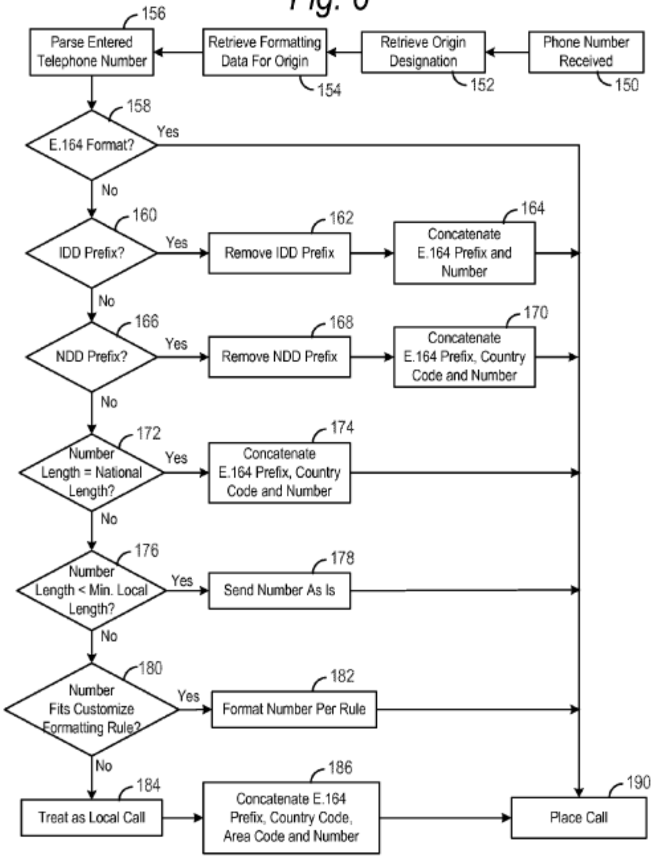
FIG. 2

While many enterprises, each with multiple premise locations (e.g., 105), may share the SP’s network, each enterprise may use its own IP addressing scheme and “dial plan,” allowing calls to be routed internally within the IP-PBX, to a different IP-PBX, and to the public switched telephone network (PSTN). *Id.* at 2:17-22; 3:61-67; 12:59-67 and 8:65-9:1. Users direct calls to the private network using "private numbers" from the enterprise's "private numbering plan." Users direct calls to the public PSTN using "public telephone numbers" from the public E.164 numbering plan. *Id.* at 9:16-17, 16:50-54, 13:8-9; **Ex. 2016** at ¶ 72 and ¶¶ 30-43.

b. Overview of Chu '366

Chu '366 discloses a method of formatting a dialed telephone number according to the E.164 standard based on a "call origin location profile." Chu '366 at 1:62-2:14. A dialed public PSTN number in Chu '366 can be formatted into the E.164 format based on the PSTN dialing conventions of a variety of geographic locations. *Id.* at 2:16-28 and FIG. 6. Chu '366's method allows travelling users who initiate VoIP telephone calls from different locations, to use the dialing patterns of the geographic location from which they are dialing. *Id.* at 5:3-14. Thus, the formatting described in Chu '366 depends on the caller's geographic location at the time the call is placed.

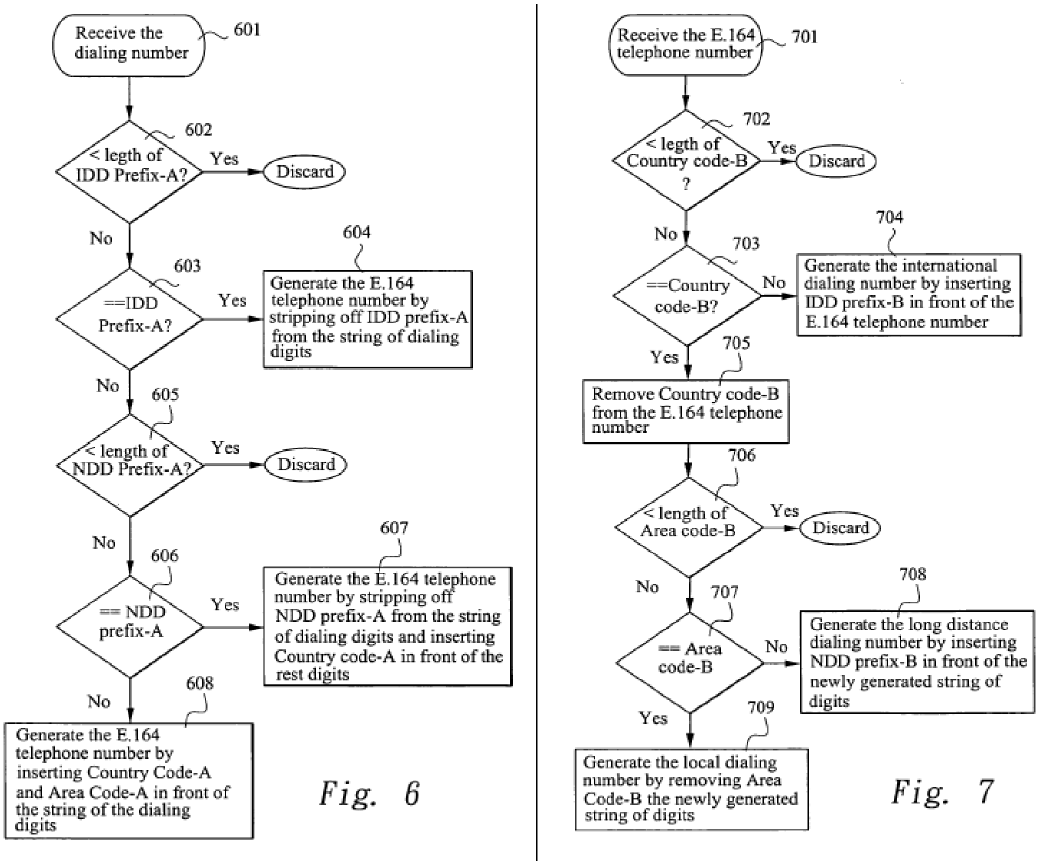
Fig. 6


156
Parse Entered Telephone Number
Retrieve Formatting Data For Origin
154
Retrieve Origin Designation
152
Phone Number Received
150
158
E.164 Format?
Yes
No
160
IDD Prefix?
Yes
162
Remove IDD Prefix
164
Concatenate E.164 Prefix and Number
No
166
NDD Prefix?
Yes
168
Remove NDD Prefix
170
Concatenate E.164 Prefix, Country Code and Number
No
172
Number Length = National Length?
Yes
174
Concatenate E.164 Prefix, Country Code and Number
No
176
Number Length < Min. Local Length?
Yes
178
Send Number As Is
No
180
Number Fits Customize Formatting Rule?
Yes
182
Format Number Per Rule
No
184
Treat as Local Call
186
Concatenate E.164 Prefix, Country Code, Area Code and Number
190
Place Call

Chu ’366 - FIG. 6

c. Overview of Chen

Chen’s method enables traveling users to place calls via local Telco switches in foreign regions or countries using a *fixed* dial plan, by translating dialed numbers from the fixed dial plan familiar to the user (“dial plan A”) to a different dial plan (“dial plan B”) not familiar to the user but understood by the local PSTN Telco switch of the foreign region or country being visited. Chen at [0002], [0014], [0025] and [0026]. The method involves: receiving a dialing PSTN number; transforming this number into an E.164 format number (*see* FIG. 6 of Chen); and transforming the E.164 number into a final dialing number understood by the Telco switch (FIG. 7). *Id.* at [0016]; *see also id.* claim 1. Thus, like Chu ’366, the formatting described in Chen depends on the caller’s geographic location at the time the call is placed.


Receive the dialing number
601
< legth of IDD Prefix-A?
602
Yes
Discard
No
==IDD Prefix-A?
603
Yes
Generate the E.164 telephone number by stripping off IDD prefix-A from the string of dialing digits
604
No
< length of NDD Prefix-A?
605
Yes
Discard
No
== NDD prefix-A
606
Yes
Generate the E.164 telephone number by stripping off NDD prefix-A from the string of dialing digits and inserting Country code-A in front of the rest digits
607
No
Generate the E.164 telephone number by inserting Country Code-A and Area Code-A in front of the string of the dialing digits
608
Fig. 6
Receive the E.164 telephone number
701
< legth of Country code-B ?
702
Yes
Discard
No
==Country code-B?
703
No
Generate the international dialing number by inserting IDD prefix-B in front of the E.164 telephone number
704
Yes
Remove Country code-B from the E.164 telephone number
705
< length of Area code-B
706
Yes
Discard
No
== Area code-B
707
No
Generate the long distance dialing number by inserting NDD prefix-B in front of the newly generated string of digits
708
Yes
Generate the local dialing number by removing Area Code-B the newly generated string of digits
709
Fig. 7

Chen – FIGS. 6 and 7

D. PETITIONER'S COMBINATIONS FAIL TO PROVIDE "PRODUCING..." WHEN AN "ATTRIBUTE" AND A "CALLEE IDENTIFIER" MEET "NETWORK CRITERIA"

The cited references, individually or in combination, fail to provide "producing a... routing message" when at least one of the "calling attributes" and at least a portion of the "callee identifier" meet "network classification criteria" as recited in claim steps [1b], [1d], [26b], [26c], [50b], [50c], [74b], [74d], [94b], [94c], [99b] and [99c], which correspond to (1b-c), (26b-c), (50b-c), (74b-c), (94b-c) or (99b-c) in the Petition.

1. The proposed combinations fail to provide at least one element in each claim

The cited references, individually or in combination, fail to provide any teaching or suggestion of establishing call classification by network (i.e., private network or public network) based on the caller's "calling attributes" and the "callee identifier" in the manner recited in the challenged claims.

The claims do not rely solely on the dialed digits (e.g., dialing "9" to indicate that a public number is being dialed), as in traditional PBX systems, but also use "calling attributes" from a caller's dialing profile and at least part of the identifier for the called party ("callee"), and establish criteria for routing the call to the appropriate network. The caller's "attributes" can include information such as international dialing digits (IDD), minimum local length, etc. (see, e.g., '005 Patent at Fig. 9).

In contrast, Chu ‘684 determines the network (i.e., public network or private network) based solely on the type of numbers dialed by the caller (i.e., public or private numbers). Chu ‘684 at 8:65-9:1; *see also* 9:16-17, 16:39-54 and 13:7-9. Chu ‘684 does not decide on the network based on whether information in the caller’s profile meets private or public classification criteria as claimed. Combination of Chu ‘684 with Chu ‘366 or Chen also fails to teach the claimed features.

Petitioner’s claim chart asserts that it would have been obvious to modify Chu ‘684 to *first* reformat a dialed number (i.e., callee identifier) according to Chu ‘366 or Chen, *and then*, “***[o]nce the callee identifier is reformatted***,” to continue processing the *reformatted number* by Chu ‘684 (e.g., step 608) to determine whether the call is local to the PBX, on-net to another PBX in the VPN, or off-net to a PSTN phone. Petition at 18-20, 26-28, 43-44 and 50-52.

The proposed combinations fail to establish a *prima facie* case of obviousness because they fail to provide at least elements [1b], [1d], [26b], [26c], [50b], [50c], [74b], [74d], [94b], [94c], [99b] or [99c], which correspond to (1b-c), (26b-c), (50b-c), (74b-c), (94b-c) or (99b-c) in the Petition, as explained below.

Chu ‘684 describes an enterprise PBX system that supports the use of both public numbers from a “public numbering scheme” and private numbers from a “private numbering scheme”, as typical in known PBX systems. Chu ‘684 at 9:16-

17, 16:39-54, and 13:7-9; *see also* **Ex. 2016** at ¶¶ 42 and 78. In such a system, a user would dial private numbers to place a call to the PBX private network, or a PSTN access code (e.g., a prefix of "9") followed by a PSTN number based on local dialing conventions to call the PSTN. A "private numbering scheme" is defined by the PBX administrator. Typically, the "private numbers" started with a digit other than the PSTN prefix to allow the PBX to unambiguously distinguish the type of number being dialed (private or public); the use of prefix digits to dial PSTN calls was ubiquitous in the field of PBX systems by the priority date of the '005 Patent. *See* **Ex. 2016** at ¶ 72 and ¶¶30-47.

Chu '684 provides only a single sentence describing step 608, relied upon by Petitioner as providing the network classification criteria. Chu '684 at 8:65-9:1. Absent an express teaching about how step 608 would work in Chu '684, a PHOSITA would rely on the aforesaid well-known PBX practices, namely: in step 608, the server 110 determines whether the dialed number is a private number or a public PSTN number and routes the call accordingly via a private network or a PSTN. **Ex. 2016** at ¶ 72 and ¶¶ 30-47.

In step 608, a PHOSITA would expect the server 110 to determine that a called number is a PSTN public number if the dialed digits start with the PSTN prefix (e.g., "9"), and if so, send the called number to the soft-switch 220 with an indication that the called number follows the "public E.164 number plan". Chu

‘684 at 9:16-17 and 5:18; **Ex. 2016** at ¶ 72. The soft-switch identifies an egress PSTN gateway to carry the PSTN call. Chu ‘684 at 13:18-20.

However, determining that the received dial string starts with a PSTN prefix, in the modified Chu ‘684 system, is not meeting a “public network classification criterion” as recited in claim element [1d] because the prefix digit is not an “calling attribute”. **Ex. 2016** at ¶ 37. Rather, the PSTN prefix is a system-wide setting. *Id*. Thus, even if step 608 classified the call as a public network call, the classification is not based on at least one “calling attribute” as recited by element [1d] (corresponding to element (1c) in the Petition).

In addition, Chu ‘684, alone or in combination with Chu ‘366 or Chen, does not disclose classifying a dialed PSTN number as a “private network call”. **Ex. 2016** at ¶ 72. Rather, it was well-known in the art of PBX systems for any dialed numbers identified by a PSTN prefix (e.g., “9”) to be passed to a central office switch (e.g., soft-switch 220 at central office 205) for processing, and the modified Chu ‘684 system proposed by the Petitioner would do the same. Chu ‘684 at FIG. 2; **Ex. 2016** at ¶¶ 30-36. Because step 608 sends numbers preceded by a PSTN prefix to the central office switch, they would not be routed as a “private network call”. Thus, step 608 in Petitioner’s proposed combinations fails to provide meeting “private network classification criteria” based on at least one of the "calling attributes" as recited in step [1b] (corresponding to step (1b) in the Petition).

In summary, the proposed combinations of Chu '684 with Chu '366 or Chen, fails to classify as recited in the challenged claims by utilizing a "calling attribute" to establish meeting "public network classification criteria" or a "private network classification criterion" as recited in steps [1b] and [1d], which correspond to (1b) and (1c) in the Petition.

2. A PHOSITA would not follow the order of steps set forth in the Petition to combine the teachings of the references.

Petitioner proposes modifying Chu '684 by inserting the *public* number reformatting method of Chu '366 or Chen before the classification step 608 in Chu '684, which processes both *public* <u>*and*</u> *private* numbers. *See supra* II(D)(1); **Ex. 2016** at ¶¶ 76-82. Far from being obvious, the specified order of steps for combining reformatting and classification in the Petition is constructed based on hindsight, and not any teaching of the cited references, and is susceptible to unreliable operation, as Petitioner's own declarant recognized during deposition. *In re Gordon*, 733 F.2d 900 (Fed. Cir. 1984) (supporting a non-obviousness conclusion if a reference's device, when modified as proposed, "would be rendered inoperable for its intended purpose").

Standing alone, Chu '684 describes that a dialed private number is received (unaltered) by the server 110 in step 608. Chu '684 at 8:65-9:1. The server determines from the private number that this is a PBX private network call. *Id.* at 8:67. The private number identifies a "local" destination (e.g., between phones at customer premises 105 in FIG. 2) or a phone at another premises 806 (e.g., in FIG. 8), which is considered "on-net" within the VPN. *Id.* at 5:18, 10:45-50, and FIG. 8. A SIP "invite" message is sent to soft-switch 220 at the central office 205, with the called number, and an indication that it follows the "private numbering plan" for the enterprise. *Id.* at 9:2-4 and 9:16-17. The soft-switch 220 would setup the call over the service provider (SP) network. *Id.* at 9:1-11:62.

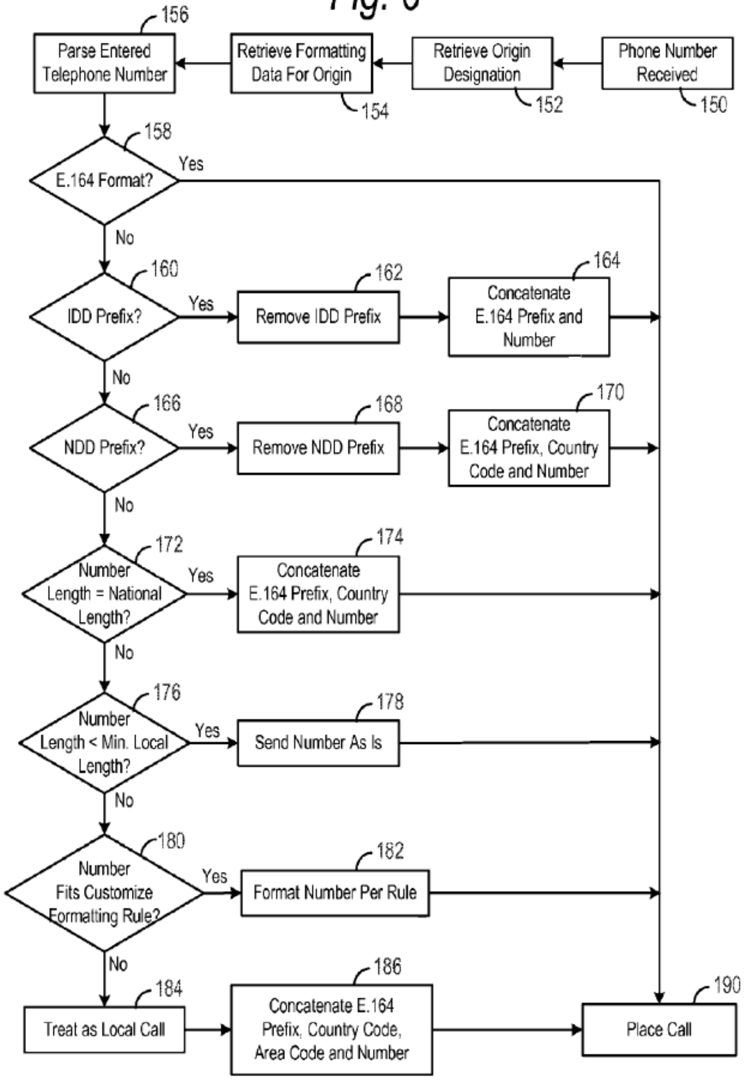
In Petitioner's proposed combinations, however, it is *unpredictable* what step 608 would do. For example, if a private number was fed into the reformatting algorithm of Chu '366 (see FIG. 6) and matched any of steps 160, 166 or 172 of Chu '366, the following step (162, 168, or 174) would corrupt the private number. **Ex. 2016** at ¶¶ 80-82. How Chu '684's step 608 would handle an invalid quantity is not disclosed, but a skilled person would understand that the call would not complete as dialed. *Id.*

Petitioner's simplistic manner of combining the references effectively causes *all* dialed digits to be fed into a reformatting algorithm designed for *public* numbers irrespective of whether they are *public* or *private* numbers. *Id*. However, Chu '366 and Chen teach reformatting processing for only PSTN *public* numbers, and do not even mention private networks or numbers. *Id*. A skilled person would know that private numbers cannot be reformatted by the methods of Chu '366 or Chen because they do not follow PSTN public number dialing/formatting conventions. *Id*. The proposed combinations ignore the common knowledge of the skilled person that private numbers in Chu '684 *must* be handled differently from public numbers, namely, by <u>not</u> reformatting the former to avoid invalid routing. *Id*.

Petitioner's Declarant addressed some of the problems arising from the Petition's alleged order-of-steps during his deposition. In analyzing certain hypothetical scenarios, Dr. Houh found it necessary to adopt new theories (not present in the Petition or Declaration) of how the references would be combined, justified by nothing more in the record than conclusory statements based on hindsight about what a skilled person would know to do. **Ex. 2044** at 133:15-156:5; **Ex. 2016** at ¶ 67 and ¶¶ 76-82.

First, Dr. Houh was asked to analyze the combination of Chu '684 and Chu '366, for a *private* number starting with "1" (extension "101"). **Ex. 2044** at 133:15-143:8. Dr. Houh's attention was directed to the reformatting algorithm taught in Fig. 6 of Chu '366 (shown below).

Fig. 6


156
Parse Entered Telephone Number
Retrieve Formatting Data For Origin
154
Retrieve Origin Designation
152
Phone Number Received
150
158
E.164 Format?
Yes
No
160
IDD Prefix?
Yes
162
Remove IDD Prefix
164
Concatenate E.164 Prefix and Number
No
166
NDD Prefix?
Yes
168
Remove NDD Prefix
170
Concatenate E.164 Prefix, Country Code and Number
No
172
Number Length = National Length?
Yes
174
Concatenate E.164 Prefix, Country Code and Number
No
176
Number Length < Min. Local Length?
Yes
178
Send Number As Is
No
180
Number Fits Customize Formatting Rule?
Yes
182
Format Number Per Rule
No
184
Treat as Local Call
186
Concatenate E.164 Prefix, Country Code, Area Code and Number
190
Place Call

When asked if step 166 (matching the NDD) would lead to corruption of "101" in steps 168-170, Dr. Houh responded that a skilled person would know how to avoid corruption by testing the number of digits dialed *before* step 166. *Id*. at 139:8-21 (otherwise it "would not make sense") and 140:8-12. Dr. Houh tried to justify testing for length by citing Chu '366's teaching of determining number length in column 10 (*id*. at 10:4-7), but that teaching describes testing length in a different context (i.e., step 172, which comes *later* in the algorithm). **Ex. 2044** at 138:5-24. Dr. Houh proposed inserting a new length test prior to step 160, simply to circumvent corruption, thereby departing from the algorithm *actually* taught by Chu '366. *Id*. at 138:20-139:25; **Ex. 2016** at ¶ 67.

Dr. Houh was next asked to analyze German-style dialing (which allows 4-digit PSTN numbers) in a PBX system with 4-digit extensions. **Ex. 2044** at 143:11-149:14. Dr. Houh recognized that, in this scenario, the "length test" he had added was inadequate to avoid dialing errors (e.g., 4-digit extensions would be misclassified and corrupted in steps 172-174). Chu '366 at FIG. 6; *see also* **Ex. 2044** at 144:1-19. Again, Dr. Houh found it necessary to propose new theories, namely, that a PBX manufacturer might *prohibit* using 4-digit extension numbers, or would use a "different dial plan" requiring the user to dial a prefix digit (e.g., "9" or "8") to indicate to the PBX that subsequent digits would be a PSTN number or a private number. *Id*.

Dr. Houh then analyzed Chu '684 combined with Chen. *See* **Ex. 2044** at 149:15-156:9. FIG. 6 (shown below) of Chen shows that private numbers would either be *discarded* or invalidly *reformatted* invalidly in numerous steps. *Id*; *see also* **Ex. 2044** at 151:8-19.

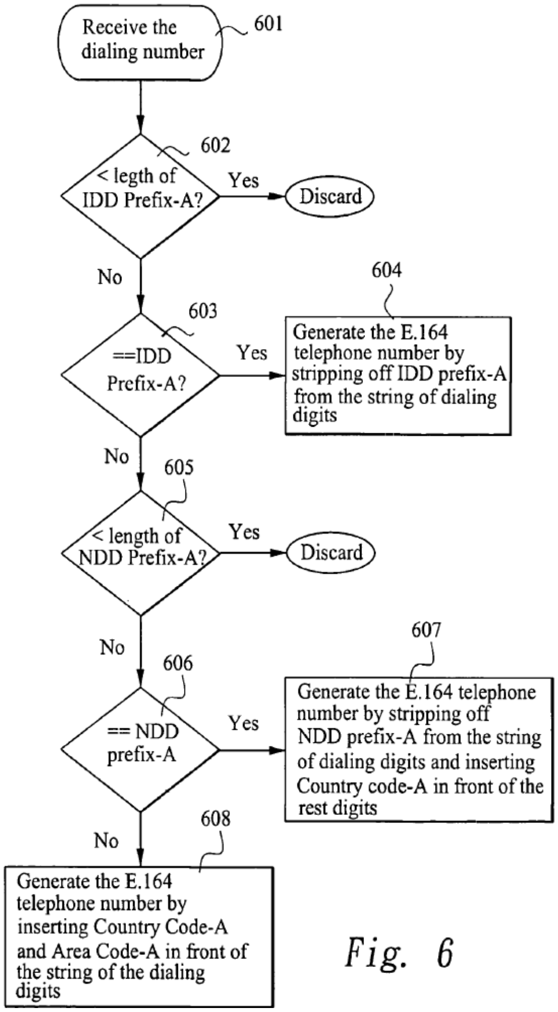

Receive the dialing number
601
602
< legth of IDD Prefix-A?
Yes
Discard
No
604
603
==IDD Prefix-A?
Yes
Generate the E.164 telephone number by stripping off IDD prefix-A from the string of dialing digits
No
605
< length of NDD Prefix-A?
Yes
Discard
No
607
606
== NDD prefix-A
Yes
Generate the E.164 telephone number by stripping off NDD prefix-A from the string of dialing digits and inserting Country code-A in front of the rest digits
No
608
Generate the E.164 telephone number by inserting Country Code-A and Area Code-A in front of the string of the dialing digits

Fig. 6

Dr. Houh maintained that a skilled person could make a combination with Chu ‘684 and Chen that “preserve[s] the private number to -- to the system in step 608 [of Chu ‘684]”. *Id*. at 152:9-10. But in order to explain *how*, Dr. Houh proposed sweeping changes to Chen’s algorithm, selectively adopting only certain *portions* thereof (“the upper algorithm” *id*. at 152:25) and adding new steps not taught by Chen, including an *ad hoc* “number length” test (*id*. at 153:1-5) and a conditional pass-through of private numbers (*id*. at 154:8-155:2). **Ex. 2016** at ¶ 67.

Dr. Houh's new combinations of Chu '684 with Chu '366 or Chen, (1) do not lend support to Petitioner's manner of combining the art, (2) lack support in the teachings of the asserted references themselves, and (3) render the proposed system deficient for its intended purpose of accurate and reliable call routing.

These factors clearly belie the Petition's assertion that its combinations would be obvious to a skilled person. Indeed, the very *need* for Dr. Houh to devise numerous *ad hoc* solutions to various conflicts between the use of private and public numbers in Petitioner's proposed combinations, strongly urges the conclusion that Petitioner's manner of combining the elements would not be obvious.

Dr. Houh was asked about whether using a PSTN prefix digit (e.g., "9") would be a solution to allow a PBX extension ("private number") to pass through the reformatting step without being changed or misinterpreted. Dr. Houh agreed that it would be easy to reformat a public number *after* it was identified by a prefix stating, "If the user dialed nine, the rest of the digits could be easily passed through the algorithm of the Chen". **Ex. 2044** at 155:8-156:4. While this manner of combining the references (e.g., reformatting *after* step 608 in Chu '684) is clearly superior to the contorted and artificial ordering of Petitioner's combinations, it does not practice the method of the challenged claims.

Accordingly, a PHOSITA would not follow the cited references to perform the steps in the order set forth in the Petition to combine the references in a way that teaches the challenged claims.

E. PETITIONER'S PROPOSED COMBINATIONS FAIL TO LOCATE ATTRIBUTES ASSOCIATED WITH THE CALLER

1. The Petition fundamentally misinterpreted the dial plans of Chu '684 as being *user*-specific instead of *enterprise*-specific

Petitioner misinterpreted "subscriber" in Chu '684 as denoting an *individual caller/user*. Chu '684, however, uses "subscriber" to refer to an *enterprise*. This led Petitioner and its Declarant to fundamentally misinterpret a *subscriber's* "dial plan" in Chu '684 as being *user*-specific.

During deposition, Petitioner's expert, Dr. Houh, realized this error on the meaning of "subscriber" in Chu '684, and repeatedly admitted that "subscriber" in Chu '684 means "enterprise" including Chu '684 at 3:55-56, 3:61-64, 8:12-16, 9:30-33, 12:15-16, 12:55-60, 12:60-64, and 14:53-61. *See* **Ex. 2044** at 221:20-222:4, 220:18-24, 178:17-181:4, 223:8-224:8, 215:20-216:6, 214:1-215:19; 217:10-23 and 218:1-220:9, respectively. Dr. Houh acknowledged that "subscriber" in Chu '684 has a distinct meaning from "subscriber" in the parent '815 Patent, where "subscriber" means "user". **Ex. 2043** at 16:11-19.

Dr. Houh's admissions confirm that Chu '684 discloses an *enterprise* dial plan, <u>not</u> a user's dial plan. *See* Chu '684 at 3:55-58 ("VPN service connects all the

IP-PBXs of a subscriber [i.e., enterprise] into a single logical network... where subscribers can use their own internal dial plan"); 9:30-33 ("the soft-switch 220 consults the dial plan for this subscriber [i.e., enterprise]... determined from the ID of the server 110"); 12:62-64 ("Each subscriber [i.e., enterprise] can use their the own IP address plan as well as their own dial plan." [sic]).

2. Chu '684's "dial plan" is *enterprise*-specific, not *user*-specific, which undercuts Petitioner's obviousness theories

Petition's arguments are all premised on its fundamental misinterpretation that Chu '684's "subscriber" is an individual user/caller, which led to the Petitioner and its Declarant to erroneously assert that Chu '684's "dial plan" is *user*-specific, rather than *enterprise*-specific. Petition at 11, 37, 18, 42. Houh Declaration at ¶45.

As discussed below, this mischaracterization of Chu '684's "dial plan" undermines Petitioner's obviousness arguments in numerous aspects including the Petitioner's analysis of claim elements, the feasibility of combining the references, and the motivation to even *make* the combinations.

a. Consulting an *enterprise* "dial plan" in Chu '684 is distinct from locating a "*caller* dialing profile"

The Petition's claim chart equates "locate a caller dialing profile" in, e.g., claim [1a] (corresponding to element (1a) in the Petition), with "Chu '684 teach[ing] locating a subscriber's dial plan". Petition at 17, 21, 23, 26 31, and 34. But this argument is premised Petitioner's initial misinterpretation of "subscriber" in Chu '684. Once this false premise is removed, the claim chart no longer meets the corresponding claim element. Petitioner's alleged disclosure in Chu '684 of "locating" a subscriber's dial plan would be "locating" an *enterprise's* dial plan. **Ex. 2016** at ¶ 64. An *enterprise* is not a *caller*; therefore "locating a subscriber's dial plan" in Chu '684 is not locating "a caller dialing profile" as recited in elements [1a], [26a], [50a], [74a], [94a] and [99a], which correspond to (1a), (26a), (50a), (74a), (94a), and (99a) in the Petition. *Id.*

Chu '366 and Chen provide no teaching that would instruct one of ordinary skill in how to modify Chu '684's enterprise dial plan to become a *user*-specific profile. *Id.* Indeed, by its very nature, the "dial plan" cannot become a *user*-specific profile because it is intended to be shared by *all* users of the PBX. *Id.*

Thus, Petitioner's fundamental misinterpretation of "subscriber" in Chu '684 as an individual user undermines the Petitioner's arguments premised on this misinterpretation, namely, its argument for elements [1a], [26a], [50a], [74a], [94a] and [99a], corresponding to (1a), (26a), (50a), (74a), (94a), and (99a) in the Petition. At least these elements are missing from any proposed combination.

b. Chu '684's enterprise "dial plan" cannot be combined with individualized profiles

As discussed *supra* II(E)(1), Chu '684 discloses that a "dial plan" is shared by a group of users. Chu '366 discloses *user*-specific "call origin location profiles" and Chen discloses a *user's* fixed dial plan. **Ex. 2016** at ¶ 57. The record is silent as to how to combine *caller*-specific *individualized* profiles with an *enterprise's* IP-PBX network-specific "dial plan." These are fundamentally different entities and cannot be combined. *Id*.

An enterprise "dial plan" is primarily designed to serve the needs of the *enterprise*, e.g., to implement and enforce a common telephony policy and private numbering plan. *Id*. Petitioner's arguments about how straightforward it would be to combine the teachings were premised on a misapprehension of *what* was being combined. Chu '684's "dial plan" is not a user "dial plan". *Supra* II(E)(1). Under a proper understanding of Chu '684's enterprise "dial plan", Petitioner's basis for combining Chen/Chu '366 with Chu '684 fails – the teachings of Chu '684 and Chen/Chu '366 are incompatible. **Ex. 2016** at ¶¶ 57 and 88.

F. PETITIONER FAILS TO ARTICULATE A PROPER REASON TO COMBINE AND OVERLOOKS WHY THE COMBINATIONS ARE UNDESIRABLE

1. No articulated reasoning for reason to combine

Petitioner fails to provide articulated reasoning with rational underpinning to support the legal conclusion of obviousness. *KSR Int'l. Co. v. Teleflex, Inc.*, 550 U.S. 398, 418 (2007).

The Petition refers mostly to irrelevant factors (e.g., both references route calls to the PSTN) incapable of providing a proper motivation to combine the references, before providing a single, conclusory sentence of motivation alleging that Chu '684 does not allow users "to place calls as if they were dialing from a standard PSTN phone". Petition at 16 and 40. The Petition does not cite Chu '684 or explain how Chu '684 possesses this deficiency. Such conclusory statements, even if parroted by an expert, are inadequate. *In re NuVasive, Inc.* 842 F.3d 1376, 1384 (holding that a declarant's asserted "'uses [that] were not disclosed in the cited prior art references'" were insufficient to address either "the benefits that could have been obtained by combining the prior art references []or the PHOSITA's motivation to combine at the time of the invention.").

In deposition, the Petitioner's Declarant shifted to a different reason to combine: "the call origin profile allows the user to do, for example, seven-digit dialing... in places where ten-digit dialing is--is required." **Ex. 2043** at 26:12-15 & 27:1-3. This justification finds no basis in Chu '684, and involves the user dialing according to rules that are in *conflict* with PSTN dialing. **Ex. 2016** at ¶ 66. Such a justification is inconsistent with Petitioner's original "reason" to combine, and fails to address any actual defect in Chu '684. *Id.* As such, it is not a proper reason to combine references.

Declarant's shifting positions on *how* to combine the references undermines Petitioner's assertion that the claims are merely a simple combination of the references. Petition at 16, 40-41. The references themselves provide no such combination, as evidenced by Declarant's shifting theories unsupported by the references.

2. No reason to reformat numbers in Chu '684

The reasons for reformatting in Chu '366 and Chen are completely irrelevant in Chu '684's system, and indeed, Chu '684 has no other need for reformatting. Chu '366 addresses the problem of how a traveler in an unknown location can dial according to the conventions of that geographical jurisdiction. See *supra* II(C)(2)(b)-(c). Chen addresses the problem of how the traveler in the unknown location can dial based on a fixed dial plan even if the Telco switch doesn't understand it. *Id.* There is no need in Chu '684 to reformat numbers for the reasons used by Chu '366 or Chen. Nor is there any need for Chu '684 to have reformatting inserted for any other reason. **Ex. 2016** at ¶ 64. As such, there is no evidentiary basis for Petitioner to assert there was a motivation to combine the references in such a manner as to meet the claim limitations. *Id.*

III. CONCLUSION

Because Petitioner fails to establish that Claims 1, 24-26, 49-50, 73-79, 83-84, 88-89, 92, 94-96, 98, and 99 of the '005 Patent are obvious over any available prior art, the Board should find the challenged claims not unpatentable.

Respectfully submitted,

KNOBBE, MARTENS, OLSON & BEAR, LLP

Dated: February 10, 2017

By: /Kerry Taylor/
Kerry Taylor, Reg. No. 43,947
John M. Carson, Reg. No. 34,303
Customer No. 20,995
Attorneys for Patent Owner
Voip-Pal.com, Inc.
(858) 707-4000

CERTIFICATE OF COMPLIANCE

This document complies with the type-volume limitation of 37 C.F.R. § 42.24(b)(2)(i). This Patent Owner Response contains 13,996 words, excluding the parts of the document exempted by 37 C.F.R. § 42.24(a)(1).

Respectfully submitted,

KNOBBE, MARTENS, OLSON & BEAR, LLP

Dated: February 10, 2017

By: /Kerry Taylor/
Kerry Taylor, Reg. No. 43,947
John M. Carson, Reg. No. 34,303
Customer No. 20,995
Attorneys for Patent Owner
Voip-Pal.com, Inc.
(858) 707-4000

CERTIFICATE OF SERVICE

I hereby certify that true and correct copy of **PATENT OWNER'S RESPONSE TO PETITION** is being served on February 10, 2017, via electronic mail pursuant to 37 C.F.R. § 42.6(e) as addressed below:

Adam P. Seitz
Eric A. Buresh
ERISE IPA, P.A.
6201 College Blvd., Suite 300
Overland Park, KS 66211
Telephone: (913) 777-5600
Adam.seitz@eriseip.com
eric.buresh@eriseip.com

Paul R. Hart
ERISE IPA, P.A.
5600 Greenwood Plaza Blvd., Suite 200
Greenwood Village, CO 80111
Telephone: (913) 777-5600
Paul.Hart@EriseIP.com

Dated: February 10, 2017

/Kerry Taylor/
Kerry Taylor, Reg. No. 43,947
John M. Carson, Reg. No. 34,303
Attorneys for Patent Owner
Voip-Pal.com, Inc.

25193422