INTERNET-DRAFT

PINT Working Group   L. Conroy
Internet Draft   M. Lautenbacher,
Expires on 21 May 1998   Roke Manor Research
                                                    Siemens AG

Pre-PINT Callback Service Implementation Experiences

<draft-ietf-pint-internet-callcenter-00.txt>

Status of this Memo

Abstract

This draft describes our experiences in developing an Intelligent Network Service Node that provides Call Center functions, using the Internet for (non-call) communications with customers and call center agents. It shows one approach to the provision of an interface between the Internet and the Public Switched Telephone Network (PSTN), via the Intelligent Network (IN).

1. Introduction

1.1. Context

This draft shows our experiences in developing an Intelligent Network Service Node that accepts requests from Internet Nodes for services to be provided on the PSTN. It builds on descriptions of the IN and PSTN in [1] and [2]. The development described preceded the PINT group, and highlights some of the factors that need to be considered in the development of PINT protocols.

1.2. About this document

First, Call Center features are introduced.

After this, the Intelligent Network Services that were developed to provide some of the features of a "wide area" Call Center operating purely on the PSTN are introduced. These are described in terms of user's interaction with the service. In the case of the first service feature, there is also a description of the activities carried out by the Intelligent Network components in providing the service on the PSTN.

The next section describes the way in which these services are provided using the Internet and the World Wide Web as an interface for all but the non-telephony related transactions. It contrasts these with the interactions involved with the "pure PSTN" Call Center configuration.

The security aspects of the use of the Internet in providing the Call Center are described next.

Finally, the lessons learnt from developing the "Internet" Call Center are described. These lessons point to a set of requirements on the interface between the Internet and the IN.

2. Call Center features

A Call Center is a system that allows a company to be organized with a group of similar individuals (agents), all of whom can either make calls to or take calls from customers. The system distributes incoming calls to the agents based on their availability and automates the placement of outgoing calls, selecting an agent to handle the call and routing the call to them only once the call request has been made of the PSTN.

The incoming call distribution feature ("automatic call distribution", or ACD) is usually coupled with a call queueing scheme. In this scheme, the callers are connected temporarily with an announcement unit that normally plays music. The calls are treated in sequence so that (once the caller is at the front of the queue) the ACD system selects the next available agent and routes the call through to them.

Another feature connects a customer making an incoming call to a unit that asks them for some information on the purpose of their call, selecting the agent to handle the call based on the particular area of expertise needed; to do this, the agents are further categorised by their knowledge (or "Skill Set"). If this skill set categorisation is used then by implication there will be separate queues for each of the skill sets. This user selection scheme can be used independently of the others. For example these so-called "voice navigation systems" can be used to select a particular department extension number, based on the function required by the customer; as such, they can automate the job of company telephone receptionist in routing incoming calls.

Where possible, the information gleaned from the customer can be provided to the selected agent, usually via a separate networked computer connection. Similarly, if an outgoing call is being made to one of a list of customers, information on the customer and the purpose of the call can be provided to the agent selected to handle the call. Such configurations are generally called "Computer Telephony Integration" or CTI systems. Strictly, a CTI system can be arranged to handle routing of incoming calls and automation of outgoing calls only (also known as computer integrated telephony features), without the agents having access to a network of computers. However, the business case for combining the telephony functions of the call center with provision to the agents of computers with customer information can be compelling.

This is often further combined with a company's order and service processing computer system. In this case, a call is treated as part of a business transaction, with the information to be exchanged captured as fields of a computer form. Whilst such a computer system is not, strictly, part of a call center, integrating the company computer system with the call center is very common. This allows the details of the call to be stored on a centralised database, allowing further automated order processing, for example. It also allows the call to be transferred from one agent to another where needed, ensuring that the new agent has the information already captured. This might be useful if someone with a different area of expertise were to be needed to handle the customer's requirements.

Traditionally, Call Centers have been used to support teams of agents working at a single site (or a small number of sites, with private telephony trunks interconnecting them). The site Private Automatic branch eXchange (PABX) was been integrated with a computer system to provide these features to people at that site. There can a business case for provision of such features to distributed teams of workers as well. In particular, the possibility of providing support for people working from home has been seen as important. Some of the Call Center features have been incorporated into public telephone exchanges or Central Offices (COs) from many manufacturers as part of their "Centrex" service offerings.

There are practical limitations in providing such features on COs. Apart from the procedures needed to configure these features for any telephone line that is to use them, the basic requirement that every agent must have a connection to the supporting CO can limit its usefulness. Another approach is to provide Call Center features via the Intelligent Network. The features might thus be provided over a Telephone Operator's entire network, and would mean that the Call Center could be configured centrally whilst still allowing agents to be located anywhere within the telephone network. It also means that the supported company can pay for the Call Center features "as they go" rather than having a high "up front" cost.

>From an Intelligent Network perspective, there are a number of services that, when combined, provide the Call Center features. Next the IN services needed to support a subset of these features are described. In particular, this subset supports the scenario in which a customer makes a request to be called back by an agent at a time of the customer's choosing to discuss an item of interest to them. The agent will be selected based on their availability and their expertise in this topic, they will be told who they are calling and the topic of interest, and then they will be connected to the customer.

3. IN Services to provide Call Center features

3.1. Call Center Services

The IN services support:

i)   Customer Request service - the customer calls a company service number, is connected to an announcement unit that gives them a list of available topics of interest, captures their selection from this list, and then prompts them to select the time at which they wish to be called back. Once this has been done the unit confirms the selection and time that the customer has chosen, and the service ends.

(Service Perspective) - the customer calls a company contact telephone number. This triggers control of the call to be passed from the Service Switching Point (SSP) to a program running the service logic on the Service Control Point (SCP). The message indicating this includes the telephone number from which the call has been made (the Calling Line Identity, or CLI) as well as the contact number they dialled (the Dialled Number, or DN). This program allocates and instructs an announcement unit in an Intelligent Peripheral, and asks the SSP to connect the call to this, temporarily.

The announcement unit plays the caller a list of available topics of interest, captures their selection from this list, and then prompts them to select the time at which they wish to be called back. Once this has been done the unit confirms the selection and time that the customer has chosen. It then contacts the service program, giving this the information it has gleaned, and breaks its connection to the caller.

The service program then releases control of the call to the SSP (with the final instruction to release the call), creates a new service data record for the customer on the Service Data Point (SDP) with the information it has been given by the SSP and the Intelligent Peripheral, informs the Call Center Agent Selection feature of this, and the service ends.

Note: The units described here (SSP, SCP, SDP, and Intelligent Peripheral) may well be combined into a single unit - the Service Node (SN). This fulfils all of the functions (Service Switching Function or SSF, Service Control Function or SCF, Service Data Function or SDF, and Special Resource Function or SRF). The effect is similar, but these functional entities are all co-located, can use proprietary protocols for internal communication, and are connected to the rest of the PSTN via an E1 access line.

ii) Agent Registration/Logon - An agent calls into a number. The service checks whether it has a record of an agent present at the telephone number from which they are calling. If not then the caller will be asked to enter their service identifier number. This identifier will be checked against a list of known identities and, if found, the caller will be asked to enter the company's agent identifier and then their PIN. If these match the record held by the service then a new session record is made of this identity and the telephone number from which they are calling.

NB:: This is very similar to the Universal Personal Telecommunications (UPT) service feature "register for incoming calls". It implies that the identified person has exclusive use of the telephone from that point onwards, so calls for them can be redirected to that number.

iii) Agent Ready - an agent who has already logged on can indicate that they are ready by dialling into the appropriate service number. The service will match the agent by the calling telephone number they are using and then mark them as being ready to handle calls in its list of available agents (with their pre-defined skill set).

iv) Agent Not Ready - an agent can dial into a service number to indicate that they are temporarily not ready to handle calls.

v) Agent Logoff - an agent can call into a service number to indicate that they are no longer associated with a particular telephone number; the telephone number is matched against a list of agents and, once found, the session record for that agent is removed and the caller is notified of this.

NB:: This is very similar to the UPT "unregister" service feature.

vi) Call Center Agent Selection and Notification - When the time that the customer selected has arrived and an available agent with the right skills has been selected from the appropriate list, this service will make a call out to that agent. Once they have picked up the call, they will be told that they have a customer to call.

NB:: This is similar to a "Message Waiting" or "Wake Up Call" service.

vii) Agent Instruction & callback - a selected agent calls into a service number. The calling telephone number they use will be matched against a list of agents expected to handle calls, and the instructions for their call will be spoken to them. They are given a list of actions that can be performed (of which the important ones are replay the message and place the call). Their choice from this list is noted, and, if it is to place the call, the service will make the call through to the customer and then connect the agent to this call. When the call completes, the instruction message is removed and the service ends.

NB:: This is similar to a "Voice Mail Replay" message service, but in this case the message is automatically generated; there is no associated voice mail record feature accessible.

The reason for the separation of service features (vi) and (vii) is that, with some systems, the message indication can be provided by a distinctive ringing of the agent's phone rather than making a call and then playing an announcement to them. The "main" part of the voice mail service is the same in either case.

3.2. IN Service Inter-component message flows

For each of the IN services, a textual description of the message contents is given, followed by a diagrammatic view of the message flows. Text items in {} are comments.

Note that the message flows include only the "successful" paths, and are simplifications. For example, most of the agent services will report a failure if there is no existing record with a matching CLI.

Note also that the use of the Initiate Call Attempt and Connect to SRF messages shown in the Agent Notification service is proprietary - it is not in the CS1 Intelligent Network standards. However, this can be achieved with equivalent CS2 messages.

```
Customer Request:
SSF->SCF: Trigger on DN. Initial DP(CLI,DP)
SCF->SRF: RequestReport Prompt and Collect
                            (Prompt = "Enter Time",
                            Collect 4Digits,
                            Hold Connection)
SCF->SSF: Make Temporary Connection to SRF
SRF->SCF: Report 4Digits
SCF->SDF: Create New Record with CLI and 4Digits
          Put it into Customer Request Table, sub-table based on DN
SCF->SRF: Play Announcement and Clear = "You will be called back at ",
                                        <4Digits>
SCF->SSF: Continue Execution, call state = Active
```

```
SSF               SCF                 SRF       SDF
o----InitialDP---->|                   :         :
:                  o-RqstReportP&C-->|          :
|<--TempConnection-o                   :         :
:                  :                   :         :
:                  |<-Report 4Digits-o          :
```

```
:                  o-Create New Record-------->|
:                  o-PlayAnnounce.-->|         :
|<-Cont.Execution--o                 :         :
```

Agent Registration/Logon:

```
SSF->SCF: Trigger on DN. Initial DP(CLI,DN)
SCF->SRF: RequestReport Prompt and Collect
                            (Prompt = "Enter Agent Ident",
                            Collect NDigits,
                            Hold Connection)
SCF->SSF: Make Temporary Connection to SRF
SRF->SCF: Report NDigits
SCF->SRF: RequestReport Prompt and Collect
                            (Prompt = "Enter PIN",
                            Collect MDigits,
                            Hold Connection)
SRF->SCF: Report MDigits
```

```
SCF->SDF: Lookup record in Agents table, sub-table based on DN,
          key AgentID=NDigits
```

SDF->SCF: Requested record value

{SCF Compares record.PIN with MDigits. On success...}

SCF->SDF: Modify Record (record.location = CLI)

SCF->SRF: Play Announcement and Clear = "Agent registered"

SCF->SSF: Continue Execution, call state = Active

```
SSF               SCF               SRF       SDF
o----InitialDP---->|                 :         :
:                  o-RqstReportP&C-->|         :
|<--TempConnection-o                 :         :
:                  :                 :         :
:                  |<-Report NDigits-o         :
:                  :                 :         :
:                  o-RqstReportP&C-->|         :
:                  :                 :         :
:                  |<-Report MDigits-o         :
:                  o-Query Record------------->|
:                  <-----Agent Record Value----o
:                  :                 :         :
:                  o-Modify Existing Record--->|
:                  o-PlayAnnounce -->|         :
|<-Cont.Execution--o                 :         :
```

Agent Ready:

SSF->SCF: Trigger on DN. Initial DP(CLI,DN)

SCF->SSF: Make Temporary Connection to SRF

SCF->SDF: Lookup record in Agents table, sub-table based on DN, key location=CLI

SDF->SCF: Requested record value

SCF->SDF: Modify Record (record.status = free)

SCF->SRF: Play Announcement and Clear = "You will be called when needed"

SCF->SSF: Continue Execution, call state = Active

```
SSF                SCF                 SRF       SDF
o----InitialDP---->|                    :         :
|<--TempConnection-o                    :         :
:                   o-Query Record------------->|
:                   <--------Record Value-------o
:                   o-Modify Existing Record--->|
:                  :                  :         :
:                   o-PlayAnnounce -->|         :
|<-Cont.Execution--o                  :         :
```

Agent Not Ready:

```
SSF->SCF: Trigger on DN. Initial DP(CLI,DN)

SCF->SSF: Make Temporary Connection to SRF.

SCF->SDF: Lookup record in Agents table, sub-table based on DN,

          key location=CLI

SDF->SCF: Requested record value

SCF->SDF: Modify Record (record.status = not-free)

SCF->SRF: Play Announcement and Clear = "You are on break"

SCF->SSF: Continue Execution, call state = Active


SSF                SCF                SRF       SDF

o----InitialDP---->|                   :         :

|<--TempConnection-o                   :         :

:                  o-Query Record------------->|

:                  <--------Record Value-------o

:                  o-Modify Existing Record--->|

:                  :                 :         :

:                  o-PlayAnnounce -->|         :

|<-Cont.Execution--o                   :         :



Agent Logoff:

SSF->SCF: Trigger on DN. Initial DP(CLI,DN)

SCF->SSF: Make Temporary Connection to SRF

SCF->SDF: Lookup record in Agents table, sub-table based on DN,

          key location=CLI

SDF->SCF: Requested record value
```

```
SCF->SDF: Modify Record (record.location = NULL)
SCF->SRF: Play Announcement and Clear = "You are no longer registered
                                         at your current location"
SCF->SSF: Continue Execution, call state = Active
```

```
SSF               SCF               SRF       SDF
o----InitialDP---->|                 :         :
|<--TempConnection-o                 :         :
:                  o-Query Record------------->|
:                  <--------Record Value-------o
:                  o-Modify Existing Record--->|
:                  :                 :         :
:                  o-PlayAnnounce -->|         :
|<-Cont.Execution--o                 :         :
```

Call Center Agent Selection and Notification:

```
{On Timer Expiry of Customer Record...}
SCF->SDF: Lookup record in Agents table, sub-table based on DN,
          key status=free, return first record only
SDF->SCF: Requested record value
```

SCF->SDF: Modify Record (record.nextCustomer = Customer Record,
record.status = allocated)

SCF->SSF: Initiate Call Attempt to record.location, Connect to SRF

SCF->SRF: Play Announcement and Clear = "You have a customer waiting"

SCF->SSF: Continue Execution, call state = Active

```
SSF               SCF               SRF       SDF
:      !Customer Record Expiry!      :         :
:                  o-Query Record------------->|
:                  <--------Record Value-------o
:                  o-Modify Existing Record--->|
:                  :                 :         :
|<-InitCallAttempt-o                 :         :
|<--Connect to SRF-o                 :         :
:                  o-PlayAnnounce -->|         :
|<-Cont.Execution--o                 :         :
```

Agent Instruction & customer callback

SSF->SCF: Trigger on DN. Initial DP(CLI,DN)

SCF->SDF: Lookup record in Agents table, sub-table based on DN,
key location=CLI {and status=allocated}

SDF->SCF: Requested record value

SCF->SDF: Modify Record(record.status=busy)

SCF->SDF: Lookup custRec in Customer Request Table, sub-table based

```
          on DN, key=record.nextCustomer
SDF->SCF: Requested custRec value
SCF->SRF: RequestReport Prompt and Collect
                            (Prompt = "Press 1 for customer details,
                                      Press 2 to call them",
                            Collect 1Digit,
                            Clear Connection)
SCF->SSF: Make Temporary Connection to SRF
SRF->SCF: Report 1Digit
{If 1Digit=1 then the SRF is requested to speak an announcement of the
 information in custRec, and then to repeat the prompt and collect
 above}
{On 1Digit=2...}
SCF->SSF: Continue Execution, call state=Analysed Info,
          data=custRec.CLI {ie. complete the call to the customer}
```

```
SSF               SCF                SRF       SDF
o----InitialDP---->|                  :         :
:                  o-Query Record------------->|
:                  <-----Agent Record Value----o
:                  o-Modify Existing Record--->|
:                  o-Query Record------------->|
:                  <-Customer Record Value-----o
:                  :                 :         :
:                  o-RqstReportP&C-->|         :       <..........
|<--TempConnection-o                 :         :                 .
:                  :                 :         :                 .
:                  |<-Report 1Digit--o         :                 .
:                  :                 :         :                 .
{if 1Digit=1, generate announcement of customer information.     .
 Play announcement of customer information, then..................
 else if 1Digit=2...}
:                  :                 :         :
|<-Cont.Execution--o                 :         :
```

4. Internet Call Center

4.1. Internet Call Center Services

The scenario supported by the Internet Call Center is virtually identical to the PINT "core" service "Internet Requested Telephony Dialback". It is also very similar to the IN-based Call Center just described. As provided via the Internet, the services involved are mostly the same as those provided via the PSTN and IN alone. The main differences lie in the use of the World Wide Web as an interface to the services rather than a telephone, SSP, and Intelligent Peripheral. Also, the feature by which a telephone call is made between the agent and the customer is split into a separate service; this is the only element in which the PSTN is involved. Within the Internet Call Center, the services provided are:

i) Customer Request service

ii) Agent Registration/Logon

iii) Agent Ready

iv) Agent Not Ready

v) Agent Logoff

vi) Call Center Agent Selection and Notification (Wake up call)

vii) Agent (web-based) instruction delivery

viii) Agent/Customer Telephony Callback

As implemented, the Internet Call Center provides two variants of the Agent Logon service. One of these provides "full" registration in a similar way to the IN-based UPT service. The other "quick" logon is used where an agent has been "bound" to their machine previously, and is merely re-registering at the start of a work session.

4.2. User Interaction

In the IN/PSTN-based system, the services have contact with the customers and agents via their telephones, SSPs, and Intelligent Peripherals programmed to play announcements to them and to capture their responses. These responses are indicated by DTMF tones sent by pressing keys on the telephones.

In this case, almost all interactions are provided via World Wide Web requests and responses. The sequence of announcements and responses for each service are "collapsed" into individual form filling transactions, and the requests are not limited to digits (or "star" and "hash"). The implications of the use of forms on service operation are covered in more detail later (under HTTP/IN Service mapping).

4.3. Service / Caller Identifiers

When provided via the IN/PSTN-based system, the services are passed the CLI of the caller and the number they dial (the DN). The CLI value is used extensively to identify the caller and (in the case of the agent) to index into service data tables to decide what to do next. Whilst an equivalent value to the DN is passed to the Web-based transactions as the requested URL, the CLI cannot be given reliably. The nearest equivalent caller identifier is the IP Address of the customer or agent's machine. However, the use of HTTP proxies means that this "original" Internet Node Address may not be available; if a proxy is used then its IP Address will be associated with the request.

In providing these Call Center features the customer only has one Web-based transaction; that of providing the initial request for a PSTN telephone callback. To do so they will have to fill in a form, as they need to specify not only the time that they want to be called back, but also the telephone number to which they want the call to be made. These values can be used if needed to identify the customer, and so the problem of originating Internet Node ambiguity is not relevant.

With the agents, however, there are sequences of coupled transactions, and the particular sequence must be identified. There will be a number of such transactions being carried out at once, and there needs to be some identifier to show which agent is being handled in each case.

Such an identifier is not part of a sequence of basic Web transactions. In a Web transaction, the HTTP client/web browser makes a request, and the HTTP server will respond to this, normally including some content in its reply message that will be processed by the browser, after which it closes the TCP connection. That's the end of the transaction; the HTTP client and server cannot normally maintain state information beyond this point. Any sequence is reduced to a set of unrelated transactions.

A result of this simple pattern is that any state information reflecting longer or more complex interactions must be stored (at least partially) in the client system. One approach is the use of cookies[3]. These can be set by HTTP servers as part of their response to a request, and will be sent back with all subsequent requests for appropriate URLs as extra HTTP headers. These cookies allow the HTTP server to identify the client in the following requests, so that it can continue an extended session with the client.

Cookies are used in providing the Internet Call Center. Persistent cookies are installed into the Web browser on machines that are to be used by call center agents as a service management (pre-service) task.

The cookie value is unique to the machine and is used to index into a list of machine IP Addresses that is stored as part of the service data.

Also, a session cookie is stored onto the agent's machine when they register, and is cleared when they de-register. This is used to identify the agent and so the IP Address of the node with which they are associated (and so from which their subsequent requests should originate). The services that interact with call center agents use the agent session cookie value as an identifier; in principle this is unnecessary but it does simplify the session data lookup procedure. The rest of the services use the persistent machine identifier in place of the CLI, indexing into their service data using it. Both cookies are sent with each agent request; if they are not present, then the request is redirected to other services (for example to the agent Logon service).

4.4. Mapping from HTTP transactions to IN based service features

All of the client-initiated services require user interaction. With the IN/PSTN-based system, the majority of the services are typified by the caller being connected to an announcement unit that plays them a list of choices and captures their selection. The caller can pre-dial the digits needed; in this case the prompts are not needed and are not made.

The pattern of operation is somewhat different in the Internet case, as the initial HTTP request returns a response, after which the Web transaction has ended. Where that initial response returns a form to be filled in by the caller, subsequently submitting of the form initiates a new HTTP transaction. This is all part of one instance of service, however. The service consists of two request/response pairs in tandem.

Although it is possible to design a service to handle this pair of web transactions as a single unit, it may be better to reconfigure it. The design of a service that deals with two web exchanges as a single extended transaction is quite complex. It must maintain state across the pair of web exchanges, and it has to handle a number of failure cases including dealing with timeouts and "out of time" submission of forms. The alternative is to split the service into two sub-features. The first of these reflects the initial request and delivery of the form by return, with the second one dealing with processing of the submitted form and returning any confirmation by reply.

The services offered don't all require form-filling, and so can be treated as a single IN feature. There are two cases where forms are required. The first of these is the Customer Request service, whilst the other one is the "Agent Registration" service. In both cases the initial web transaction (by which the form is requested and returned to the client) need not involve specific service logic processing; the initial delivery of the form to a customer or agent can be handled by a "normal" web server. In both these cases the service logic is only triggered when the form is submitted; this means that, again, each of the services can be treated as a single IN feature.

The IN service logic that deals with these requests has a general pattern of action. An HTTP request is received, and this triggers the IN service logic into action. The service logic "sees" this as an InitialDP message and starts its processing as if it had been sent from an SSF. The SCF uses an "Internet Intelligent Peripheral" to collect the parameters of the request, and then to send back final announcements to the requesting entity.

The main difference, from the perspective of the IN service logic running on the SCF, is that the service does not need to instruct the SSF to make a temporary connection to the Intelligent Peripheral. It is as if this connection had already been made. Similarly, there is no need to close the service transaction by sending an explicit "Continue Execution" message to the SSF.

The sequence of "prompt/collect" instructions used to collect service parameters from a caller in an IN service maps quite well to a sequence of requests to extract a data value from the HTTP request, based on a tag. This is a fairly standard feature of Web Server CGI or Servlet processing. Using this mapping minimises the changes to the service design, in that the service logic "sees" an Intelligent Peripheral to which it sends normal "RequestReportPrompt & Collect" messages, and from which it receives data values in response.

All services have to fit in with the underlying HTTP interaction pattern, and so will be expected to send a final "Announce" instruction to the Intelligent Peripheral at the end of the service; this is done in many IN services anyway and in all of the service features described here. These announcements form the content returned to the web client.

4.5. Non-World Wide Web Interactions

There are two exceptions to the sole use of the World Wide Web for interaction. The first one occurs in the "Message Waiting"/"Wake Up Call" service by which the selected agent is informed that they have a callback to handle. World Wide Web transactions are very simple; the client browser makes a request for content associated with a particular HTTP URL, and the server sends a response, marking the end of the transaction. The server cannot make a spontaneous association with a client; it must be initiated by the client request.

Whilst it would be possible for the server to defer closing an earlier transaction (by not sending back all of the content specified and leaving the TCP connection open) it was decided that an alternative scheme would be more convenient. The "wake up call" was arranged by an "Internet Intelligent Peripheral" sending a request to a daemon process running on the selected agent's machine, using the Finger protocol[4]. The daemon sent back a standard response, but in addition the Web browser on the agent's machine was triggered into making a further HTTP request of the server. In this way the "Agent Instruction" transaction is started automatically, whilst still allowing it to use a normal HTTP request/response pattern.

The second exception occurs in the final "Agent/Customer Telephony Callback" service. Whilst this transaction is initiated by the agent selecting a link on the "call instructions page" returned to them, and includes a "confirmation" page being sent back to them in an HTTP response, the purpose of this service is to make a telephone connection via the PSTN between the agent's telephone and the customer's telephone. It is the only service element that involves the PSTN directly. From an IN/PSTN perspective, the resulting telephone connection is different from that provided in the scheme using the IN and PSTN alone. In this case, a PSTN call is made out to the agent's telephone, another call is made out to the customer's telephone, and these calls are bridged. This differs from the earlier scheme, where the agent has originated a call to the voice mail replay system, and this call is redirected to a new destination (the customer's telephone). As this feature differs in purpose from the other services, and it requires a different implementation within the IN and PSTN system, it was organised as a separate service in this case.

5. Security Implications

In the case of this system, assumptions were made that the interface presented to requesting agents and customers was provided via a fire wall to deal with most attacks on the Service Node. The interface appeared as a Web Server, and there was no direct access to the HTTP documents served, nor to the servlets providing the service logic.

The Callback service was deemed to have simpler security requirements than other IN services as it was akin to a free phone "1-800" service access number; the agents work for the service subscriber and are not charged directly. Similarly, the requesting customer is not charged for their request, nor for the resulting call back. Service subscribers would be wiling to pay the costs of telephone calls generated as a result of this cluster of services, and the costs of running the agent services could be charged directly to them. As such the authorisation for service is defined by the contract between the service subscriber and the service provider.

Authentication of agents was seen as a problem. As an interim measure, cookies were used, but this scheme delivers the cookie data as a plain text item (a header of the web request). Secure Socket Layer connections were required for communication with the agent services, and this had an impact on the performance of the Service Node.

6. Lessons

Security is seen as a major issue. As already mentioned, we used a firewall to control access to the Service Node. Similarly, we used SSL for communication with the Agents, so as to protect the cookie values that they were sending with their requests.

For other PINT core services, it is likely that the requesting entity will be charged for the service to be rendered. This has implications in terms of authentication and authorisation of service provision at the time of the request. It is necessary for the service to be authorised in such a way that non-repudiation is ensured; this is likely to mean that a certificate of identity be provided from the person making the request, and that this can be tied in with a financial account that that person has with the service provider. The certificate can then be stored as part of the billing record. Whilst the process of electronic commerce is outside of the scope of PINT, the mechanism by which a request for confirmation of identity is passed out to the requesting user and is delivered back to the service logic must be considered.

When changing from a "pure " IN/PSTN system to one supporting requests via the Internet, the differences in the way that clients interacted with the services meant that the service logic had to be redesigned. We realised that maintaining the state of a service during its processing was going to be a problem; we sidestepped this problem by re-engineering the services as form processors, allowing them to deal with fully specified requests as a single (web) transaction. In addition, we used a "normal" web server to deliver the forms to the users. This is a change from the IN system, where the equivalent of the form (the prompts) were sent in sequence as part of the same service process.

The Call Center features provided suited this change. However, this may not be the case for other IN services. It is quite common for services to be designed such that the user is prompted for a response, and the service continues dependent on this response. The web form presents all of the options at once, so this kind of variant prompt/collect sequence is not possible. From this, it is difficult to see how an IN service could be reused without some degree of modification.

We provided an intermediate "gateway" system to "cocoon" the service logic as far as possible from the details of the components with which it was working. Where needed, this unit translated calls from the service logic into commands that operated with the Internet (and the Web Server that acted as the interface).

In our case, the "Internet Intelligent Peripheral" with which the service logic communicated was running as a separate program on the same node. Where more complex behaviour is required of it (such as conversion of text to speech data and interface with the PSTN) then it would almost certainly be on a separate node. If data is transferred from the Internet in such a scheme, any intermediate gateway would be involved in relaying the data to this node.

7. Author Address

Lawrence Conroy
E-mail: lwc@roke.co.uk
Telephone: +44-1794-833666
Fax: +44-1794-833434

Roke Manor Research Limited
IT&N-INIA Group
Roke Manor, Old Salisbury Lane,
Romsey, Hampshire   SO51 0ZN
U.K.

Markus Lautenbacher
E-mail:markus.lautenbacher@oen.siemens.de
Telephone: +49-89-722-37805
Fax: +49-89-722-23528

Siemens OEN MD13
Machtlfinger Strasse 10

Munich   D-81359

Germany

8. References

[1] I. Faynberg, M. Krishnaswamy, and H. Lu.
INTERNET-DRAFT:draft-faynberg-telephone-sw-net-00.txt,
"A Proposal for Internet and Public Switched Telephone
Networks (PSTN) Internetworking". Issued March 1997

[2] L. Conroy, M. Lautenbacher, and R. Scholl.
INTERNET-DRAFT:draft-conroy-interfaces-in-net-00.txt,
"Analysis of Services and Interfaces used when Interworking
Between the Internet and the Intelligent Network (I.N.)".
Issued June 1997.

[3] D. Kristol and L. Montulli, RFC2109,
"HTTP State Management Mechanism"

[4] D. Zimmerman, RFC1288,
"The Finger User Information Protocol"