within the presence management system is provided to manage these profiles.

For example, an entity's profile contains the following items which may be entered by the entity or watched party or may be default information that is prespecified :-

- Contact addresses for that entity on each of a number of access communications networks or component parts of an access communications network.
- A set of the possible context presence values for the entity, e.g. "at-home", "at-work", "travelling", "not-able-to-contact" or "unknown". For many of these the system also holds attributes, e.g. geographical location.
- Minor changes to the default rules for moving from one context presence to another.
- The entity's own view of what potential audiences its exposed presence has, e.g. family, friends, colleagues. Members of each potential audience see a different projection of the entity's presence. In application to enterprises, the administrator is able to provide audience definitions common to many entities (cf. E-mail distribution lists).
- A watcher authentication level for each audience to allow the entity to have safeguards that a watcher really does belong to an audience.
- For some sorts of communication (e.g. instant messages informing of new E-mail) the entity may wish to exercise fine-grained control and insist on the watcher making a

micro-subscription. The attributes of such a subscription can also be subject to an entity's rules (cf. E-mail filters)

When an entity is first registered with a presence management system server 10 there is a substantial amount of data capture required. In order that the work-load of the entity is reduced reliance is put on default information and a type of "wizard" is used to help the user enter the required information quickly and easily. Later, the data can be subsequently modified by the watched party or entity. These functions are carried out by the profile management system 56.

For example, in the case of an entity representing a human user the interaction provided by the "wizard" data capture process can be as follows:

- The user chooses from a number of life-styles, "office-worker", "travelling-salesman" or "executive". Each life-style being represented by a template with default values.
- Each choice leads to a dialog tailored for that choice. For example, to establish the geographical location of the home and of the office. Additional locations are created if appropriate.
- In one embodiment, the presence management system is able to detect and capture details about the devices that the entity uses for communicating.
- Definitions of potential audiences for the user's presence (family, friends etc.)

- Finally, the user will decide which contact information is notified to members of which audiences for what changes in context presence.

Some parts of the configuration are complicated and these are initially masked from the user until they press the "advanced options" button.

The presence management system also contains a raw presence manager 55 which manages an unprocessed collection of presence indicators. The raw presence manager 55 orders the presence indicators into a table or other structure which is indexed by entity. In this way each presence indicated is accessible on the basis of the entity to which it relates. Large volumes of presence indicators can be stored by the raw presence manager 55. However, by their very nature, presence indicators become redundant once more recent presence indicators about a given entity become available and a process for updating presence indicators or giving weights to these indicators on the basis of age is provided.

The multiple access communications network 11 is made up of one or more communications networks, such as a public switched telephone network 52 and a wireless communications network 51. An event gateway, specific to the type of communications network, is provided for each communications network or each technology, such as mobile telephones and copper wire telephones. Note that the Internet is logically a federation of several access networks and a core data network using a common Internet Protocol (IP) substrate.

Event gateways receive raw data from the access networks, pre-process this and pass it onto the raw presence manager via one or more event inputs 16. Examples of these events are keyboard activity on a terminal, movement between cells in a cellular communications network and change of VLR (Visitor Location Register) and off-hook events in a PSTN. Other events include use of the "last offered incoming call" service on a public switched telephone network, details from calendar services and the opening of a Palm Pilot 7 (trade mark) lid. Event gateways produce an authenticated, time ordered series of location data events that have been filtered and homogenised. That is, only information about users being tracked by this presence system is passed to the raw presence manager. This is achieved by using a filter in the event gateway or in the events input 16. Also, events are presented to the raw presence manager in a technology neutral format (all the idiosyncrasies of different access network technology are hidden by the event gateway as far as possible).

An event gateway is divided into two parts. A first part interfaces with the presence management system 10 whilst a second part interfaces with the access network. The two parts of the event gateway communicate using a protocol over an interface. Different types of event gateway are provided according to the type of access network used.

As described above a subscription is made when a watcher registers an interest in the presence of an entity. A

subscription manager 57 is provided within the presence management system 10 in order to manage the subscription process.

Users or services (watching parties) outside of the presence management system may subscribe to presence information about particular clients (watched parties) of the presence system. If the presence management system comprises multiple servers this request is routed to the home server for the client (watched party) in question. The subscription request is made over one of a number of protocols: IMPP, WAP or possibly ICQ. In each case the subscription request contains a return address for notifications (e.g. an IMPP address), some identification information for the subscriber or watching party (so that the presence client's propagation restrictions can be obeyed) and the identity of the presence client about whom presence information is sought.

In one example, the lifetime of a subscription is controlled by the subscriber, i.e. the subscriber (watching party) declares when its interest in a given presence client is over. However, if the watching party does not take this action to declare the end of its interest then old subscriptions eventually become garbage. In one case, subscriptions are held persistently (for fault recovery) and this creates an issue of garbage collection. For example, all subscriptions that are inactive for one hour may be deleted.

In the case of Instant Message notification of an awaiting e-mail, for example, the decision on whether the client is "on-line" to a particular e-mail notification may be a function of the sender, the size and possibly of keywords within the subject of the e-mail. Rather than the subscriber presenting this, second order, information at the point of sending the Instant Message, the initial subscription may capture all the information upon which the reachability decision is made. What might have been a single subscription by an e-mail notification service regarding a single client may then be replaced by a set of fine-grained subscriptions each regarding that client's willingness to receive a certain e-mail notification.

In this case, IMPP is extended to carry additional parameters. This may be in the form of an XML string. For example, in the case of an e-mail notification service, the IMPP subscribe also contains <subscription-detail type="IM" reftype="e-mail" from="fred">. One possibility is that the presence system responds to such a subscription by asking for further information. e.g., if the rules asserted by the client cover the "to" field of the e-mail (i.e. who else has/will receive it) then the response is to refuse the subscription giving an appropriate reason.

The presence management system 10 also comprises an exposed presence manager 58 as illustrated in Figure 5. As described above "exposed presence" is a view of an entity's presence that is exposed to a watcher. It is derived

according to rules defined for that entity and in this way, an entity exhibits different exposed presences to different audiences. The function of the exposed presence manager 58 is to "decide" what information to make available to watching parties on the basis of rules 15 and other criteria set by watched parties or defined as defaults.

In the case that there is a change in the information available, for example, because new information arrives via an event gateway 53, then the exposed presence manager 58 is triggered. The exposed presence manager then checks for active subscriptions to the watched party in respect of whom the new information has arrived. If some active subscriptions are present, the exposed presence manager 58 consults the rules 15 and determines what (if any) information to make available to the watching parties who have active subscriptions.

The presence management system 10 may also have an interface 59 to ICQ and/or IMPP 60 or WAP 61 protocol outputs. Information that is made available to watching parties who have active subscriptions can be made via this interface 59 or these outputs.

The presence management system also comprises a connection manager 59 in some examples. In the cases that the presence management system itself forwards communication requests direct to watched parties, the connection manager 59 provides this forwarding facility using connect gateways 54. At least one connect gateway 54 is provided for each type of

access network. Part of the function of the connection manager 59 is to "decide" which connect gateway 54 should be used for a particular communication request and to manage changes between connect gateways 54 according to the communication requirements. As well as this the connection manager determines which access media is used for a given interaction in the event that more than one access medium is available and suitable.

In one embodiment the connection manager 59 comprises a mediation component 62. This enables messages in one format e.g. voice to be changed into another format e.g. email, in order to provide flexibility for watched parties and watching parties. The mediation component 62 uses rules to decide whether and how best to change the format of a given message.

The presence management system also comprises a raw presence manager 55. The raw presence manager 55 obtains and stores information from the event gateways 53 and processes this information to form a "context presence" for each watched party. A plurality of default contexts are defined such as "at home", "at work desk", "travelling" and information received from the event gateways 53 together with other watched party information is used to determine which context applies for a given watched party at a particular time. Rules are used to aid this determination process. For example, rules for mapping location indicators onto presence contexts are pre-defined. Watched parties are also able to enter their own rules and presence contexts for use by the

raw presence manager 55.

The rules used by the raw presence manager 55 and the exposed presence manager 58 are always locally resolvable. Many of the other components of the presence management system use rules. These may be stored in one rule base or alternatively stored in conjunction with the particular components of the presence management system. For example, the connection manager 59 uses rules to decide which of several possible connection options to prefer.

The interface between the raw presence manager 55 and the exposed presence manager 58 may be distributed. The two managers 55, 58 have different requirements. The raw presence manager 55 makes only local updates to a data store 14 based on its input data. It exhibits a low latency so that location data from event gateways 53 need not be backed up persistently. That is, once processed by the raw presence manager 55 any significant changes are held in the data store 14.

The exposed presence manager 58 has to engage in a number of external interactions, over a variety of protocols to propagate presence information to subscribers. It is either triggered directly from the data store 14, as a reflex to the updates made by the raw presence manager 55 or there is a direct interface between the two.

In one embodiment a computer program stored on a computer readable medium is provided. Figure 6 is a flow diagram of the method implemented by the computer program.

This computer program is adapted to control a presence management system such that connections are managed between watching parties and watched parties. This presence management system is suitable for use in a multiple access communications network, and said computer program is arranged to control said presence management system such that:-

(i) a store of watched party information is created (box 701 of Figure 6);

(ii) a set of rules about connection criteria are formed (box 702 of Figure 6);

(iii) information about events that occur in said multiple access communications network is received in use via an input (box 703 of Figure 6); and

(iv) connection requests are received from watchers in use; and on receipt of a request from a watching party for a connection with a watched party, information about whether that watched party currently permits connections to be established with it is provided, said information being determined on the basis of said store, said rules and said input information about events; and wherein at least one party is a service (box 704 of Figure 6).

Figure 7 is a flow diagram of another example of such a method implemented by a computer program. In this case, watched parties first register 801 with a presence management system and input information 802 which is used to form rules. This information is about the watched party's communication

criteria and preferences, such as what types of terminal he or she uses and which modes of communication are preferred at different times of the day.

The presence management system may then receive a number of different types of input. For example, a subscription 803 from a watching party may be received, in which case the subscription details are recorded 809. For example, this could be a request by a watching party to be informed about any change of state in a watched party.

The presence management system may also receive an incoming event 804. In this case, the method involves checking 805 all the current subscriptions to see if the incoming event is relevant to any of these. If so, notifications are sent out 806 to the watching parties who made the relevant subscriptions. The record of the subscriptions is then updated 805. For example, a subscription may be arranged to be activated only once and then deleted.

The presence management system may also receive an incoming connection request 807 from a watching party. In this case the presence management system checks the availability of the watched party and if appropriate establishes a connection or forwards the connection request to the watched party 808.

In one example, a presence management system comprises: a first input arranged to receive requests from watching parties in use, each of which may be of one of three types:

(a) a notification request being in respect of a watched party and, optionally, a specific event, (b) a contact request in respect of a watched party, or (c) a fetch request in respect of a watched party. The presence management system also has a second input arranged to receive information about events, relating to said watched parties, that occur in said multiple access communications network in use. As well as this a third input is provided, arranged to receive and store information entered by a watched party. This information relates to the said watched party and is used to transform the incoming events and control watching parties in respect of the information that they may receive about the watched party. The information is stored so that it can be associated with the watched party to which it relates. The presence management system has a processor arranged such that in use, when information about an event relating to a particular watched party is received, the information is transformed in accordance with the information previously received from the watched party. When information about an event relating to a particular watched party is received, any watching parties who made notification requests about that particular watched party are notified. This is done provided that the stored information does not forbid or restrict the transmission of a notification at that time. For example, the watching party may have specified that he or she should not be interrupted between 9 am and 10 am. If a contact request is received, contact is only established by

the presence management system if the required terminals are available and if the user preferences of both parties allow contact to be established. If a fetch request is received, part or all of the information relating to a watched party is transmitted to the watching party, according to user preferences and criteria set by the watched party. In this way a watching party can quickly and easily obtain all available information about a watched party.

A range of applications are within the scope of the present invention. These include any presence management systems in which a watched party is able to control who has access to his or her direct contact details and for how long; the watched party can also prevent others from gaining access to his or her direct contact details. The invention also encompasses a computer program for controlling such a presence management system and a multiple access communications network which comprises such a presence management system.

Claims

1. A presence management system suitable for use in a multiple access communications network by watching parties and watched parties, said presence management system comprising:-

(i) an input arranged to access a store of watched party information;

(ii) information about connection criteria;

(iii) an input arranged to receive information about events that occur in said multiple access communications network in use; and

(iv) an input arranged to receive connection requests from watchers in use; and wherein said presence management system is arranged such that, in use, on receipt of a request from a watching party for a connection with a watched party, a connection address for that watched party is provided under controlled conditions, said conditions being determined on the basis of said watched party information, said information about connection criteria and said input information about events.

2. A presence management system as claimed in claim 1 wherein said connection address is only operable for a limited time.

3. A presence management system as claimed in claim 1 or claim 2 wherein said connection address is only operable for a limited number of connection attempts.

4. A presence management system as claimed in any preceding claim wherein the request from the watching party is forwarded to the connection address provided, in such a way that the watching party has no access to that connection address.

5. A presence management system as claimed in any preceding claim wherein at least one of the parties is an automated service.

6. A presence management system as claimed in claim 5 wherein said automated service is a watching party.

7. A presence management system as claimed in claim 5 wherein said automated service is a watched party.

8. A presence management system as claimed in any preceding claim wherein at least one of said parties comprises a plurality of individuals.

9. A presence management system as claimed in any preceding claim wherein said presence management system is further arranged to provide information about the geographical location of a watched party in use, on the basis of said received information about events that occur in said multiple access communications network.

10. A presence management system as claimed in any preceding claim wherein said presence management system is further arranged to provide information about the current activity of a watched party, on the basis of said received information about events that occur in said multiple access communications network.

11. A presence management system as claimed in any preceding claim wherein said presence management system is arranged to provide information about types of connection that a watched party is able to participate in.

12. A presence management system as claimed in any preceding claim which further comprises a monitor for monitoring said received information about events and wherein said presence management system is arranged to provide information about a change in the availability of a watched party.

13. A presence management system as claimed in any preceding claim wherein said store of watched party information comprises watched party connection preferences.

14. A presence management system as claimed in any preceding claim and wherein a plurality of said events are initiated by watched parties and comprise a communication via said multiple access network.

15. A computer program stored on a computer readable medium, said computer program being adapted to control a presence management system, said presence management system being suitable for use in a multiple access communications network by watched parties and watching parties, said computer program being arranged to control said presence management system such that:-

(i) a store of watched party information is accessed;

(ii) information about connection criteria is accessed;

(iii) information about events that occur in said multiple access communications network is received in use via an input; and

(iv) connection requests are received from watchers in use; and on receipt of a request from a watching party for a connection with a watched party, a connection address for that watched party is provided under controlled conditions, said conditions being determined on the basis of said watched party information, said information about connection criteria and said input information about events.

16. A multiple access communications network comprising a presence management system, said presence management system being for use by watching parties and watched parties, said presence management system comprising:-

(i) An input arranged to access a store of watched party information;

(ii) information about connection criteria;

(iii) an input arranged to receive information about events that occur in said multiple access communications network in use; and

(iv) an input arranged to receive connection requests from watchers in use; and wherein said presence management system is arranged such that, in use, on receipt of a request from a watching party for a connection with a

watched party, a connection address for that watched party is provided under controlled conditions, said conditions being determined on the basis of said watched party information, said information about connection criteria and said input information about events.

17. A method of operating a presence management system suitable for use in a multiple access communications network, said presence management system being for use by watching parties and watched parties, at least one of said parties being an automated service, said method comprising the steps of:-

(i) accessing a store of watched party information;

(ii) accessing information about connection criteria;

(iii) receiving information about events that occur in said multiple access communications network; and

(iv) on receipt of a request from a watching party for a connection with a watched party, a connection address for that watched party is provided under controlled conditions, said conditions being determined on the basis of said watched party information, said information about connection criteria and said input information about events.

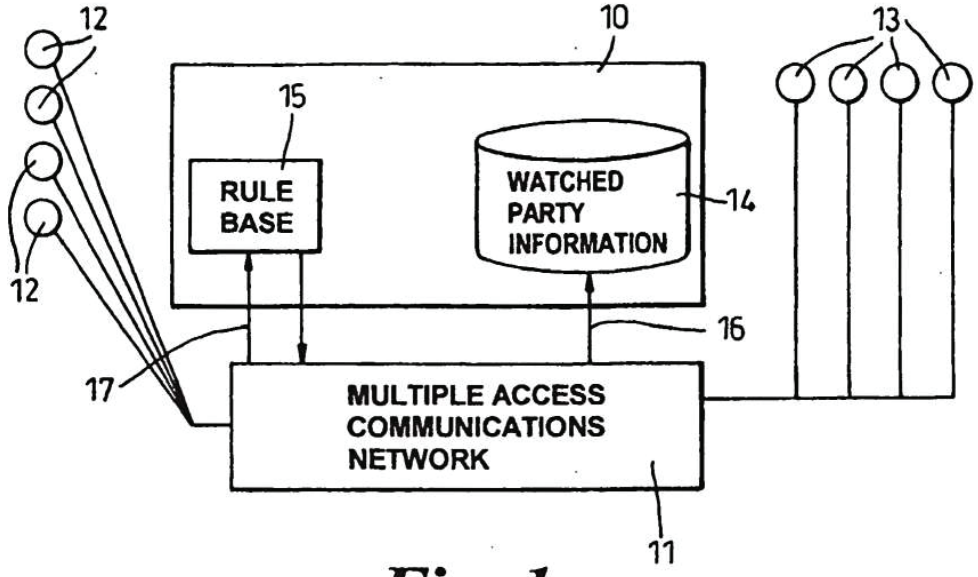

12
10
13
15
RULE BASE
WATCHED PARTY INFORMATION
14
12
16
17
MULTIPLE ACCESS COMMUNICATIONS NETWORK
11

Fig. 1

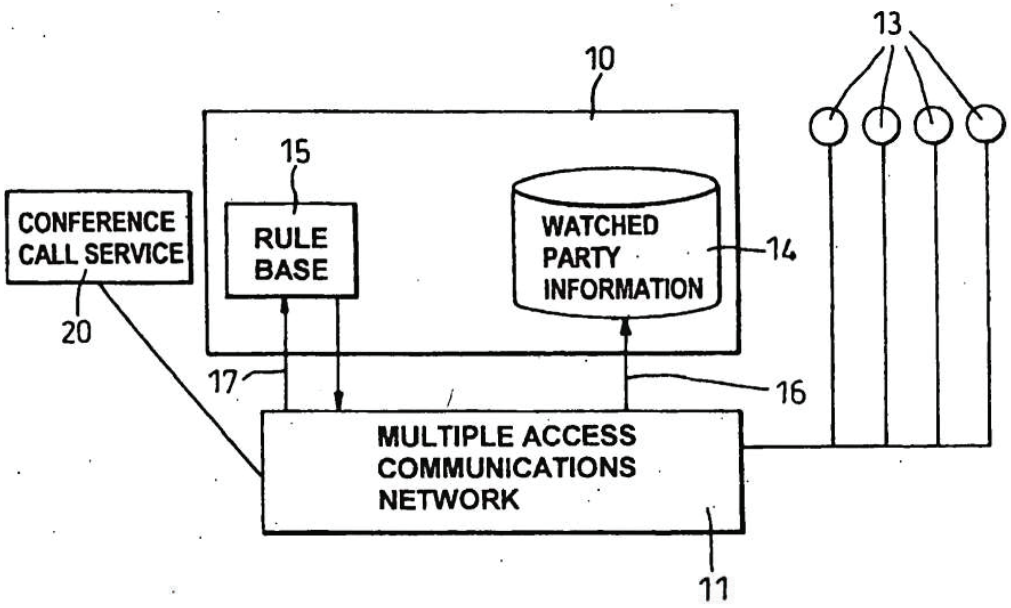

13
10
15
CONFERENCE CALL SERVICE
RULE BASE
WATCHED PARTY INFORMATION
14
20
17
16
MULTIPLE ACCESS COMMUNICATIONS NETWORK
11

Fig. 2

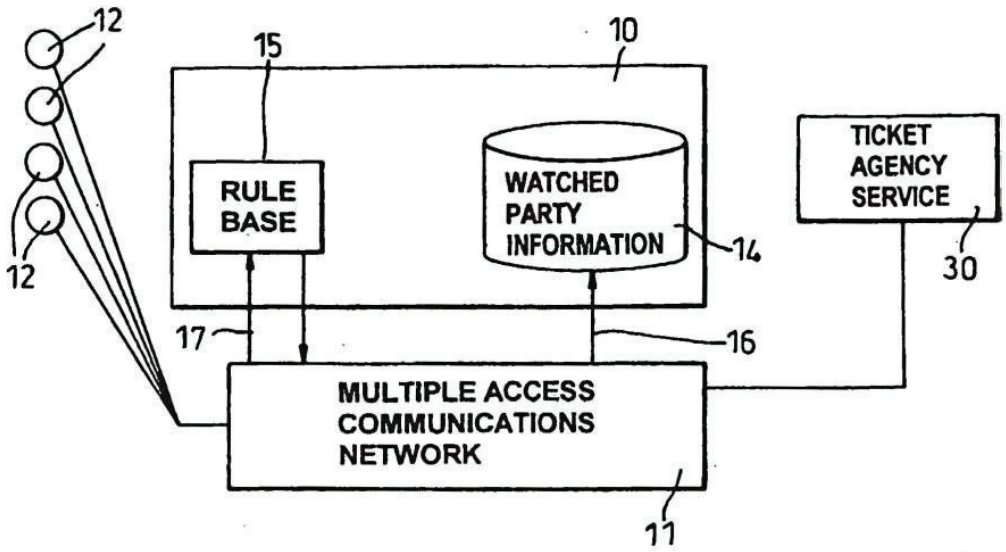

12
15
10
RULE BASE
WATCHED PARTY INFORMATION
14
TICKET AGENCY SERVICE
30
12
17
16
MULTIPLE ACCESS COMMUNICATIONS NETWORK
11

Fig. 3

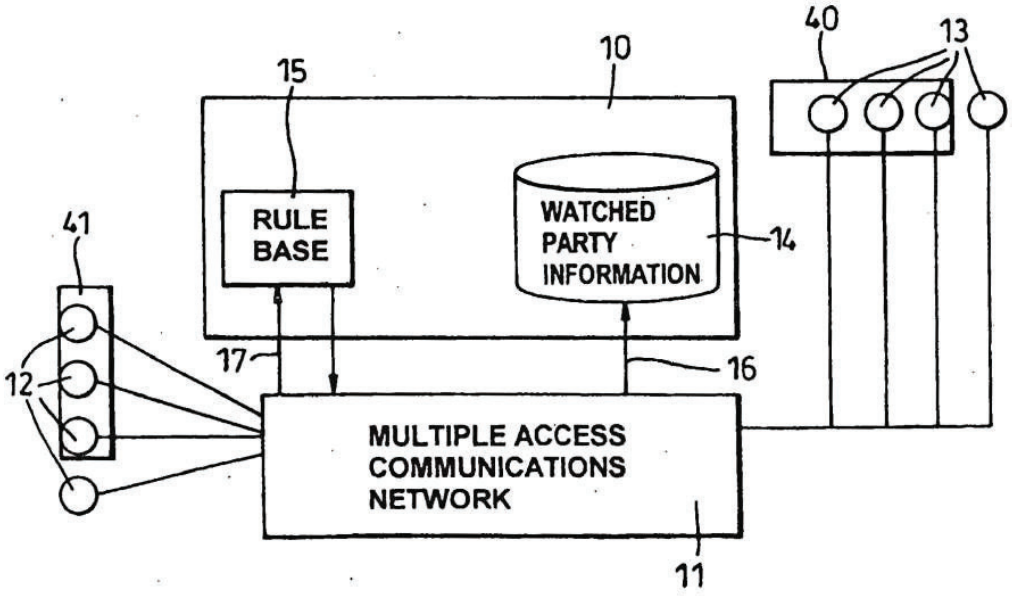

40
13
10
15
RULE BASE
WATCHED PARTY INFORMATION
14
41
12
17
16
MULTIPLE ACCESS COMMUNICATIONS NETWORK
11

Fig. 4

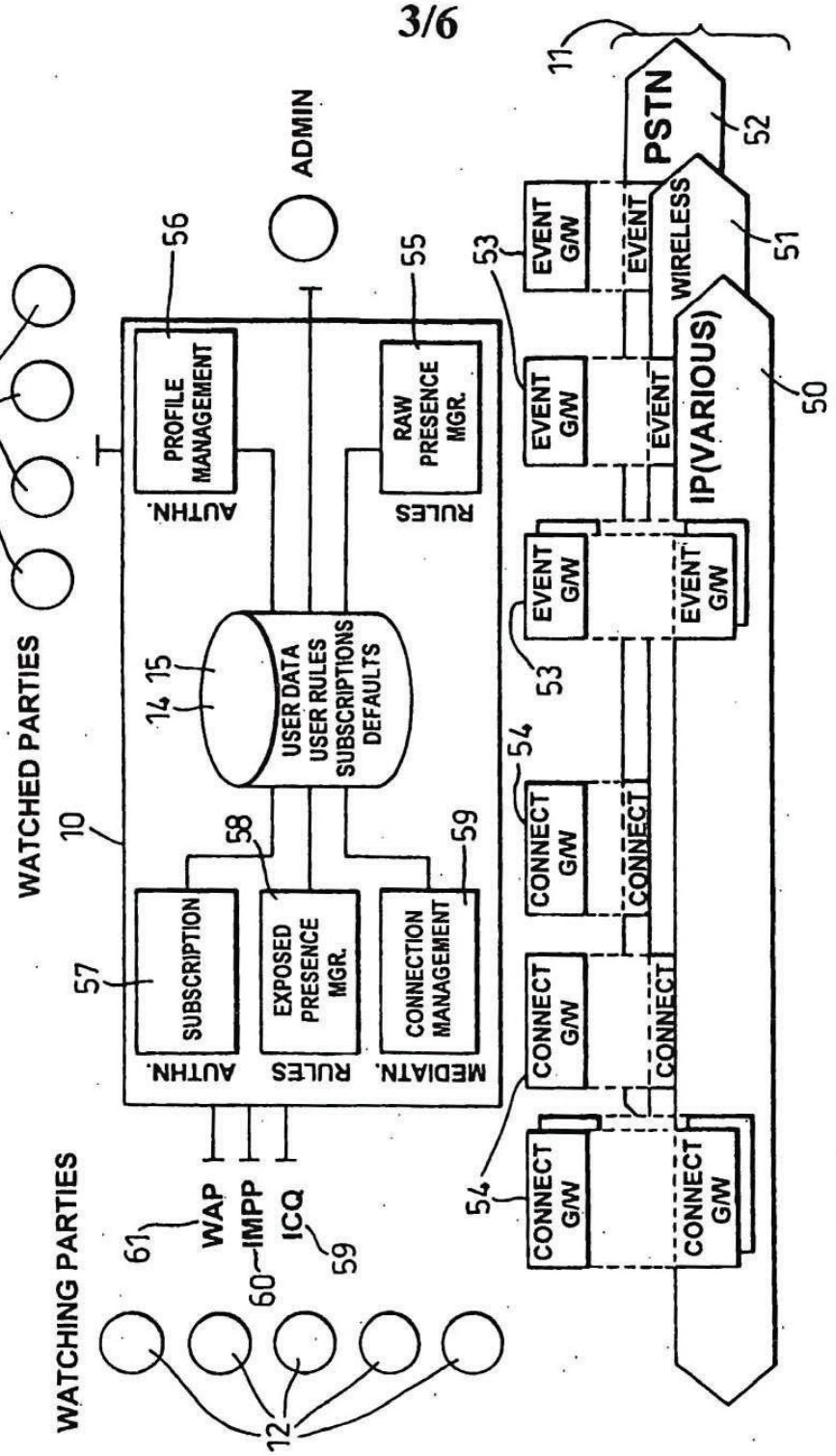


Fig. 5

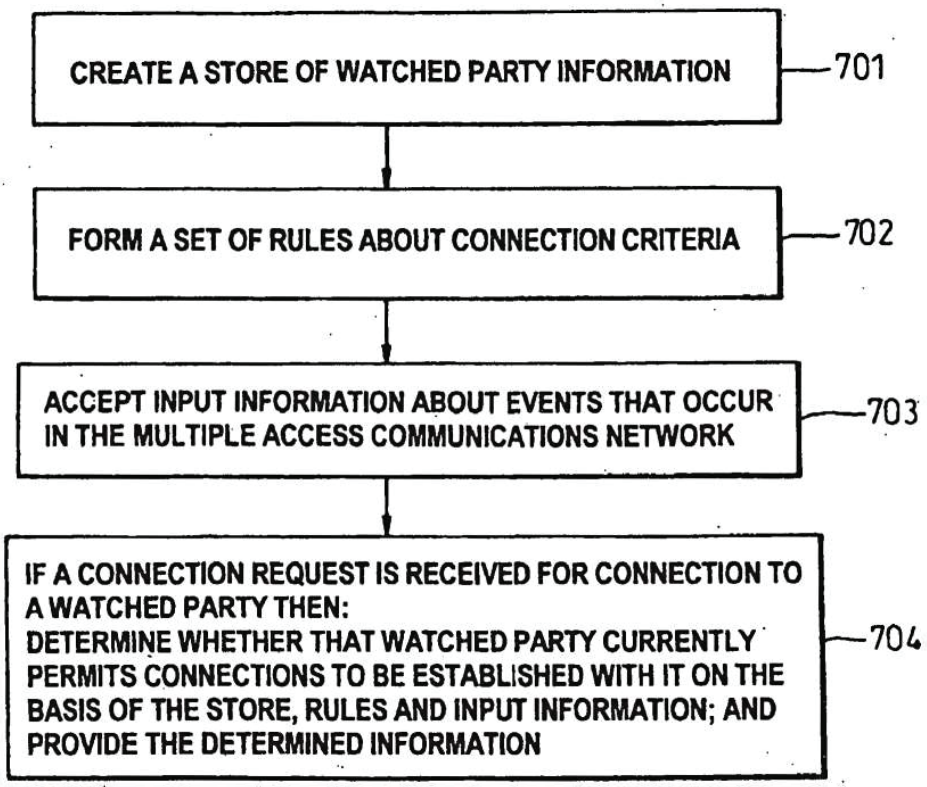

CREATE A STORE OF WATCHED PARTY INFORMATION
701
FORM A SET OF RULES ABOUT CONNECTION CRITERIA
702
ACCEPT INPUT INFORMATION ABOUT EVENTS THAT OCCUR IN THE MULTIPLE ACCESS COMMUNICATIONS NETWORK
703
IF A CONNECTION REQUEST IS RECEIVED FOR CONNECTION TO A WATCHED PARTY THEN:
DETERMINE WHETHER THAT WATCHED PARTY CURRENTLY PERMITS CONNECTIONS TO BE ESTABLISHED WITH IT ON THE BASIS OF THE STORE, RULES AND INPUT INFORMATION; AND
PROVIDE THE DETERMINED INFORMATION
704

Fig. 6

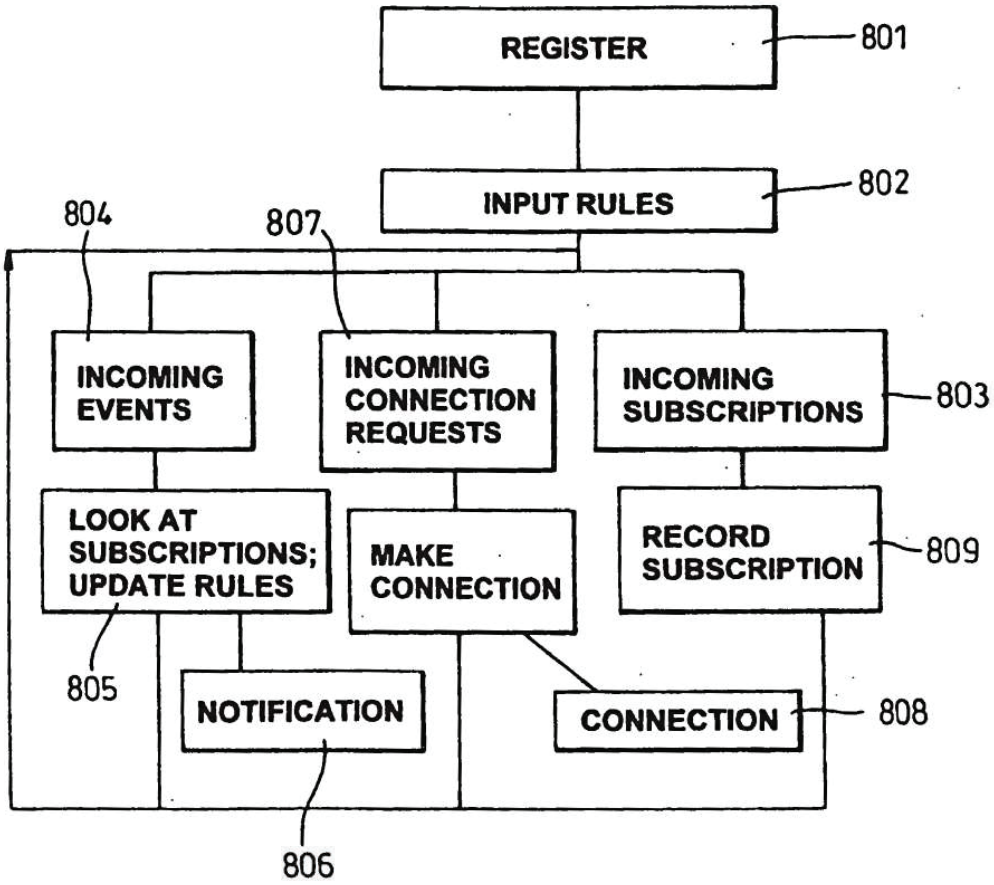

REGISTER
801
INPUT RULES
802
804
807
INCOMING EVENTS
INCOMING CONNECTION REQUESTS
INCOMING SUBSCRIPTIONS
803
LOOK AT SUBSCRIPTIONS; UPDATE RULES
MAKE CONNECTION
RECORD SUBSCRIPTION
809
805
NOTIFICATION
CONNECTION
808
806

Fig. 7

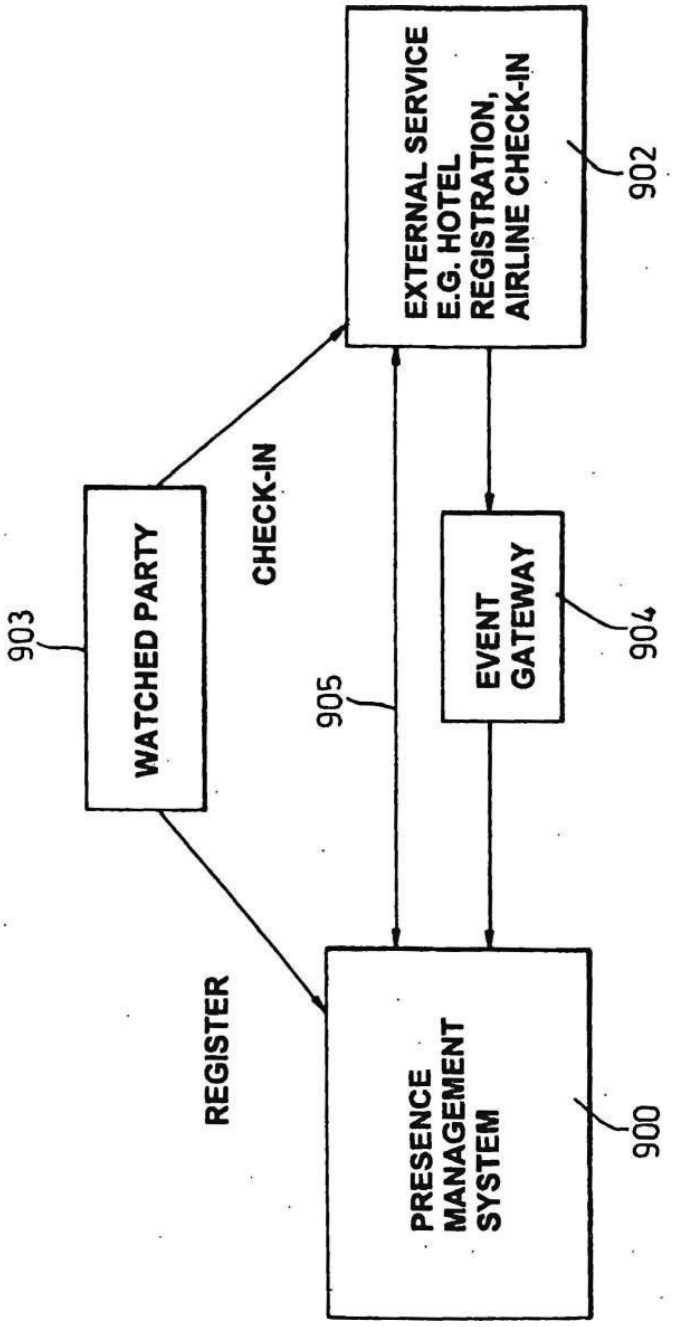

EXTERNAL SERVICE
E.G. HOTEL
REGISTRATION,
AIRLINE CHECK-IN
902
WATCHED PARTY
903
CHECK-IN
EVENT
GATEWAY
904
905
REGISTER
PRESENCE
MANAGEMENT
SYSTEM
900

Fig. 8

« Ocri2006 Showcase
Ottawa Asterisk User Group kickoff »

iotum History

You may, or may not, know the history of iotum. We incorporated in December 2003, and hired our first developer, Todd Jefferson in August 2004. From August 2004 until March 2005, we experimented with a number of different product concepts. We built, tested, and threw away prototypes until we arrived at the current concept for iotum. It was codenamed Boomerang Server Edition, or BSE.

Last Friday, the team updated our desktop plugins and server code, based on several months of feedback from our first beta. This morning we released our Asterisk integration kit to the web, and also our first broad beta. And, partway through the morning, I got this email from Todd.

With this commit into Subversion:

---

r516 | andrewh | 2005-03-14 15:36:26 -0500 (Mon, 14 Mar 2005) | 2 lines

Continuous integration for #305:

Beginning to port the classes from the old rule code.

Boomerang Server Edition was born.

621 tests,

1396 webapp tests,

6 platform/switch integrations,

~110000 lines of code

3 awards

3 patent applications

and 385 days later, we release.

Cool! Congratulations to the team. It's a great accomplishment.

This entry was posted on Monday, April 3rd, 2006 at 11:07 pm and is filed under Business, News. You can follow any responses to this entry through the RSS 2.0 feed. You can leave a response, or trackback from your own site.

9 Responses to "iotum History"

1. *1.1 Million Lines -- Alec Saunders .LOG* Says:
   April 3rd, 2006 at 11:14 pm

   [...] A little iotum history was made today. Read about it on SimplyRelevant. [...]

2. *Martin Dufort* Says:
   April 4th, 2006 at 8:14 am

   Ah!!! The mythical LOC number. Funny how time passes and we are still sticking to that figure!!!. However, I think the best part of this post is the number of tests, close to 2000. I think the best figure to publish is the ratio between TLOC (T=Testing) and ALOC (A=Application).

   Striving for a high number within this area is key and this is the difference we are now seeing between projects done at the end of the 80's / early 90's and today's project.

   Congratulations and give us a sense of the T/A LOC ratio.

   Bye - Martin

3. *Frank Miller* Says:
   April 4th, 2006 at 8:16 am

   I'm confused. Is the 1.1 Million supposed to be 110,000 lines of code?

4. *Randy Charles Morin* Says:
   April 4th, 2006 at 10:08 am

   Congrats!

5. *Rick Duff* Says:
   April 4th, 2006 at 10:30 am

   Was the lines of code 1.1M or 110k? (your title says one thing, the text another)

   Regardless, congrats are in order.

   Rick..

6. *Alec* Says:
   April 4th, 2006 at 10:49 am

   Yes - I must be getting old! Dropped a zero. Will correct it.

   A

7. *Jim* Says:

April 8th, 2006 at 5:26 pm

Congratulations, Alec!

8. *Jim* Says:
April 8th, 2006 at 5:42 pm

And congratulations to Todd and the other contributors, I better add!

9. *Alec* Says:
May 8th, 2006 at 10:11 am

Testing the new capcha tool to block comment spam.

Leave a Reply

Name (required)

Mail (will not be published) (required)

Website

You must read and type the **5 chars** within **0..9** and **A..F**, and submit the form.

submit

Oh no, I cannot read this. Please, generate a new ID

Simply Relevant is proudly powered by WordPress

iotum™
SIMPLY RELEVANT

ABOUT IOTUM™

OUR SOLUTION

MOBILE PROFESSIONALS

SERVICE PROVIDERS

NEWS

SIGNUP TO BETA TEST IOTUM™

Our Solution

IOTUM – Innovating the Frontiers of Communications Services

IOTUM is revolutionizing communications by enabling busy professionals to connect with the right people at the right time.

Partnering with telephony, cellular and VoIP companies worldwide, IOTUM is driving the development of breakthrough communications services that transform the way people work – making them more creative, productive and responsive.

The company's patent-pending technology was recently awarded the 2005 Product of the Year Award by Internet Telephony magazine.

The iotum™ Relevance Engine™

The iotum™ Relevance Engine™ is the world's first smart platform to intelligently filter, rank and prioritize calls based on their relevance to you.

The typical office worker is interrupted every three minutes by a phone call, e-mail, instant message or other distraction. With the ever-rising influx of distractions, how can we stay focused on our priorities? How can we determine which communications are relevant to the task at hand? How can we be assured that important calls will always get through and find us?

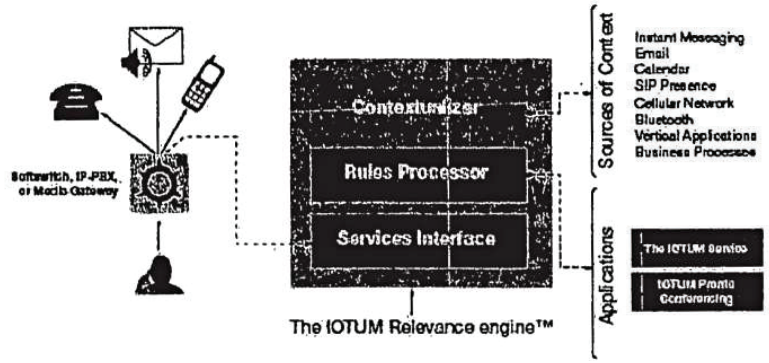
Enter iotum, the world's first smart platform that solves the problem of communication overload. iotum prioritizes all voice communications so that busy professionals can talk to whom they want, when they want, and on the device they want – without changing the way they work or live.

By mapping inbound communications to work behaviour and priorities, IOTUM dramatically improves productivity by ensuring that inbound communications are relevant to the task at hand. IOTUM is simple for users to configure and seamlessly connects to their calendar and instant messaging (IM) tools to determine how specific calls should be handled - making filtering decisions based on who's calling, time of day, what is on the calendar, etc.

iotum products and services meet the needs of today's ultraconnected mobile professionals and service providers

How It Works

The IOTUM service is based on the patent-pending IOTUM Relevance Engine™ – a network-based communications platform that understands who is calling and knows what to do with the call based on the user's preferences, priorities, behaviors, and other contextual information. IOTUM intelligently filters and routes calls - offering an enhanced user experience and enabling the quick deployment of innovative, value-added services.



Like a search engine, IOTUM's Contextualizer constantly indexes and categorizes contextual information. It provides 'what, where, when' information to the Rules Processor, which determines if and how a call should be routed. Utilizing an easily implemented XML-RPCServices Interface, IOTUM connects to your Softswitch, Media Gateway or IP-PBX.

Interested in joining the relevance revolution?
Sign-up to try it

To learn more about the IOTUM Relevance Engine™:
> Datasheet (PDF)
> Voice 2.0: Manifesto for the Future (PDF)
> simply relevant (Blog)

Iotum Corporation 6 Gurdwara Road, Suite 205 Ottawa, ON Canada K2E 8A3
P: +1 613 482-9099 | F: +1 613 482-9098 | E: iota@iotum.com | Legal | Privacy Policy

iotum™
SIMPLY RELEVANT™

We bring relevance to communications

IOTUM is revolutionizing the way people manage their communications. Founded in 2003, IOTUM makes busy professionals more productive and responsive by connecting them with the right people at the right time – ensuring that inbound communications are relevant to the task at hand.

Distractions: Our Biggest Productivity Drain

The typical office worker is interrupted every three minutes by a phone call, e-mail, instant message or other distraction.[1] With the ever-rising influx of distractions, how can we stay focused? How can we determine which communications need our immediate attention and which ones can wait?

IOTUM

IOTUM is a Web 2.0 service that solves the problem of communication overload. It prioritizes all voice communications so that busy professionals can talk to whom they want, when they want, and on the device they want – without changing the way they work or live. By mapping inbound communications to personal priorities, IOTUM dramatically improves users' ability to focus their attention on things that matter most.

How It Works

The IOTUM service is based on the IOTUM Relevance Engine™ – an intelligent communications platform that understands who is calling and knows what to do with the call based on the user's preferences, priorities, behaviors, and other contextual information.

IOTUM is simple to configure and seamlessly connects to users' calendar and instant messaging (IM) tools to determine how specific calls should be handled – making filtering decisions based on who's calling, time of day, what is on the calendar, etc. By paying attention to calling habits, IM presence, schedule, and location, IOTUM can:

- Bring relevant calls to users' attention and send the others to voicemail;
- Dynamically route important calls to cell phones when users are away from their desks or out of the office;
- Notice calling trends, such as back-and-forth calls to the same person, to ensure that important calls are brought to users' attention;
- Proactively contact conference participates and add them to the call – eliminating the need for everyone to dial into a conference bridge and to remember pass codes
- Decide that calls from people scheduled for meetings later in the day should be given higher-than-usual priority – anticipating the potential need to postpone, change or reschedule the upcoming appointment.

With IOTUM, it's simple: *important calls get through, unimportant calls don't*. Users gain productivity – and better still, lose a lot of stress.

That's the power of relevance!

Technology Excellence

IOTUM uses standards-based protocols and has a patent pending on the IOTUM Relevance Engine. The Beta version of IOTUM was released in Fall 2005 and will be commercially available in 2Q2006. IOTUM was awarded the **2005 Product of the Year Award** by *Internet Telephony* magazine.

Our Customers

The IOTUM service is delivered to end-user professionals through communications service providers. Leading the way for relevant communications, IOTUM works with key telecommunications companies around the globe - providing compelling value-add services that:

- Generate additional revenue;
- Create sustainable competitive advantage;
- Improve customer loyalty and reduce churn.

[1] *Cnet News*, Ina Fried, 21 July 2005

Call: +1 613 482 9099 visit: www.iotum.com Blog: www.simplyrelevant.com
Iotum Corporation, 6 Gurdwara Road, Suite 205, Ottawa, ON K2E 8A3, Canada

iotum™
SIMPLY RELEVANT™

We bring relevance to communications

Communication Overload: The Biggest Threat to Productivity

How often does this happen? You're working on a project. And, no sooner than you hit a good pace, the phone rings – completely throwing you off. You'd like to take the phone off the hook, but you're waiting for a colleague to call with a critical piece of information.

You're not alone. The typical office worker is interrupted every three minutes by a phone call, e-mail, instant message or other distraction.[1]

With the ever-rising influx of distractions, how can we stay focused on our priorities? How can we determine which communications are relevant to the task at hand? How can we be assured that important calls will always get through and find us?

It's the challenge of communication relevance.

Deliver More Than Just Calls... Deliver the Right Call, to the Right Place, at the Right Time

Enter IOTUM, the world's first smart platform that solves the problem of communication overload. IOTUM prioritizes all voice communications so that busy professionals can talk to whom they want, when they want, and on the device they want – without changing the way they work or live.

By mapping inbound communications to work behaviour and priorities, IOTUM dramatically improves productivity by ensuring that inbound communications are relevant to the task at hand. IOTUM is simple for users to configure and seamlessly connects to their calendar and instant messaging (IM) tools to determine how specific calls should be handled - making filtering decisions based on who's calling, time of day, what is on the calendar, etc.

By paying attention to calling habits, collaboration needs, IM presence, schedule, and location, IOTUM can:

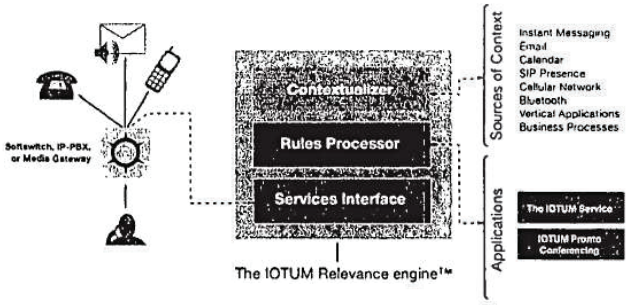
- Bring relevant calls to users' attention and send the others to voicemail;
- Dynamically route important calls to cell phones when users are away from their desks or out of the office;
- Notice calling trends, such as back-and-forth calls to the same person, to ensure that important calls are brought to users' attention;
- Proactively contact conference participates and add them to the call – eliminating the need for everyone to dial into a conference bridge and to remember pass codes
- Decide that calls from people scheduled for meetings later in the day should be given higher-than-usual priority – anticipating the potential need to postpone, change or reschedule the upcoming appointment.

With IOTUM, it's simple: *important calls get through, unimportant calls don't*. Your customers gain productivity – and better still, lose a lot of stress. You make your customers more loyal – and attract new ones, too.

That's the power of relevance!

How It Works

The IOTUM service is based on the patent-pending IOTUM Relevance Engine™ – a network-based communications platform that understands who is calling and knows what to do with the call based on the user's preferences, priorities, behaviors, and other contextual information. IOTUM intelligently filters and routes calls – offering an enhanced user experience and enabling the quick deployment of innovative, value-added services.



[1] *Cnet News*, Ina Fried, 21 July 2005

iotum™
SIMPLY RELEVANT™

Like a search engine, IOTUM's Contextualizer constantly indexes and categorizes contextual information. It provides 'what, where, when' information to the Rules Processor, which determines if and how a call should be routed.

Utilizing an easily implemented XML-RPC Services Interface, IOTUM connects to your Softswitch, Media Gateway or IP-PBX.

Adding Value Drives Your Bottom Line

In today's commoditized telephony market, gaining a sustainable competitive advantage means offering more unique value to customers. By helping solve the distraction problem, providers gain a powerful means to integrate services into the day-to-day work life of customers – giving those customers more reason than ever to stay loyal.

Consider the opportunities:

Make money today
IOTUM is ready out-of-the-box and easily integrated into your network. So, you can instantly add value to your existing services;

Drive sales of your other existing services
IOTUM is great added to a phone line. But, it becomes truly revolutionary once you combine it with wireless, conferencing, instant messaging and Internet services;

Develop even more value-added services
IOTUM provides a comprehensive platform with a services interface designed to ease third-party development – enabling you to add even more value to your services.

IOTUM – Innovating the Frontiers of Communications Services

IOTUM is revolutionizing communications by enabling busy professionals to connect with the right people at the right time.

Partnering with telephony, cellular and VoIP companies worldwide, IOTUM is driving the development of breakthrough communications services that transform the way people work – making them more creative, productive and responsive.

The company's patent-pending technology was recently awarded the 2005 Product of the Year Award by *Internet Telephony* magazine.

FEATURE	BENEFIT
Market-ready IOTUM Relevance Engine™	Go to market instantly with IOTUM, an intelligent call filtering and routing application that integrates with your existing services. It ensures mobile professionals never miss an important call.
IOTUM Pronto Conferencing	Make conference calls more productive. With the new conference call feature, the first person that calls the IOTUM number triggers the other participants to join eliminating conference bridge numbers and pass-codes.
Single point of contact	Users only need to publish one number – IOTUM automatically directs calls across devices, networks and providers.
Context awareness using standard information tools	Beyond mere presence, IOTUM assesses user context by linking seamlessly with existing MS Outlook and MSN Messenger tools.
Extensible plug-in architecture	IOTUM gets more intelligent as plug-ins that gather more contextual information are added to its extensible architecture.
Network agnostic	Open and proprietary call protocol support allows for low-cost implementation with your existing infrastructure. IOTUM leverages IP convergence and works across multiple networks and presence systems.
XML RPC interface and snap-in architecture	IOTUM integrates within a matter of days and transitions seamlessly to next-generation networks.

Call: +1 613 482 9099 visit: www.iotum.com Blog: www.simplyrelevant.com
Iotum Corporation, 6 Gurdwara Road, Suite 205, Ottawa, ON K2E 8A3, Canada

The Mobile People Architecture

Guido Appenzeller Kevin Lai Petros Maniatis Mema Roussopoulos
Edward Swierk Xinhua Zhao Mary Baker
{appenz, laik, maniatis, mema, eswierk, zhao, mgbaker}@cs.stanford.edu
http://mosquitonet.stanford.edu

Techical Report: CSL-TR-00000
January 1999

Computer Systems Laboratory
Departments of Electrical Engineering and Computer Science
Stanford University
Stanford, California 94305-9040

Abstract

People are the outsiders in the current communications revolution. Computer hosts, pager terminals, and telephones are addressable entities throughout the Internet and telephony systems. Human beings, however, still need application-specific tricks to be identified, like email addresses, telephone numbers, and ICQ IDs. The key challenge today is to find people and communicate with them personally, as opposed to communicating merely with their possibly inaccessible machines—cell phones that are turned off, or PCs on faraway desktops.

We introduce the *Mobile People Architecture*, designed to meet this challenge. The main goal of this effort is to put the *person*, rather than the devices that the person uses, at the endpoints of a communication session. This architecture introduces the concept of *routing between people*. To that effect, we define the *Personal Proxy*, which has a dual role: as a *Tracking Agent*, the proxy maintains the list of devices or applications through which a person is currently accessible; as a *Dispatcher*, the proxy directs communications and uses *Application Drivers* to massage communication bits into a format that the recipient can see immediately. It does all this while protecting the location privacy of the recipient from the message sender. Finally, we substantiate our architecture with ideas about a future prototype that allows the easy integration of new application protocols.

Keywords

Personal Mobility, People-level Routing, Personal Proxy, Personal Online Identity

I. Introduction

One of the defining trends of the 1990s has been the explosive growth of the Internet. A growing number of people have Internet access at work, at home, and on the road. Meanwhile, other types of networks, such as cell phones and pager networks, are proliferating rapidly. In the next decade, more and more people will expect ubiquitous network access—the ability to communicate with anyone, anywhere. These trends present us with a number of challenges:

Enabling ubiquitous reachability. Most people will continue to use a variety of network-enabled devices and applications to communicate with others. The notion of a one-size-fits-all communication device is just as misguided as a universal network link or operating system. Basic tradeoffs like weight, speed, and ease of use will not vanish anytime soon; in the meantime, people will use different devices and applications at different times. Our ideal of ubiquitous network access cannot be achieved unless people can be reached regardless of the communication devices or applications they choose to use.

Maintaining location privacy. Enabling ubiquitous network access unfortunately makes privacy issues even more urgent than they are now. A system that keeps track of how a person is reachable and distributes that information without limits could be used to deduce the person's location and compromise his privacy. Ideally, people should be able to receive messages anywhere, without revealing their whereabouts to the entire world.

Thwarting "spam." Receiving unwanted messages is another type of invasion of privacy. Many messaging applications have no way to deliver messages unintrusively. For example, most telephones can either ring or not ring when a call arrives, instead of ringing for some callers and taking a message for others, or ringing during the day and taking a message at night. Users should be able to have all their incoming communications prioritized and filtered on their behalf.

Converting among protocols. Not all application-layer communication protocols can be used by all devices. For example, most phones are not capable of receiving email. Optimally, communications would be converted automatically from the sender's preferred type to the recipient's preferred type.

We have designed the Mobile People Architecture (MPA) to address each of these challenges. In Section II, we describe how MPA fits into the big picture of networking. In Section III, we give an overview of MPA. In Section IV, we describe the design of MPA by giving detailed descriptions of four different usage scenarios. In Section V, we describe the functions of the Personal Proxy, the key component of the MPA system, which tracks the mobile person and handles communications on his behalf. In Section VI, we describe related work, and in Section VII we state our conclusions.

II. The Role of MPA in the Network Layer Model

In this section, we describe how MPA fits into the overall picture of networking and argue that MPA, or something like it, is a logical extension of the current model of networking.

Networking systems are traditionally organized using a layering model composed of Application, Transport/Network, and Link layers (Figure 1). This model is useful in clearly defining the responsibilities and restrictions of software that exists at each level.

For a layer to be fully implemented, it needs a naming scheme, a way to resolve those names, and a way to route communications. The *Name Types* column of Figure 1 shows the naming scheme that Internet email uses at each layer. Some examples of names are shown in the *Packet Headers* column. These naming schemes usually mandate that the names are unique and change infrequently. In addition, each layer in the figure has a protocol to map its names to lower layer names (the *Name Lookup* column in Figure 1). This mapping facilitates routing a communication to its destination.

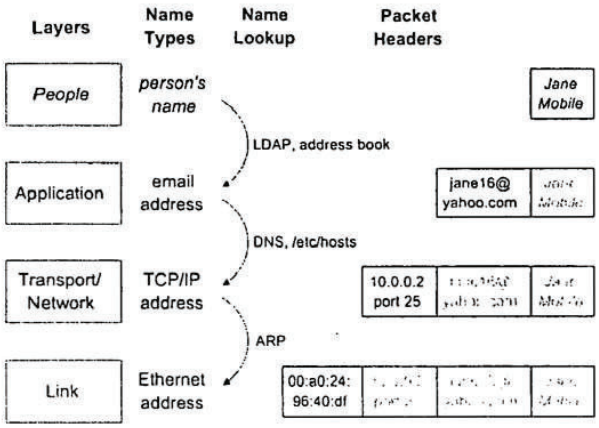


Fig. 1. The Layering Model. We show the traditional networking layers, extended with the *People* layer. *Name Types* shows examples of the kinds of names used at each layer. *Name Lookup* shows some methods of mapping names from each layer to names in the next lower layer. *Packet Headers* shows examples of actual names at each layer, and their relative locations in a typical email packet.

There is one key problem with the traditional layering model: it does not explicitly include people. It seems odd that a communication model would not model people when probably the most important communication is from one person to another.

To model the full process of personal communication, we need to extend the model to include people (the new People layer is shown in Figure 1). Although the layer is new in the model, it is not new in reality. As a result, it is currently implemented in an ad hoc, non-unified way. People are not always named in a unique way, although a name or nickname is often unique among those with whom a person communicates frequently. These names (e.g., Jane Mobile) are resolved into application-specific names (e.g., `jane16@yahoo.com`) using a directory service (e.g., LDAP [WKH97]), an address book, or simply from a person's memory. By directing messages to application-specific addresses, it is the sender who controls their ultimate destination rather than the recipient.

As a result, messaging applications (and therefore their users) have difficulty delivering messages to people who move from one application-specific address to another. For example, if Jane Mobile's email address changes because she travels between home and work, Dan Sender's mail client (and therefore Dan) cannot reliably send email to her. Even worse, if Jane is temporarily unavailable by email, but is reachable by phone, Dan cannot communicate with her until she is available by email. The problem is that Dan cannot identify Jane in a way that is independent of how she is reachable.

The solution is to create a unified implementation for the People layer. Such an implementation needs to name people, map people's names to application-specific addresses, and route communications between people (which we refer to as *people-level routing*). Although the first two functions are partially implemented today, no implementation exists for a people-level router.

The role of a people-level router is similar to that of an IP router: it takes communication from a variety of interfaces and directs it out one or more interfaces, based on the recipient's preferences and on characteristics of the communication itself. The closest current approximation is a human

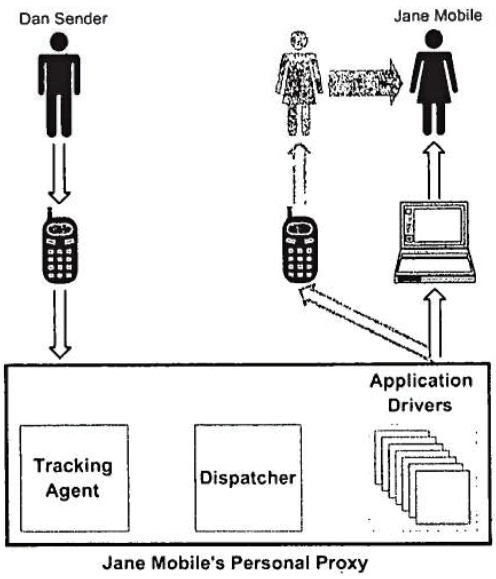


Fig. 2. A Typical Usage Scenario. Dan Sender uses his cell phone to place a call to Jane Mobile. The call is redirected to Jane Mobile's laptop through her Personal Proxy. The gray figure and arrow indicate Jane was recently accessible through her cell phone, but is now accessible through her laptop only.

assistant who answers Jane's phone, reads her email and forwards her messages by calling, emailing, or paging her. Aside from wasting the assistant's time, this implementation would have difficulty forwarding real-time communication (e.g., forwarding an IP telephony call to her cell phone).

The people-level router is a necessary component of any implementation of the People layer. The People layer is a logical extension of the traditional layering model which is the basis of current networking architectures. Therefore, the people-level router is also a logical extension of traditional networking architectures. The MPA implementation of the people-level router is the Personal Proxy, which we describe in greater detail in Section V.

III. Architecture Overview

The main goal of MPA is to route communication to a mobile person, independently of his location or the communication applications he is currently using. This people-level routing uses an addressing scheme that uniquely identifies people. In MPA, these addresses are called Personal Online IDs (POIDs). The architecture does not depend on how POIDs are maintained or how people retrieve the POIDs of other people.

Figure 2 shows a typical usage scenario in which Dan Sender wants to initiate communication with Jane Mobile. If Dan's communication application (which could be anything ranging from email to a fax machine) supports MPA, then it uses Jane's POID to direct communication to her Personal Proxy. If Dan's application is not MPA-aware or if a POID naming scheme is not widely deployed, then an alternate scheme is used (see Section IV).

The Personal Proxy is the heart of MPA and consists of three components: the Tracking Agent, the Dispatcher, and a set of Application Drivers. We briefly describe their functions here, and give

more detailed descriptions in Section V.

The Tracking Agent in Jane's Personal Proxy is responsible for keeping track of her as she moves from an application on one device to another application (possibly on another device). For example, in Figure 2, Jane has switched her cell phone off and is now accessible only via email on her laptop. The Tracking Agent makes this information available to the Dispatcher in her Personal Proxy.

The Dispatcher processes any communication that arrives at the Personal Proxy. Using Jane's accessibility information and her preferences, the Dispatcher directs the communication to the appropriate application. In some cases, the Dispatcher may call upon an Application Driver to convert the communication into a form understandable by the receiving application. In Figure 2, Dan Sender calls Jane on her cell phone. Since she is accessible only via email, an Application Driver converts the voice message into an email message with an embedded sound file. This sound file is then forwarded to Jane's laptop. An Application Driver could also enforce user-specified restrictions (e.g., to block spam), or convert intrusive forms of communication into less intrusive ones (e.g., a phone call into voicemail).

IV. Design

In this section we will describe the design of the Mobile People Architecture by outlining in detail four different usage scenarios between Dan Sender and Jane Mobile. In these scenarios, Dan initiates communication with Jane. We assume the existence of a Personal Online ID (POID) system.

A. Scenarios for MPA-aware Applications

In the following two scenarios we assume that all applications are MPA-aware:

Figure 3(a) shows a scenario in which Jane wants privacy; she does not want to reveal her location to anyone. She also wants to receive communication from Dan, regardless of what application he is using. To achieve these goals, Jane has her Personal Proxy receive communication on her behalf and forward it to her. The Personal Proxy acts as an enhanced online analog to the human assistant referred to in Section II. We show below how the Personal Proxy achieves the goals of privacy and application-independent communication.

Dan enters Jane's POID into his communications application. If Dan is going to communicate with Jane, he knows her POID, just like he knows her real name. The application sends a query with Jane's POID to a Directory Service (DS) such as LDAP. Based on the POID and the application type, the Directory Service returns the relevant *Proxy Application-Specific Address* (PASA) of Jane's Personal Proxy (e.g., `jane@janemobile.nom` for email or 555-1000 for telephony). For each type of application that Jane uses, her proxy has a corresponding PASA. This allows the Personal Proxy to intercept and redirect all communication to Jane's applications which are at undisclosed Application-Specific Addresses (ASA). Some examples are `jane16@yahoo.com` or 123-4567.

Dan's application initiates communication with Jane's Personal Proxy at the returned PASA. Her proxy determines which of her applications should receive the communication. If necessary, it also converts the communication into a different format and then forwards it to Jane's application. Note that at no point is Dan or his application aware of the redirection; this ensures Jane's location privacy.

Figure 3(b) shows a scenario in which Jane does not care to conceal her location. Here, the Personal Proxy does not participate in the communication between Dan's and Jane's applications. Instead, the Personal Proxy updates the Directory Service with the ASAs of Jane's currently available applications. In the figure, we refer to this as *Tracking Info*.

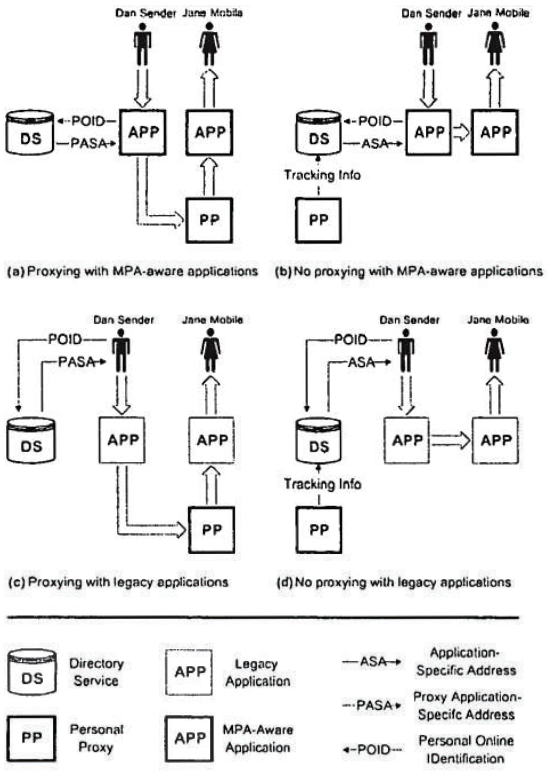


Fig. 3. Four Usage Scenarios of MPA. Wide arrows indicate the transfer of communication data, and thin arrows indicate the transfer of control data.

Dan enters Jane's POID into his communications application. The application sends a query with Jane's POID to a Directory Service (DS) such as LDAP. Based on the POID and the application type, the Directory Service returns Jane's current ASAs.

Dan's application initiates communication directly with Jane's application using the returned ASA. While this scenario is more efficient than the first scenario, it does not offer the same privacy and application-independent communication benefits.

B. Scenarios for Legacy Applications

In the following two scenarios we assume that no applications are MPA-aware. We illustrate that MPA is flexible enough to support legacy applications.

Figure 3(c) shows a scenario in which Jane desires privacy and application-independent communication. Since Dan's application does not recognize POIDs, Dan must manually query the Directory Service to obtain Jane's PASA. Dan feeds the PASA into his application. The application sends the communication using the PASA as a destination address. The communication is routed to the Personal Proxy. As before, the Personal Proxy determines which of Jane's applications should receive the communication. If necessary, it converts the communication and then forwards it to Jane's application using that application's ASA.

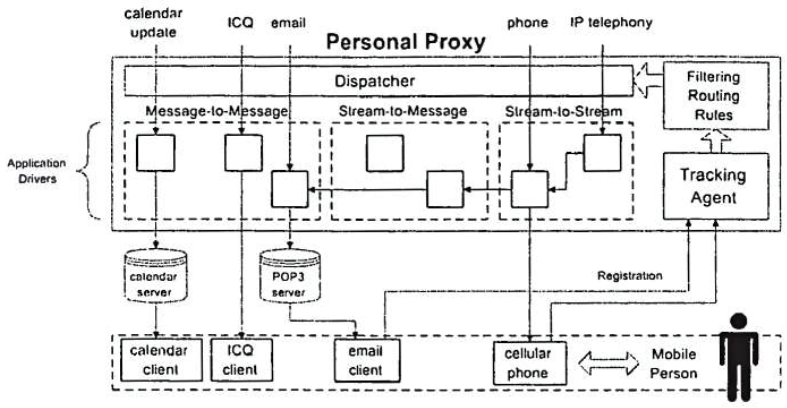


Fig. 4. The Personal Proxy Design. Various communication types are shown entering the Personal Proxy through the top. Arrows indicate possible routes through the Dispatcher and the Application Drivers to the mobile person's applications.

The scenario in which Jane does not care to hide her location from Dan, is similar to the second scenario in the previous subsection. The only difference is that Dan must query the Directory Service to obtain the ASA of Jane's available application. We illustrate the scenario in Figure 3(d).

C. Sender Privacy

In the scenarios above we emphasize the receiving person's preferences for privacy. MPA is flexible enough to support location privacy for the sender as well as the receiver. If a sending person has requested location privacy, all of his communication must travel through his Personal Proxy. If his application is MPA-aware, this is straightforward: he configures the application so that it always sends communication to the PASA of his Personal Proxy. However, if the application is not MPA-aware, we need to perform some kind of application-level encapsulation. That is, the user must incorporate the recipient's POID within the application's data and must set the application's destination address to be the PASA of his Personal Proxy. When the Personal Proxy receives the communication, it must use the receiver's POID to obtain from the directory service the appropriate ASA (or PASA if the receiver has also requested privacy) to use when forwarding the communication.

V. Personal Proxy Design

The Personal Proxy performs a number of key functions in the MPA system: it keeps track of the mobile person's whereabouts; accepts incoming communications on the person's behalf; converts or filters communication data; and delivers communications to the correct Application-Specific Address.

The general design of the Personal Proxy is shown in Figure 4. The Tracking Agent keeps track of the mobile person's whereabouts. Meanwhile, communications arrive through a variety of application-specific protocols, shown at the top of the diagram. The Dispatcher uses the Filtering Routing Rules (derived from the person's preferences) and his current location (obtained from the

Tracking Agent) to determine which Application Drivers should be invoked to convert or filter the communication. The Dispatcher then sends the communication to the correct application.

In the following sections we discuss the components of the Personal Proxy in more detail.

A. Tracking Agent

A.1 Tracking

The Tracking Agent keeps track of the applications through which a mobile person is most likely to be accessible at a particular time. The person conveys this information by registering applications with the Tracking Agent. A registration does not guarantee that the person will be accessible through this application; it merely indicates that he is likely to be reached at the registered application.

A.2 Registration

An application can be registered in a variety of ways; the method used depends on the application type and user preferences. The registration can be manual, automatic, polled, or based on some user-specified profile:

- *Manual registration* requires the mobile user to perform some task to indicate that he is likely to be accessible through an application. He might enter his username and password into a secure web page or dial a particular phone number and enter a personal code. The user might provide an estimate of how long he expects to use the application, or might perform another manual task to deregister the application.
- *Automatic registration* relieves the user from any manual task; instead, the application or operating system senses a user's presence and automatically registers with the Tracking Agent. For example, a device might assume that a user is present when he turns on the device, or when it detects that the user's "smart badge" is within range. This is just a hint that the user is present; it is the responsibility of automatic registration mechanisms to maximize the probability that this information is accurate while still being user-friendly. This automatic type of registration requires new software on the device.
- *Polled registration* requires no effort from either the user or the application. The Tracking Agent periodically polls each of a user's applications or devices to detect if the application is running or the device is turned on. For example, polling might be done by pinging the device or by sending a message to the application and waiting for an acknowledgement. Polled registration is not practical for certain devices, such as one-way pagers.
- *User-specified profiles* allow users to specify *a priori* which applications they are likely to be using in the future. A user might have a profile that indicates the days and times he is likely to use each application. The user can modify the profile as often as desired. Although this option does not provide dynamic detection of active applications, as the previous methods do, it is simple to implement and may be the only feasible option for receive-only devices like one-way pagers.

B. Dispatcher

The Dispatcher receives incoming communications and decides whether the data need to be processed through an Application Driver. It bases this decision on preferences set by the recipient (e.g., "Send all ICQ communications to my pager") and information from the Tracking Agent about the recipient's location. If the message needs to be converted, the Dispatcher attempts to find the

Application Drivers necessary to convert the communication to the appropriate application-specific protocol.

To make the Personal Proxy easily extensible, the Dispatcher is not able to make decisions based on application-specific properties of the communication. Actions such as "Throw out all emails that contain Java objects" have to be made by Application Drivers, which are described in the next section.

C. Application Drivers

A vital goal of MPA is to allow for the easy integration of new applications. To achieve this goal, we limit application-specific knowledge to small, modular blocks called Application Drivers. All other parts of MPA have to know only what types of application-specific protocols there are, not the details of each protocol.

Ideally, a driver should be able to convert any communication to any other format; however, in practice it is doubtful whether this is practical or desirable. In some situations the best a driver can do is to embed one format within another (e.g., embed a voicemail message as a sound file within an email message). Generally we can distinguish four types of drivers:

- *Stream-to-stream drivers* transform one live stream into another. This is currently most likely to be of interest for voice-based applications. Since these drivers must perform complex operations under stringent real-time constraints, they are the most difficult to design.
- *Stream-to-message drivers* convert a live stream into a message. If a user is contacted with an interactive streaming format but is only reachable by non-interactive message types, the Personal Proxy can use this type of driver. This is similar to the functionality of a voicemail system.
- *Message-to-message drivers* convert one message format to another, a process which need not be performed in real time.
- *Filter drivers* enable one of the key features of MPA: enabling users to direct incoming communications based on application-specific properties, such as keywords.

It is important to note that while the Application Drivers are shown in Figure 4 as being completely contained within the Personal Proxy, a driver can be built using a client-server model. Most of the real work could be accomplished by a server on a different machine; the local driver would be a simple stub that communicates with the server. The Berkeley NINJA Project [GWBC99] demonstrates the potential of such a system (see Section VI).

Message storage, performed in the figure by the *calendar server* and the *POP3 server*, is outside the scope of the Personal Proxy. In our design, we leverage the support for message storage already provided by applications.

VI. Related Work

Several projects and products are related to our work on MPA. This is a very good indication of the growing interest in supporting convenient and instant communication with people on the move. While these other efforts share goals with our project, they do not provide a coherent end-to-end model that integrates people with the communications hierarchy. In our model, *people* are the ultimate endpoints.

The AT&T Easy Reach 500 Service [ATT] and the ever-popular instant messaging schemes, such as ICQ [ICQ] and AOL's Instant Messenger [Ame], clearly reflect people's desire to stay connected. The 500 Service is somewhat primitive. It does not track the owner of the 500 number; instead, it calls a predetermined list of numbers in turn, until somebody answers. The instant messaging services use proprietary naming schemes, thus hindering interoperability.