Project Number:	**IST-2001-32603**
Project Title:	**6NET**
CEC Deliverable Number:	**32603/IBM/DS/D5.2/A1**
Contractual Date of Delivery to the CEC:	30 June 2002
Actual Date of Delivery to the CEC:	11 July 2002
Title of Deliverable:	**Identification of User Community for Activities 5.3 and 5.4**
Work package contributing to Deliverable:	WP5
Type of Deliverable*:	S
Deliverable Security Class**:	PU
Editors:	IBM France
Contributors:	Workpackage 5

* Type: D - Demonstrator, O – Other, S – Specification, P - Prototype, R - Report,
** Security Class: PU- Public, PP – Restricted to other programme participants (including the Commission), RE – Restricted to a group defined by the consortium (including the Commission), CO – Confidential, only for members of the consortium (including the Commission)

Keywords:
IPv6, Applications, E-business solutions, Edge services, Users

Abstract:

Deliverable D5.2 lists and briefly describes the communities identified as potential users for the applications listed in deliverable D5.1.

This document deals with users communities for activities A5.3 “E-business solutions” and A5.4 “Edge Services for IPv6”. It will be complemented by deliverable D5.4 that will describe the users communities for activities A5.1 “Real-time video-conferencing and media streaming” and A5.2 “On-line games”.

This first version of the deliverable is limited to describing the profiles of the users communities for the different applications. Subsequent versions will identify specific users either within the personnel of different 6NET partners, from the National Research Networks or from universities or other bodies interested in exercising the 6NET applications.

Table of Contents

INTRODUCTION

This deliverable lists and briefly describes the communities identified as potential users for the applications listed in deliverable D5.1 for activities A5.3 “E-business solutions” and A5.4 “Edge Services for IPv6”.

This first version of the deliverable just describes the profiles of the users communities for the different applications. Subsequent versions will identify specific users either within the personnel of different 6NET partners, from the National Research Networks or from universities or other bodies interested in exercising the 6NET applications.

For each application, a brief functional overview is provided followed by the description of the users community.

1 Activity 5.3: E-business solutions

1.1 IBM Websphere Portal Technology

Overview of the application:

6NET portals will provide to their associated community of users a secure, single point of access to diverse information and applications. Each community will be able to have its own personalized environment (for instance for the support of a national language). Thanks to the communication environment of 6NET and the use of appropriate application level protocols, the information and applications will be integrated and shared seamlessly and easily providing a potential wide collaborative environment. The portals will also handle information on the users that will allow presenting a per-user customized view. Among applications, the portals will provide the secure environments to be associated with commerce applications.
The 6NET portals will be developed with the use of existing IBM Websphere Portal Server technology associated with existing IBM Websphere Edge Server technology and IBM Websphere Commerce Suite technology.

User community:

The population of 6NET connected users is made of coherent communities interested in accessing similar information and applications requiring common communication tools. A portal for such a community will consolidate all those needs in a controlled secured nutshell that will be the default application entry point on the network.. In the background, the community of developers will make available services and applications that will be shared and accessed through the use of web services technologies.

1.2 GLOBUS 2.0

Overview of the application:

The GLOBUS toolkit is an open source middleware suite that supports Grid computing.

Quoting from the www.globus.org web site, “The Globus Project is a community effort, led by Argonne National Laboratory and the University of Southern California’s Information Sciences Institute”.

Globus is developing the basic software infrastructure for computations that integrate geographically distributed computational and information resources. The Toolkit is first and foremost a "bag of services," a set of useful components that can be used either independently or together to develop useful grid applications and programming tools... This release includes new many features, including the Globus Project's Data Grid software, MDS-2, and GRAM 1.5.

This release is also the first Globus Toolkit release to use NCSA's Grid Packaging Technology (GPT), and the first to offer binary releases on popular platforms including Linux 2.x, Solaris 8, Compaq's Tru64, IRIX 5.1, and AIX 5.1."

User community:

The Globus Toolkit is one which is being used increasingly by Grid and E-Science users to enable high capacity distributed computing.

A number of NRENs within the 6NET project already serve established Grid communities, e.g. in the UK there are nine regional E-Science centres who exchange data over the JANET infrastructure. One such centre is located at the University of Southampton, a 6NET project partner. There is also significant Grid effort at UCL, another 6NET partner, where the view is not
if IPv6 should be used for E-Science applications, but when.
Thus we can foresee at least national activity in the UK between Southampton and UCL, initially between researchers, and then also similar trials across other NRENs.

In addition to national activities, pan-European projects and initiatives such as the European Data Grid are potential users of an IPv6-enabled Globus infrastructure. It should be possible to enable IPv6 Globus trials between countries across the 6NET infrastructure.

While the NAT deployments that hamper direct peer-to-peer communication between hosts is more prevalent in the commercial domain, IPv6's address space advantage should also prove beneficial
in academic networks, allowing more end user groups to more readily take part in Grid computations.

Globus 3.0 is expected to be ready for public use by early 2003.
We hope, through the 6NET project, to get IPv6 support enabled first as a patch for Globus 2.0, but later as an integral part of Globus 3.0. Having "out of the box" support for IPv6 in Globus 3.0 will allow the IPv6 functionality to reach a much wider user base.

1.3 Agent framework - SoFAR, SLITE

Overview of the application:

The Southampton Framework for Agent Research (SoFAR) is a Java framework used primarily for RMI communication. A "light" version, aimed at use of multicast for service discovery, has recently been developed (SLITE).

The framework provides a registry so that agents can advertise their services and others can find them, and it supports several communication patterns including queries and a publish-subscribe model. The framework is being used to explore the application of software agents to multimedia systems.

The IPv6 advantage lies in the addressability of multiple SoFAR/SLITE devices, who can communicate directly, peer to peer, IPv6's multicast support is also an important feature for SLITE operation.

User community:

UoS has a strong agent-based computing community in our department's IAM group. Existing and future SoFAR applications will be tested under IPv6, and the main SoFAR code made IPv6-aware.
The SLITE "leightweight" SoFAR framework will be made available to researchers at Southampton, where we will investigate further the applicability of the SLITE framework to existing activities,
and also seek to discover new application areas for the SLITE approach.
We hope to release open source code that research communities in 6NET and beyond can trial.

1.4 Hypermedia link services

Overview of the application:

Hypermedia link services are a key component of many of the multimedia applications developed in the research lab at University of Southampton.

A simple link server accepts a query from a client and returns a list of available links. The query could, for example, be a location in a temporal media stream. The key performance factor in link services is latency, particularly if the link service uses referrals or query routing or if there are synchronization requirements with temporal media.

As an output of 6WINIT, in order to experiment with an IPv6-enabled link service, two servers were ported:

- DLS1 is a specialised service with an HTTP interface. It is a standalone reference implementation in use by current research projects. DLS1 is a 'context sensitive' link service that makes use of information about the device making the query, as part of a pervasive computing infrastructure. Hence in addition to porting of the networking code, we are also introducing IPv6 addresses as part of the context handling mechanism.
- DLS2 is based on an LDAP directory service and provides a distributed implementation. DLS2, based on work in collaboration with BT, has recently been interfaced with the agent framework

User community:

UoS' IAM group makes heavy use of hypermedia link services and the DLS architecture. Making this IPv6-ready will enable IPv6 to pervade into other research projects.
As with the work on SLITE, what we intend make our IPv6 ready link services available to other researchers at Southampton and beyond. We believe such services will encourage development of novel services where link information, stream metadata, and other "markup" information associated with a certain piece of media can be sourced from distributed locations. Such architectures could be used for entertainment (e.g. an enhanced distributed jukebox service) or in a teaching environment (where different metadata threads, potentially from different hosts, may be selected by students).

IPv6's enhanced address space enables this goal, and components such as the OpenLDAP server are already ported for IPv6. The user community will initially be researchers, but during the course of the project we hope that demonstrator applications can be made available to a wider end user audience.

1.5 FunnelWeb

Overview of the application:

FunnelWeb is a system that runs on a node to provide an active services platform. Active services are loadable objects which provide particular application level functionality in the network. FunnelWeb is an implementation of an Application Level Active Networking (ALAN) active networking execution environment (EE). FunnelWeb was developed initially at UCL and has continued development at UTS, Australia under BT funding.

Specifically FunnelWeb provides an execution environment for java based active applications, known as proxylets. The FunnelWeb EE is termed the Execution Environment for Proxylets (EEP), which provides a java environment with a Remote Method Invocation (RMI) control interface for loading, running, modifying operation and stopping proxylets. The proxylets are java applications implementing the *Proxylet* base class, which exposes methods for initialising, starting, modification of operation, and stopping of the proxylet.

IPv6 functionality is possible using the Java JDK 1.4 which provides IPv6 functionality on Solaris8 and Linux.

User community:

FunnelWeb is deployed at a number of sites globally and provides a platform for a large variety of active services deployment. Thus the user community is potentially very wide, from small group use to large project uses.

1.6 Transcoding Active Gateway – TAG

Overview of the application:

The Transcoding Active Gateway (TAG) was developed to extend the functionality of an earlier tool, known as the UCL Transcoding Gateway (UTG). The implementation was based on the FunnelWeb [ALAN] Active Networking architecture. The key points to the new design were:

- Automatic configuration of a multicast session using the Secure Conference Store.
- Use of Active Networking for locating and positioning a reflection point
- re-multicasting of the reflected media streams on the client
- Modular approach to media relays, implemented as Java proxylets

As mentioned previously, TAG builds upon Funnel Web to provide its functionality. It was essential that components of the system were separated, to provide an easy upgrade when a new version of Funnel Web became available. The TAG client application is separated into two components that communicate using Remote Method Invocation (RMI):

- The Funnel Web EEP component of the client runs the Routing, Discovery and local Reflector proxylets. The Routing and Discovery proxylets are used by the client to identify its location in relation to other parts of the Active Network.
- The user interface component of the client is used to communicate both with the EEP component and with a remote EEP via the RMI interface. The server configuration section of the user-interface allows the user to query the local Routing proxylet for information regarding the current EEPs available and the closest EEP in relation to the local host. Once an EEP has been selected the controls for starting, stopping and configuring media streams are enabled.

TAG has recently been updated to provide an additional mode that allows it to connect a client into a conference VPN and subsequently provide access to the VPN media streams. Using this mode, a client will automatically request to join the VPN once local multicast activity is detected. The join request takes the form of a registration of a local EEP to a Reflector Manager (RM). The registration triggers the RM to notify each node within the VPN to forward media streams back to the registered EEP. In addition the RM can be used to convey rate and access information to each remote node.

User community:

TAG provides active gatewaying functionality which is of particular benefit to mobile users, or users on the edge of connectivity. TAG may also provide access to VPN's for unconnected entities. The gateway will benefit users registered for use with the TAG.

1.7 TZI Stargate

Overview of the application:

StarGate provides call signalling and media transcoding gateway functionality for connectivity between different kinds of endpoints interconnected through different types of networks (hence the name *Gate). This is expected to include in particular:

– Conversion between the three most important call signalling protocols (H.323, SIP, and ISDN) including media stream conversion if necessary;
– Actively accessing Mbone sessions from H.323 endpoints; and
– Inviting H.323 endpoints into Mbone sessions for audio and optional video communications.

StarGate was developed under the joint auspices of TELES and UB TZI. It also contains code from UCL's RAT tool.

The architecture of StarGate also allows for extension of the number of supported call signalling protocols. In addition, if feasible from the standardisation point of view (i.e. the necessary specification are complete and stable), security aspects will be incorporated into the StarGate implementation.

StarGate is conceptually built upon the same general Mbus architecture as AudioGate [AG], with largely different Mbus entities and different interactions between them, of course. AudioGate provides a dial-in point for users on any telephone network and enables them to participate in Mbone audio conferences.

All three of the aforementioned control protocol entities share a core set of Mbus messages to set up, tear down, and monitor progress of a call. In addition, each entity supports protocol-specific Mbus extensions that may not be (easily) mapped to other control protocols. The Mbus controller is expected to understand all these Mbus commands, route incoming messages, and optionally perform translation between different protocols.

Call control messages are intended for interaction with call control and invitation protocols such as H.323 and SIP. They are designed to constitute the union of the call control messaging needed by endpoints, gateways, proxies, multi-point controllers, and gatekeepers. This allows the use of the Message Bus to act as gluing mechanism to create any type of system from roughly the same building blocks. Mbus call control messages are based on a common basic message set defined in the following that will be supported by any kind of call control protocol entity. The basic message set may be augmented by protocol-specific extensions required for protocol specific interactions between a local controller and/or local applications on one side and the respective protocol engine on the other. While the basic Call Control commands have been worked through, they still need to be mapped to H.323, SIP, and ISDN-specific messages.

<u>User community:</u>

The Stargate is a gateway specifically for multimedia participants in SIP, H.323, and ISDN based conferences. The gateway will benefit users within an administrative domain wishing to access conferencing on these protocols.

1.8 LightWeight Directory Access Protocol – OpenLDAP

<u>Overview of the application:</u>

OpenLDAP is used in many middleware applications and it is important that it is available over IPv6. OpenLDAP Software (http://www.openldap.org/) is an open source implementation of the Lightweight Directory Access Protocol (v3), base protocol is specified in RFC 2251. OpenLDAP is a simplification of the X.500 DAP.We have ported OpenLDAP to IPv6 and this is in the official 2.x versions.

We want to get OpenLDAP tested on a wide variety of platforms, and make any necessary changes to make IPv6 work on those. Some changes have already been necessary, since IPv6 stack implementations do things in slightly different ways. We will also see if we can use OpenLDAP as a proxy to give IPv4 clients access to IPv6 LDAP servers and vice versa.

A number of 6NET applications depend on LDAP, and should obviously use it over IPv6. Part of the work will be to assist them with any LDAP IPv6 problems.

User community:

There are probably tens of thousands of OpenLDAP users, and OpenLDAP will by default support IPv6. As operating systems and networks are IPv6 enabled, the number of people using OpenLDAP over IPv6 will grow.
One novel application of OpenLDAP over IPv6 lies in Southampton's tunnel broker service, where OpenLDAP has been used to implement the schema associated with tunnel broker connections. This work has been undertaken on the 6WINIT project, but will be made available in 6NET. The tunnel broker architecture uses FreeBSD, OpenSSH and OpenLDAP, all freely available components.

IPv6-enabled LDAP is also used in the PKI activity described below.

1.9 Public Key Infrastructure – PKI (University of Murcia)

Overview of the application:

The purpose of a Public Key Infrastructure (PKI) is to define the mechanisms and elements needed to manage and enable the effective use of public key encryption technology on a medium or large scale.

The base components are a certification authority, one (or several) registration authorities, and a directory server. Some additional components, like smart cards, time stamping servers, OCSP servers, can be present depending on the services offered by a particular PKI implementation.

The Public Key Infrastructure of the University of Murcia is based on the design and implementation of IPv6-enabled X.509 certification services. It can be used by an organisation to provide its users with a range of public key mechanisms for securing their communications.

The end users of the PKI are able to carry out the majority of their PKI operations using a web browser, e.g. to request a certificate, renew it, revoke it, or look for another user's certificate. The PKI also allows the use of smart cards to store cryptographic information, enabling greater key mobility and also increasing the security of the system.

The most important characteristics of this PKI are:

- It allows certificates to be requested, renewed and revoked for every entity (end user or process) of an organisation.
- It allows the use of an LDAP directory to store the users and Certificate Authority (CA) certificates and Certificate Revocation Lists (CRL).
- Final users can carry out a variety of certification operations from their own web browser or through the Registration Authority (RA).
- Users can use smart cards to store cryptographic information (private key, certificate and CA's certificate). This allows mobility and increases the security of the system.
- It supports the definition of a Certification Policy that will establish the restrictions inside an organisation. This policy is defined by the administrator and is applied in every PKI component (registration authority, certification authority, request server, etc.).
- It is completely developed in Java, allowing the use of any operating system to run an implementation of the PKI.
- It is based on those drafts and standards specified by the IETF inside its PKIX working group.
- It supports the Simple Certificate Enrolment Protocol (SCEP), enabling router certificate requests.
- It supports the Online Certificate Status Protocol (OCSP).
- Time Stamping is implemented in the system.
- The end user interface for the system is IPv6 enabled, e.g. the LDAP server and web server, so final users can access the system using this network protocol.
- Work is underway by the University of Murcia to convert internal communication to use IPv6.

User community:

The PKI provides a wide range of facilities to a range of users. The administrative functions would be a closely controlled activity. General users within an administrative domain may utilise various certification operations. Users running services may also make use of the certification facilities for secured operations.
The PKI for IPv6 will enable secure applications to be demonstrated.
One of the goals of collaboration with the Euro6IX project is to run conferencing between the project participants using SRTP over IPv6. Thus the initial user community for such a demonstrator will be the project members, but the scope of a secure conferencing application is a much wider user community.

2 Activity 5.4: Edge services for IPV6

2.1 Edge Server Proxy

Overview of the application:

IBM Edge Server is a powerful device providing a better service both to users who access information on the enterprise's server and to internal users accessing to the Internet. Such devices are close to the boundary between the enterprise's network and the Internet, that's the reason for the name Edge Server.

Four systems are included in IBM Edge Server: Network Dispatcher, Application Service at the Edge, Content Distribution and Caching Proxy. The Caching Proxy intercepts a request from a client, retrieves the data from content host and sends it back to the client. Although HTTP(S) requests are often done, it can also deal with FTP and Gopher traffic. Caching is done by storing cacheable content before sending it to the client, so next requests to the same content can be delivered more quickly and with saving network bandwidth.

Proxys can be used in two different ways: Forward Proxy when located on the client's network, and Reverse Proxy when located on the server's network.

User community:

Proxy services are already widely spread. Main users are ISPs, Campus, content distribution and e-business networks. In fact, the Proxy is not per se an end application. It is a "gateway" between a user and the server. For the moment, the Proxy prototype deals with IPv4-v6 HTTP traffic, including forward and reverse functions. Due to that, the user community will depend on the HTTP servers or clients' users the Proxy will be used for. For example, it could include users of MUST, WebSphere Portal or CDN.

2.2 Contents Delivery Networking – CDN

Overview of the application:

Content Distribution networks scale and accelerate content services by distribution content at the edge of the network and redirecting client request to the most appropriate edge server by means of a content routing process.

A CDN consists of a content distribution management function responsible for optimizing the distribution of content to the edge of the network, a content routing process to redirect client requests to the closest edge delivery node whereby closest is related to a metric based on RTT and content availability and the edge delivery node serving the content using different protocols like HTTP or RTSP.

6NET will study how the CDN infrastructure can be used to accelerate and scale other application services developed in Workpackage 5.

<u>*User community:*</u>

The CDN user community can be divided into 2 categories:

The first category is addressing the 6NET user side. Implementing CDN technologies in a network like 6NET will improve the user experience, give him better response time and better reliability. Large and rich media contents can be pre-positioned on the edge of the network at strategic locations, chosen as being the closest to the user communities for which this content is of real interest.
As an example, a scheduled e-learning session on brain surgery that contains a large presentation or several videos will only be of interest to the medical students community. All that content can be pre-positioned on edge servers located at medical universities before the scheduled e-learning session starts. All students taking the course from university premises will be redirected to the local cache where the content will delivered to this population in better conditions. However the same content will remain available to other users on the origin server, where it can be reached by anybody that has not been targeted as a user of primary interest to that content.

The other population interested in CDN technologies are Content Providers such as e.g. video producers and teachers producing web-based courses. They will be able to selectively address the population they are interested to reach with that content, by choosing the locations where the content should be cached.

The examples mentioned above are targeted at the academic community - teachers and students. It can be extended to all 6NET users and providers of applications that deliver large et very large multimedia content, accessible for example through web portals.

-- End of document --