INTERNATIONAL TELECOMMUNICATION UNION

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

Y.130

(03/2000)

SERIES Y: GLOBAL INFORMATION INFRASTRUCTURE AND INTERNET PROTOCOL ASPECTS

Global information infrastructure – General

Information communication architecture

ITU-T Recommendation Y.130

(Formerly CCITT Recommendation)

ITU-T Y-SERIES RECOMMENDATIONS

GLOBAL INFORMATION INFRASTRUCTURE AND INTERNET PROTOCOL ASPECTS

GLOBAL INFORMATION INFRASTRUCTURE	
General	**Y.100–Y.199**
Services, applications and middleware	Y.200–Y.299
Network aspects	Y.300–Y.399
Interfaces and protocols	Y.400–Y.499
Numbering, addressing and naming	Y.500–Y.599
Operation, administration and maintenance	Y.600–Y.699
Security	Y.700–Y.799
Performances	Y.800–Y.899
INTERNET PROTOCOL ASPECTS	
General	Y.1000–Y.1099
Services and applications	Y.1100–Y.1199
Architecture, access, network capabilities and resource management	Y.1200–Y.1299
Transport	Y.1300–Y.1399
Interworking	Y.1400–Y.1499
Quality of service and network performance	Y.1500–Y.1599
Signalling	Y.1600–Y.1699
Operation, administration and maintenance	Y.1700–Y.1799
Charging	Y.1800–Y.1899

For further details, please refer to the list of ITU-T Recommendations.

ITU-T Recommendation Y.130

Information communication architecture

Summary

This ITU-T Recommendation outlines an Information Communication Architecture (ICA) to guide the future development of information communication networks and services. The architecture is aimed at enabling applications to communicate information using infrastructural capabilities that are based on middleware service components.

Source

ITU-T Recommendation Y.130 was prepared by ITU-T Study Group 13 (1997-2000) and approved under the WTSC Resolution 1 procedure on 10 March 2000.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications. The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Conference (WTSC), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSC Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementors are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database.

CONTENTS

Page

ITU-T Recommendation Y.130

Information communication architecture

1 Introduction

This ITU-T Recommendation outlines an Information Communication Architecture (ICA) to guide the future development of information communication networks and services. The architecture is aimed at enabling applications to communicate information using infrastructural capabilities that are based on middleware service components.

The ICA provides a functional architecture for those information communication systems in the coming broadband and information era. ICA will take the opportunity to incorporate the advantages of the latest information technologies.

2 Purpose

The ICA is aimed at both user[1] and network operators.

For users the purpose of the architecture is to:

- decouple user from technology changes;
- provide easy access to new services.

For network operators the purpose of the architecture is to:

- ensure effective use of network assets;
- simplify deployment of network technologies;
- enable evolution to new services and network technologies;
- provide smooth transition to the GII (Global Information Infrastructure).

To meet the above, the ICA is designed to:

- identify the functions and behaviours between the user and network facilities in terms of middleware services;
- ensure adequate decoupling between end user and network technologies;
- define functions and behaviour of middleware components;
- identify points at which open interfaces might be defined.

3 Scope and field of application

The scope of the architecture defined in this ITU-T Recommendation covers the following aspects of communication:

a) user-to-user communication;

b) user-to-machine communication (and vice versa);

c) machine-to-machine communication.

---

[1] In this context "user" covers both "end-user" as well as "user" as a client in any client server situation. In addition, content providers and service providers may also be considered to be users.

Thus, the scope is not restricted to a narrow interpretation, i.e. limited to traditional telecommunications products and services. Rather, the scope covers any combination of products and service suppliers involved in provision of the above communications. For example, local area networks, cable networks, IP-based networks, and their related services are included within the scope of ICA, as well as ITU-T defined networks and their related services. Such services may involve the transfer of data, voice, or video, or some combination thereof. Similarly, service providers may include, among others, local area network operators, cable network operators, Internet Service Providers, entertainment service providers, etc. In short, the scope involves any of the parties and technologies involved in the moving of, and/or, the related processing of such information.

This is not to say that the ITU-T will define all the component elements of the architecture. It may be expected that the particular specialisms of other standards organizations internal to, and external to, the ITU-T will be taken into account. In this regard, the architecture is expected to draw on outside expertise of internal and external organizations, and will benefit from the appropriate inter-organizational collaboration.

Accordingly, the ICA will take account of a number of existing architectures from other organizations such as TINA[2], OMG[3], GSM-MoU[4], ODP-RM[5], etc. with an initial focus on functions, reference points, and roles.

3.1 General positioning

The scope of ICA middleware is wide and functionality can be combined in many different ways. Middleware service components can, in general, be presented in terms of three functional aspects:

a) functionality which represents the vertical aspects;

b) functionality which represents the horizontal aspects;

c) functionality which represents infrastructure aspects.

3.2 Vertical aspects

The vertical aspects are shown separately in Figure 1, which illustrates that ICA is positioned to provide functionality to intelligently bridge the gap between user applications and raw transport facilities.

In accordance with the GII framework, as defined in ITU-T Recommendation Y.110, the primary "area of concern" of the ICA in relation to the GII is shown in Figure 1.

---

[2] Telecommunications Intelligent Network Architecture.

[3] Object Management Group.

[4] Global Systems for Mobile Communications – Memorandum of Understanding Association.

[5] Open Distributed Processing – Reference Model.

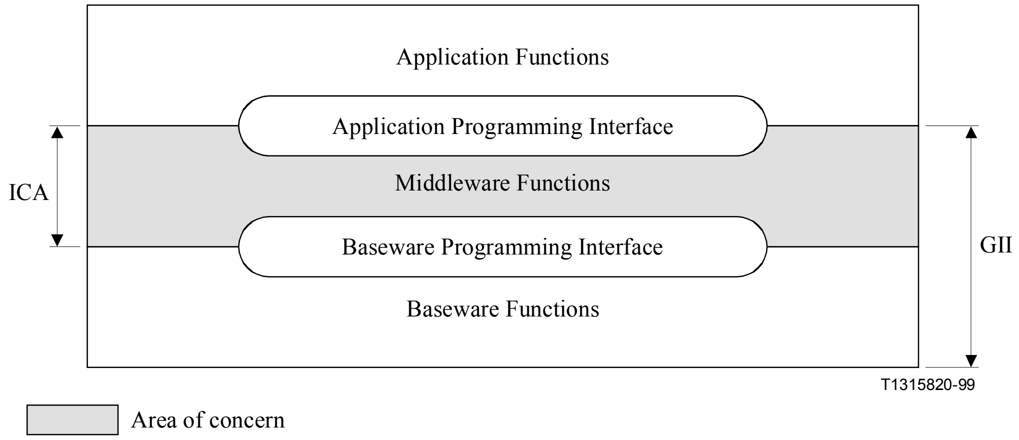

Figure 1/Y.130 – Vertical scope and relationship between the ICA and the GII

ICA middleware functions will, in general, use both GII transport and processing/storage classes of baseware service components. This ITU-T Recommendation addresses middleware interoperating with transport service components. It is envisioned that the ICA model will be expanded to interoperate with processing/storage service components.

3.3 Horizontal aspects

Figure 2, taken from ITU-T Recommendation Y.110 shows the horizontal aspects, in the sense that ICA functionality will be distributed at appropriate points throughout the network, to form a set of geographically distributed cooperating entities. The figure is intended to show that the scope of ICA embraces the distribution of intelligence at various points in the "path" between the end points. This figure is for explanatory purposes only, and is not intended to indicate any specific components or specific distribution thereof.

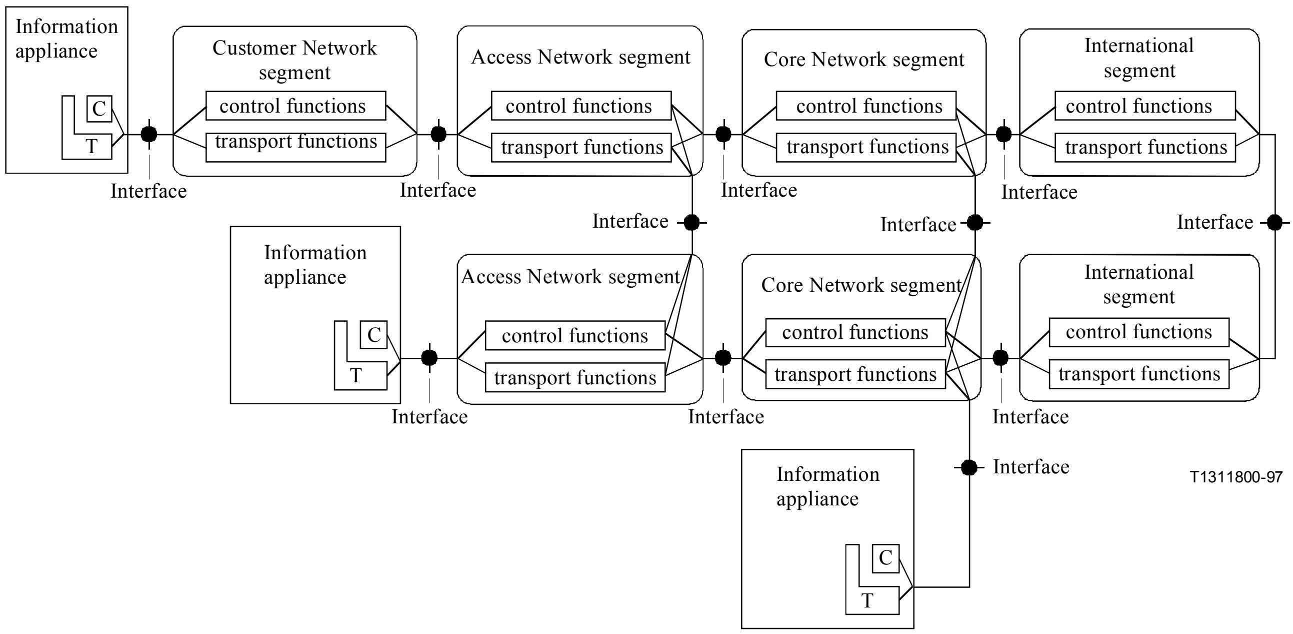

Figure 2/Y.130 – An example of horizontal network components (extracted from ITU-T Recommendation Y.110)

3.4 Infrastructure aspects

The provision of intelligent functionality is at the heart of the ICA, and is the primary subject of the remainder of this ITU-T Recommendation. ICA provides varying degrees of functionality depending on the services requested by end users, and the available network facilities, based on the intention to provide an amount of network-based intelligence commensurate with providing users with network-enhanced services whilst at the same time decoupling users from technological and operational concerns.

4 Design objectives

The major design objectives of the ICA are to ensure that:

4.1 The ICA, initially, focuses on a functional description of middleware architectural components as the first phase of the work. As a second phase the development of reference points and any related interface standards is envisaged. In any case, it should be noted that any work to specify reference points and interfaces in detail will be performed in abstract terms (i.e. independent of implementation).

4.2 The architecture shall be designed to accommodate scalability, i.e. the ICA should not contain any obvious impediments to scalability.

4.3 The ICA provides a unifying framework, with a minimum of prescription, to allow for options and evolution.

4.4 The ICA provides an open unified modular architecture for information communication systems which can be used as a basis to provide services, the management of services, and the management of networks and resources.

NOTE – The word "open" is used to indicate that the ICA has reference points at which it may or may not be appropriate to specify standardized interfaces, that it is open to all stakeholders, and that the ICA is extendible.

4.5 The ICA enables the reuse of specifications and architectural components, rapid service provision, service customization, and flexible deployment of architectural components in the network. The ICA should help achieve interoperability between systems and services located in different administrative domains.

4.6 In this ITU-T Recommendation, the ICA is expressed as concepts and principles. These concepts and principles will provide a basis for the further development of appropriate specifications. The object and interface specifications[6] are subject to constraints that govern objects, which objects should use other objects, and related procedures. In the case that an object-oriented specification technique is used for specifying certain aspects of the ICA, this will not imply that physical realizations must use object-oriented implementation techniques and/or object-oriented technology.

4.7 The ICA enables multi-vendor supply by ensuring a clean separation between:

a) services;

b) operations (service, network provider and resource management);

c) computing platforms for distributed processing;

d) network technology.

---

[6] The method of specification (e.g. ODP-RM's ODL or IDL) is for further study.

4.8 The ICA accommodates the needs of different roles such as user, subscriber, network provider, service provider, network designer, service designer, developer, deployer/withdrawer, network manager, service manager, and service broker.

4.9 The ICA is applicable to communications systems that are partitioned into different administrative domains.

4.10 The ICA is applicable to communications systems regardless of the techniques used to transport user information, control information and management information.

4.11 The ICA allows for a variety of transport techniques at user-network interfaces, including existing ones.

4.12 The ICA is applicable in the construction and operation of a wide range of services including user services.

4.13 The range of services provided by the ICA is open-ended as far as possible, in order to allow adaptation to market needs while protecting investments.

4.14 The ICA allows for a variety of Signalling and management protocols at user-network interfaces, including existing ones.

4.15 The ICA is concerned with generic middleware architectural components (e.g. connection management architectural components).

4.16 The ICA is useful at all life-cycle stages of architectural components, from need capture, construction, deployment, operation to withdrawal.

4.17 The ICA is technology independent, i.e. it should not limit the type of computing technology or hardware that can be employed.

4.18 The ICA work will cover both execution of operations and association/negotiation of stream information related resources.

4.19 The ownership of each entity can be generally separated from the technical development of an architecture for this activity, although such aspects may be covered by referring to some example configuration scenarios.

5 Requirements of the ICA

The ICA shall be designed in accordance with the requirements listed below.

It may not be possible to meet all these requirements. Some trade-offs may be required. Furthermore initial designs of the ICA lack some of the required features, which may be left for a second stage or extension of an initial basic version.

General

5.1 The ICA should be modular.

5.2 The ICA should allow the separation between Terminal, Access and Core (Transport and Control) components.

5.3 The ICA should allow evolution from GSM, ISDN, IN, etc.

5.4 The ICA should define open, secure, generic interfaces and protocols, e.g. Application Programming Interfaces (APIs), Middleware Programming Interfaces (MPIs) and Baseware Programming Interfaces (BPIs).

NOTE – These terms are used as defined in ITU-T Recommendation Y.110.

5.5 Service and management interoperability: The ICA should allow architectural components in different federated administrative domains to inter-operate in a consistent manner for seamless execution of services and management.

5.6 The ICA should allow maximum flexibility in the ways in which architectural components may be geographically distributed, within the constraints of maintaining the required defined interrelationships. This will allow varying degrees of bundling/unbundling, whilst maintaining multi-supplier interoperability.

Reuse of architectural components

5.7 The ICA should enable design reuse of architectural component specifications when new services or management capabilities are created.

5.8 The ICA should enable run-time reuse of architectural components so that the architectural components can be accessed in providing new services and management capabilities.

NOTE – This requirement is not satisfied by object-oriented analysis and design nor distributed processing alone: it requires the structured definitions of component libraries as part of the ICA.

5.9 The ICA should define means by which existing architectural components can be reused to build services and management capabilities.

Distributed execution

5.10 The ICA should not dictate the location of architectural components.

5.11 The ICA should facilitate transparent distributed computing. The ICA should allow architectural components to use all of distribution transparency defined in ODP-RM or a subset, according to the needs. This does not imply that every node needs to provide all the transparencies.[7]

5.12 The ICA should allow services to be provisioned and accessed from anywhere in the information communication system except that such access is subject to the security control implemented in that information communication system.

5.13 The ICA should be applicable to architectural components of customer domains, including low-end systems (such as PCs, set-top boxes, mobile terminals).

NOTE – This objective is also relevant for scalability (i.e. downsizing).

Support of services

5.14 The ICA should support but, be free from the limitations of, traditional call models (e.g. call-associated triggers, protocol-dependency) so that it can support new types of services, such as multimedia communications and information services.

5.15 The ICA should support (terminal, personal, and session) mobility services.

5.16 The ICA should facilitate cooperation between service and management capabilities in customer premises equipment/network and in other domains in the information communication system; this includes both network-based distributed services as well as end user services. For example, multimedia libraries, interactive games, in which the user can utilize the distributed processing capability of the information communication system.

5.17 The ICA should enable end user services to be tailored to meet different customer requirements.

5.18 Concurrent creation of service and management capabilities: The ICA should facilitate creation of end user services together with their associated management services.

---

[7] The ODP Reference Model identifies access, location, migration, federation, transaction, group, failure, resource, and replication transparency.

5.19 The ICA should include fail-safe mechanisms for handling unexpected service interactions. The fail-safe mechanisms will ensure predictable and stable system behaviour occurs across all architectural components.

Support for management

5.20 The ICA should contain no obvious impediments to the management of ICA components distributed across different domains and players.

5.21 The ICA should enable effective management of information communications systems in which architectural components are supplied by multiple vendors.

5.22 The ICA should enable, and make available, a collection of metering information from resources suitable for further processing by billing entities.

5.23 The ICA should provide means to deal with the availability and congestion aspects of the appropriate architectural components.

Security

5.24 The ICA should support authentication and authorization of entities involved in an interaction. The entities are, in general, stakeholders and they may reside in different administrative domains. Mutual identification and authorization should be also supported.

5.25 The ICA should make possible to collect and maintain audit information about actions performed by entities (defined as in above item). This is required to maintain accountability within the system conformant to the ICA.

5.26 The ICA should provide means for preventing stakeholders from performing actions that they are not entitled to perform. This is required to maintain integrity, confidentiality and availability of the system conformant to the ICA.

5.27 The ICA should enable monitoring of rogue and unusual activities, and adoption of countermeasures. This is required to maintain integrity, confidentiality and availability within the system conformant to the ICA.

5.28 The ICA should support protection of information (messages or stored data); this includes maintaining integrity of information and confidentiality within the system conformant to the ICA.

5.29 The ICA should enable realization of security controls across equipment provided by different vendors.

NOTE – Some networks are required by national regulation to provide certain security and safeguards features as built-in features.

User control

5.30 It is expected that future services and/or features offered by service providers will permit a degree of user control over such services. The ICA should allow the consumer to select from a variety of services for any instance of communication. Typically this may include some combination of:

a) Service type, e.g. voice service, video service, data service (IP service).

b) Session type, e.g. connection-oriented service, connectionless service, multicast service.

c) Quality of Service (QoS), e.g. throughput, delay, jitter, etc.

d) Additional encryption services.

e) Additional authentication services.

f) Mediation services, e.g. device adaptation, user presentation, etc.

g) Other appropriate services/features.

The main objective is to allow the user to specify the services required on a dynamic basis, which may vary from one communication instance to another.

NOTE – In some cases these features may be extra features over and above the built-in features of security and other safeguards features as required by national regulation.

Operator control

5.31 The ICA should enable the operators (service providers) to control:

a) policy aspects of service provision;

b) contractual aspects of service provision;

c) service availability and advertising aspects of service provision.

Scalability

5.32 The ICA should accommodate, and allow the evolution of, the scale of networks and service and management capability from very small to very large (of global scale) in terms of the number of users, the number of physical entities, the number of administrative domains, etc.

Mobility

5.33 Mobility shall be a key feature of ICA. In general all aspects of mobility are embraced by ICA, including both personal mobility and terminal mobility. For the majority of cases the ICA assumes the existence of a mobility-supporting system.

Compatibility with existing telecommunication systems

5.34 The ICA should accommodate interworking between existing and future infrastructural components and systems within the overall ICA umbrella. The ICA should provide means by which future systems within the ICA interwork with existing legacy systems within an ICA framework.

This includes:

i) enabling access from existing legacy systems to services deployed in future systems in the ICA;

ii) enabling access from future systems of the ICA to existing services.

NOTE – Accessibility from future systems in the ICA to existing services is required under the condition that such interworking does not give undue strong constraint to the development of future systems.

6 Basic concepts, assumptions and principles for ICA

The ICA guides the design of advanced information communications services and the systems supporting their provision. The principle of separating physical-connection-oriented call processing from service-oriented call processing is advanced by ICA middleware service components. The notion of calls is replaced by the notion of service instances and communication sessions. For example, this would be the case for connectionless services and cases where multiple calls and multiple streams are involved in a given session.

The ICA will be defined in a way that encompasses many different service architectures. In this regard, many different service architectures may be deployed to implement the ICA.

The basis of the ICA framework is that it provides for the clear separation of concerns. For example, service control from connectivity control and Signalling from transport network. Service control components provide the means to access, control, and manage services and connectivity. Interaction between these components and the infrastructure and Signalling network is provided through Middleware Programming Interfaces (MPIs) and Baseware Programming Interfaces (BPIs).

The remainder of this clause outlines some of the basic concepts, assumptions and principles underlying the design of the ICA.

6.1 Middleware services

ICA is the intermediary layer where independent middleware services intelligently bridge the gap between content applications and network transport services. This is illustrated in Figure 1. This intermediary layer will enable the transformation from a voice-dominated network to a data-dominated network.

The ICA architecture has been designed to meet the challenge of enabling multimedia communication across a diverse landscape of content providers, service providers and network providers. Multimedia was chosen because it is a ubiquitous element in the way people work, play, and conduct business. It also places high demands on network infrastructures to deliver quality-of-service guarantees.

The value of any architecture is best realized when capabilities are expressed in terms of functions that solve real needs or challenges. Challenges that are persistent over time are ideal candidates. The network challenge is to provide infrastructure services that seamlessly deliver multimedia information in a way that preserves natural human interactions across diverse set of data networking technologies and applications. In the context of this work, the definition of multimedia entails the technical aspects of using one or more media types (text, images, graphics, speech, audio, video, and data files) for the purpose of communicating across transmission, storage, access and content creation applications.

Seamless delivery of multimedia information between communicating parties (i.e. human-to-human and human-to-machine) encompass the following:

- user interface and information presentations;
- processing of multimedia information;
- organization, storing and retrieving of multimedia information;
- searching, browsing of multimedia documents and libraries;
- formatting, compression and coding of various media types;
- multiplexing and control coordination;
- end-to-end QoS and QoS Interworking;
- interactive voice and video communication.

Not all aspects of multimedia processing are part of the network. Some persistent capabilities, such as Graphical User Interface (GUI) presentation and processing of information, naturally reside in the domain of end user applications. Since ICA is positioned as network-based middleware, the capabilities to be supported directly reflect the location and movement of information across networking technologies. ICA capabilities encompass the transport or movement of information between end users and servers. At the most basic level these capabilities evolve around formatting and movement of information bits, and any necessary transformations thereof.

In the ICA middleware represents the service-processing layer. Service processing is carried out in an open distributed way:

- by structuring services into objects[8];
- by relying on a distributed processing platform for open interaction and cooperation among objects;
- by having a technology-independent view of the network resources, by way of a BPI which encapsulates implementation details and allows them to be seen by the Middleware as object interfaces;

[8] In this ITU-T Recommendation the word "object" is used in the sense of an abstract specification technique, and not in the sense of an implementation technique.

- by having a technology-independent support of applications, by way of an API which encapsulates deployment details and allows them to be seen by the Middleware as objects.

Within the context of the GII, the ICA:

- should focus upon a limited number of standardized generic APIs, MPIs, and BPIs allowing services built on these interfaces to be standardized, in an open-ended way that allows providers to compete with differentiated services;
- should be modular design framework with standardized interfaces between the modules providing the flexibility to satisfy a wide range of "enterprise role"[9] requirements, and a flexibility to allow service providers to adjust their interfaces as their requirements change but in a cost effective and efficient manner;
- should position the intelligence of the system at the highest level so that open interfaces are provided for service and network control;
- should separate areas of concern, i.e. applications from infrastructure by way of middleware;
- should support evolution from IP-based networks, ISDN/IN, GSM, CTM, etc.;
- should allow for the provision of high-level APIs to service providers so that they have a high level of network control in order to build differentiated services;
- should dictate concepts, rules, guidelines, prescriptive models for service creation;
- should give focus to the vertical interfaces (APIs, MPIs and BPIs) for the Middleware, which is an abstract execution environment for services;
- should ensure decoupling of Transport technology from specific services.

The conceptual aspects of the ICA include generic design concepts like middleware components, service instances and communication sessions. Conceptual support is meant to enable service and system designers to follow the key design principles. Conformance to these principles should result in good and flexible design of open-ended systems offering a multiplicity of (customized) quality services. Such systems also exhibit service flexibility allowing public and proprietary services to be integrated and combined in a modular and flexible way.

Other key design principles of the ICA include:

- Coherent design of management and control.
- Customization and personalization of services.
- Nested services for defining and providing new services.
- Separation of application and session as well as resource and communication-oriented problems.
- Separation of media services from their control and management.
- Separation of service access from service core.
- System-independent modelling of services.
- Reference Points at which it may be desirable to have open interfaces.

The application of this collection of design principles and concepts is expected to be adequate to support modelling of services, because of their intrinsic generality.

[9] Business roles as perceived by applying the ODP-RM include: Content Provider, Service Provider (Brokerage), Customer (User), and Transport Network Provider, etc. A role is a set of activities carried out by a stakeholder. A stakeholder can play different roles. A stakeholder is an individual or organization that partakes in some activity related to a business.

6.2 Relationship to GII Recommendation Y.110

The ICA's focus on middleware is as defined in ITU-T Recommendation Y.110, "*Global Information Infrastructure principles and framework architecture*". The primary area of concern of ICA is the Middleware functions as shown in Figure 3.

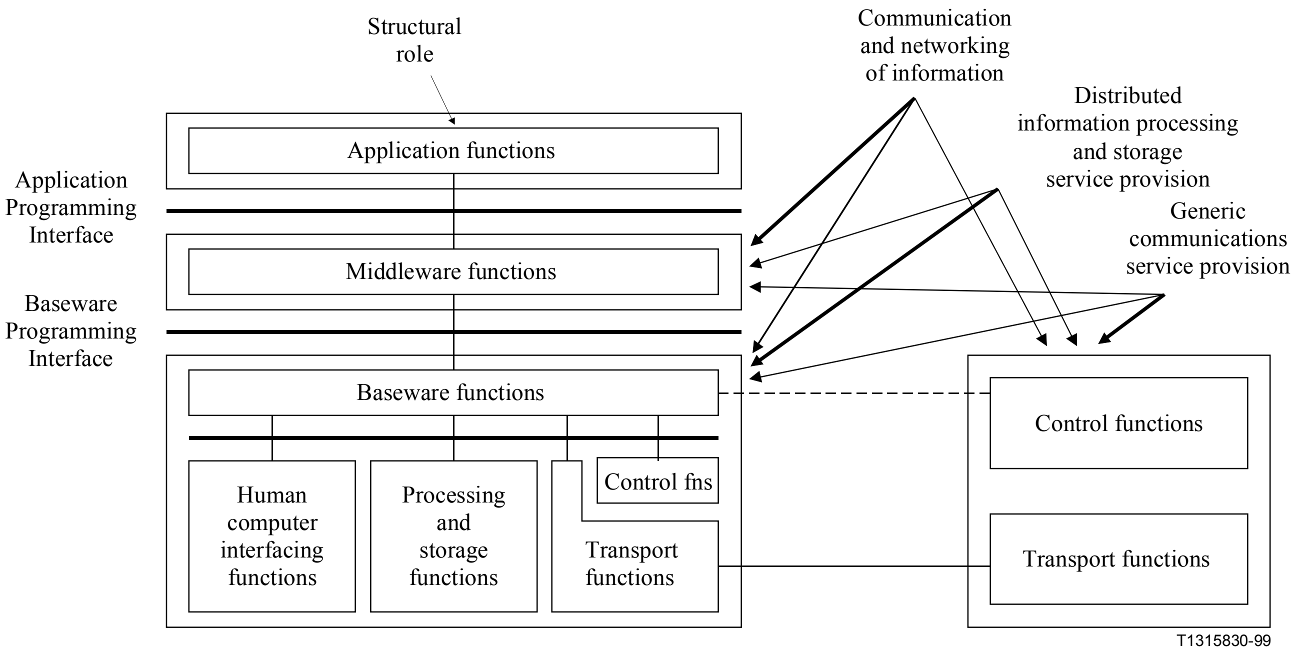

Figure 3/Y.130 – Area of concern (extracted from ITU-T Recommendation Y.110)

ITU-T Recommendation Y.110 should be consulted for further information on this subject and the related general issues of players, role, domains, etc.

NOTE – In some cases, ITU-T Recommendation Y.110 for example, the Baseware Programming Interface shown above has been termed the Basic Programming Interface.

6.3 The Agent concept

One of the biggest changes taking place in the telecommunications industry is the choice that the user has in selecting and engaging various communication services.

Alternatives now exist for the user in terms of:

a) the type of service that can be provided;

b) choice of infrastructure technologies available to provide the various services;

c) the choice and/or options available for satisfying/providing the required service in terms of quality and cost.

It is envisaged that many services and technologies will be made available to meet customer requirements. The "algorithm" required to make optimum choices, within the limits of the set of available services and technologies, can be extremely complicated. In many cases this is too great a burden for the user to bear in a rapidly changing environment. Users in general are not too concerned about the underlying technology to be used. They are more concerned about the type of service, the Quality of Service (QoS) and the cost. It is the responsibility of agents, operating as middleware services, to simplify the process of selecting and engaging communication services.

For the purpose of this ITU-T Recommendation the following definition of an agent is used. An agent is an element that performs some task on behalf of some party (i.e. user, machine, application, or another agent), rather than have the party, itself, perform the task. The term "party" refers to either a client or server application involved in communicating with others.

The concept of agents appears in many existing architectures, such as the User Agents in TINA and the User Agents in the Message Handling System (MHS) defined in the X-series Recommendations.

6.4 The Problem Space and Driving Forces for Agent operations

To provide any instance of communications service, the following information must be derived:

a) The parties to be involved in the communication.

NOTE 1 – Generally, deriving the knowledge needed for any communication may include identification techniques, actual identities, location information, and customer service policies for all parties involved.

b) The nature of the communication, i.e. the session characteristics, in order to control and manage it.

NOTE 2 – Generally, this requires knowledge of characteristics of the service requested, and the devices (information appliances) and thus, how to control and manage the communication session.

c) What transport technologies to use for the communication.

NOTE 3 – Generally, the type of technology to be utilized for a particular instance of a communication session depends on the service requested, any associated specific Quality of Service (QoS) requirements, and considerations of available resources and applicable costs.

Consequently, the ICA architecture is predicated upon the following three key functions:

d) **Party Identification** (including information location and profile identification).

e) **Session Management** (including device management).

f) **Transport Selection** (i.e. choice of technology and interoperability among multivendor Transport technologies and providers).

The approach taken to providing these three functions is the use of the appropriate arrangement of agents within the Information Communication Architecture (ICA).

In particular, the following agents are proposed in respect of the identified problem space:

A **Contact Agent** – for Party Identification.

An **Exchange Agent** – for Session Management.

A **Transport Agent** – for Transport Selection.

These concepts are illustrated in Figure 4.

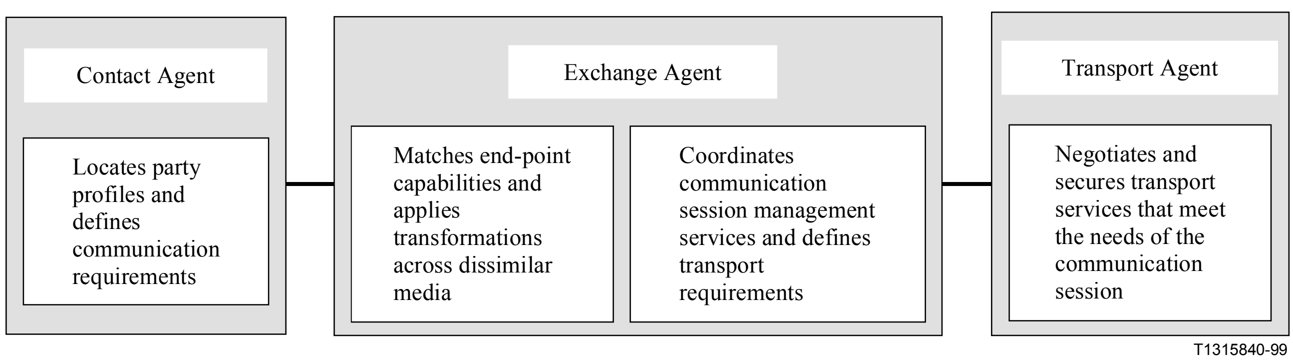

Figure 4/Y.130 – Basic concepts of ICA

There are a number of forces driving communication infrastructures, these include:

- networking solutions need to take into account the diversity of powerful edge devices and content services;
- expanding IP-based information infrastructures;
- evolution of intelligence in communication networks;
- increasing availability of bandwidth on demand across a variety of transport systems;
- availability of ubiquitous access to the Network via LANs, wireline, and wireless networks;
- ever-increasing amount of low-cost memory and computation power;
- proliferation of appliances that can handle a wide range of media;
- digitization of virtually all aspect of communication;
- growing number of providers offering differentiated services;
- explosion in number of multimedia service models.

6.5 Mobility

The ICA will take a two-tiered approach to the provision of mobility services.

Tier 1 mobility will be device oriented, and as such ICA will exploit any mobility system(s) inherent in the baseware subsystems, such as those associated with the access network for example. In some cases this will relieve the Middleware, itself, from having to deal directly with mobility mechanisms. This approach is sometimes referred to as the "bottom up" approach.

Thus, in some cases, the intelligence for the support of mobile communication services is considered as part of the baseware. In this sense the mobile systems, for example GSM, and IMT-2000, are assumed to provide the functions needed in support of mobility; the Middleware will include an interface to these systems as it does to other basic information communication services.

In accordance with the above assumptions, and considering that mobility functions are already provided in the baseware that includes third-generation mobile systems, the main impact on the ICA is that the Middleware has to deal with the interface provided by the underlying mobile system and has to enhance its quality. In other words, the Middleware only has indirect control over mobile system functions. In particular, the Middleware only has control over certain parameters of the information communication service provided at the interface between the Middleware and the mobile system (e.g. Quality of Service, cost of service, etc.).

Tier 2 mobility will be based on ICA's middleware capabilities. This will permit a richer set of mobility services to be provided, based say, on application specific and/or user personal mobility rather than device mobility. This will be required in cases where the capabilities of tier 1 are not sufficient in scope or extent to meet user requirements. This approach is sometimes referred to as the "top down" approach.

In Tier 2 mobility, middleware intelligence can be used to locate individuals whose nomadicity spans multiple networks, multiple addresses, multiple devices, and frequent time-related geographical moves and changes. Inference engines and heuristic techniques within middleware can be used to track, derive, and if necessary, actively search for an individual's location at the time(s) required.

Tier 2 middleware will construct a data profile (i.e. device type, voice/text/video media and security preference) containing an individual's communication requirements at any given time.

Further background on Global Mobility, consideration of bottom-up and top-down approaches, and a review of the ETSI Global Mobility Framework, is given in Annex D.

6.6 Relationship of Agents to layering

From ITU-T Recommendation Y.110, it has already been established that the GII has three layers. ICA is primarily concerned with the middleware layer. Within the middleware layer shown in Figure 5, various agents can be employed within the architecture to achieve dynamic instantiation of appropriate communication session components per service instance.

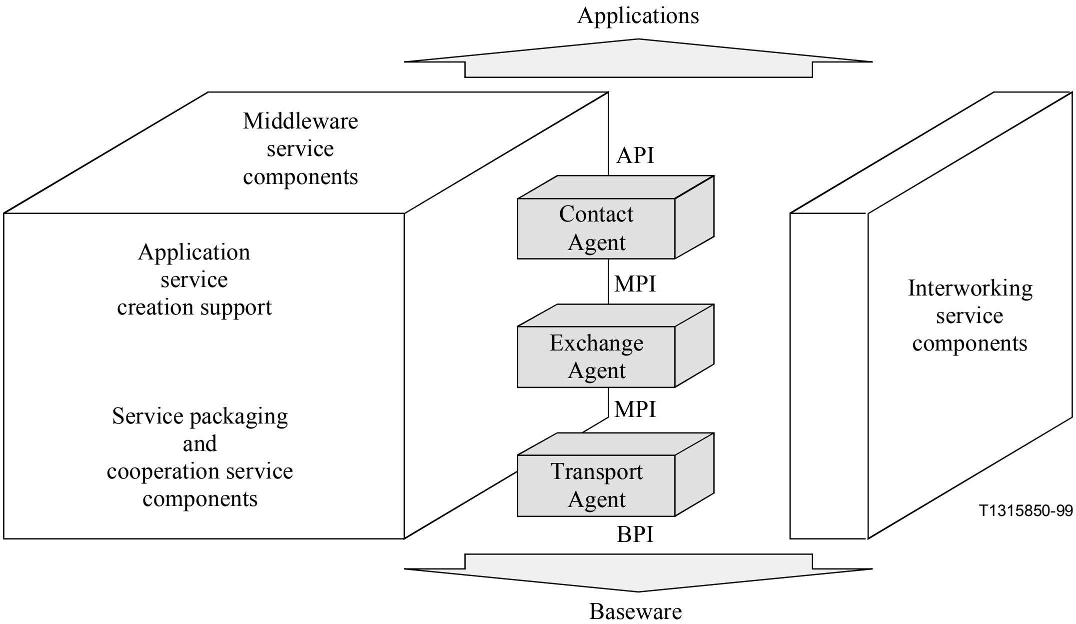

Figure 5/Y.130 – Relationship of Agents to GII Middleware service components

This arrangement of middleware service components enables:

a) Application Functions to communicate via an Application Programming Interface.

b) An Application Programming Interface to make use of Middleware.

c) Distributed Middleware Functional Entities to cooperate with other Middleware Functional Entities.

d) Middleware to communicate via a Baseware Programming Interface.

e) Baseware Programming Interface to make use of Baseware.

As shown in Figure 5 the specific use of agents augments the middleware layer. The role of agents is to enable parties to communicate easily and effectively across diverse network domains. In combination, the agents serve to combine infrastructure capabilities into a communication session that meets the goals defined by the parties. The term "agent" also serves to emphasize that the "delegation" serves to insulate each layer from the details that lie beneath it. This separation of concerns ensures that the architecture meets many of the key design principles of ICA.

6.7 Generic protocol model

Generally speaking, services are provided by the instantiation of protocol "stack", i.e. several layers of protocols. There are two aspects to the use of protocols:

a) the vertical aspects; and,

b) the horizontal aspects.

6.7.1 The vertical protocol aspects

Figure 6 shows a layered protocol stack. In this figure the end of the communications/network-oriented protocol stack and the beginning of the application-oriented stack is denoted by the datum at layer n. So, for example, in an Internet context, layer n would be the IP layer, and n+1 would be the TCP or UDP layer. A variety of underlying layers, n−1 through n–i, will exist depending on the underlying technologies used to support IP (e.g. FR over ATM over SDH, or whatever).

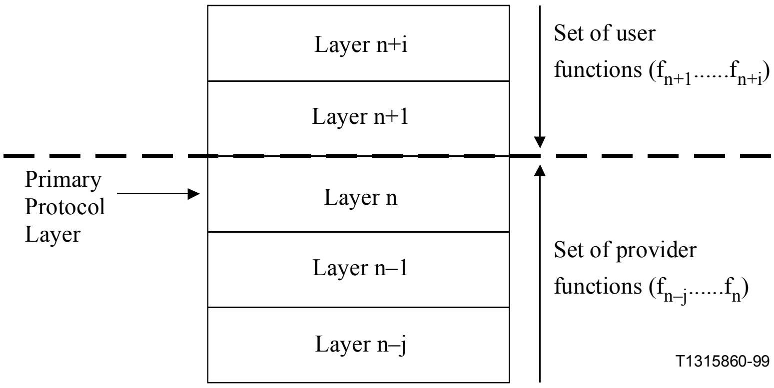

Figure 6/Y.130 – Generic layered protocol stack architecture

Each layer above and below will have its own set of functions. As a general principle it is desirable not to duplicate functions in each layer. Elimination of such duplication is not always possible or desirable. Addressing and flow control at different layers of a protocol stack can fall into the category of necessary duplication in some cases. The total functionality provided to Layer n+i is the sum of functions by all of the underlying layers.

For universal communication it is desirable that the protocol in layer n be a single unique protocol and be universal also. However, this is not necessarily practical in an evolving environment, and over the years a number of "universal" protocols have been defined and deployed, such as X.25, IP, etc. which are universal but only in their own discrete universes. For our purposes we shall call the N-layer protocol the "primary protocol layer".

Layer n+1 is the layer that provides the user's service requirements and associated operational parameters. It is this information that leads to the choice of the protocol for layer n and its suitable operational policies.

Layer n is the layer of concern to both the service user and provider, and is termed the primary protocol layer. This layer provides a protocol bridge between service instances and transport sessions in the form of communication sessions. The choice of protocol is made dynamically and persists for the entire life cycle of the service instance.

Layer n+1 is the topmost layer of concern to the transport service user and provider. Layer n+1 supports the application-to-application communications and is not interpreted by the transport service provider.

The layers below primary protocol layer are concerned with end-to-end transport functions associated with communication sessions and are not interpreted by application users.

6.7.2 Horizontal aspects

ITU-T Recommendation Y.110 defines the relationship between domains, segments and service support platform. Figure 7 shows an example of the use of the ITU-T Recommendation Y.110 components and concepts as applied to ICA. In this figure the lines that join circles together represent an interface at which peer entities come together. This is sometimes referred as a "peering" point.

A peering point interface is considered to be a point at which a service is offered. In a complete system many peering points will exist which represent the services provided at each point. The protocol stacks are likely to differ from one peering point to another. The differences will result from intermediate technology transformations and value-added operations that occur between one peering point and another. Peering points facilitate multivendor and multi-organization interoperability as well as providing an interface to the end user. From the end user's point of view a peering point represents the boundary point of the network to which he/she is connected.

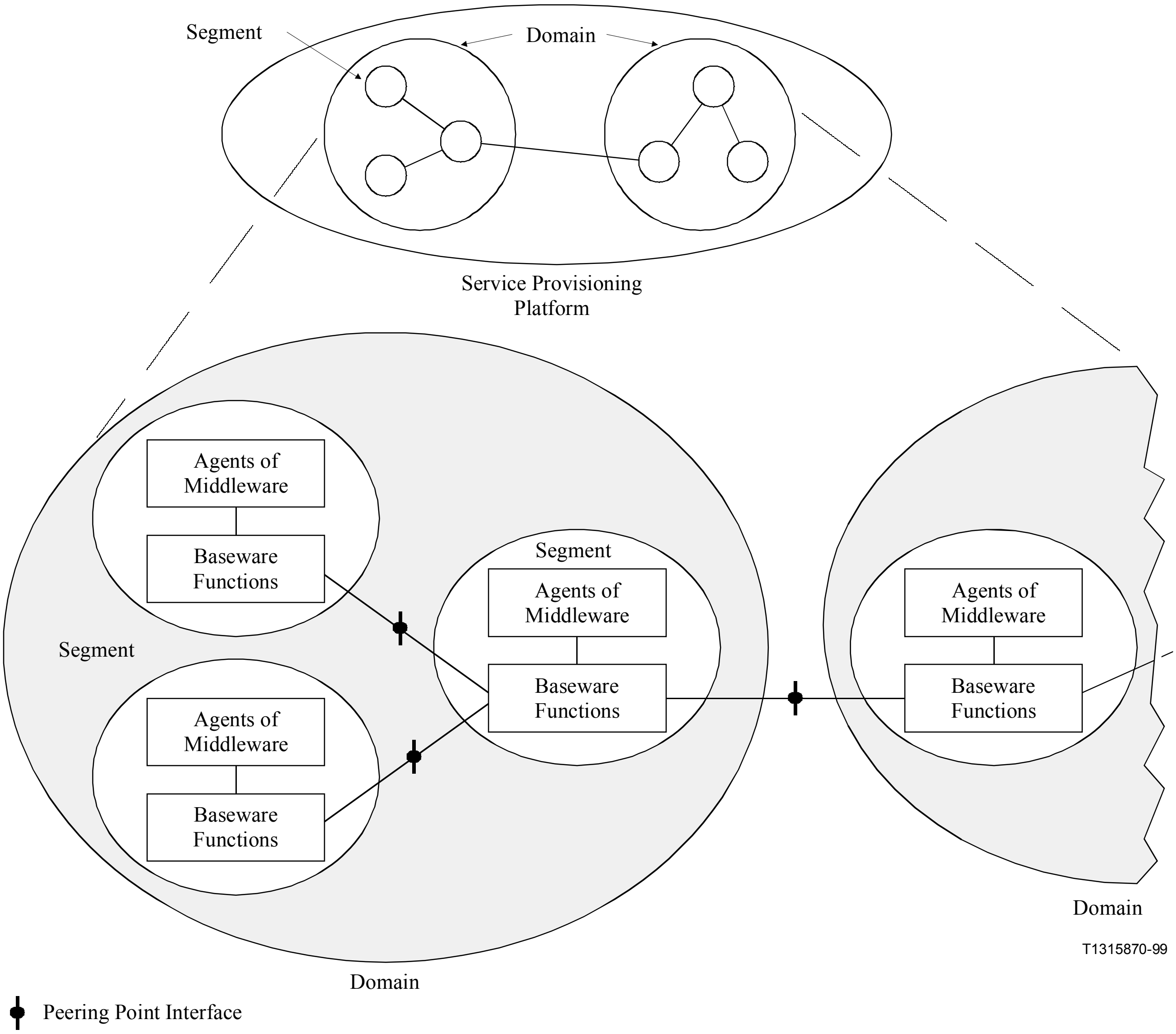

Figure 7/Y.130 – Example Peering Point relationship of Agents to Y.110 elements

The middleware agents shown in Figure 7 are the three agents shown in Figure 5, namely the contact, exchange and transport agents.

6.7.3 Intelligent network boundary

From the above vertical and horizontal models it can be seen the conjunction of a horizontal interface with a vertical protocol stack forms a significant boundary between peer entities.

The concept under consideration is both an organizational and a technical one. ICA identifies separation points that reflect boundaries of ownership as well as boundaries of unique technology capabilities.

Generally, a separation point can provide two functions:

a) the brokerage function to negotiate connectivity; and

b) the transformation function to adapt between dissimilar systems.

The broker function is used for initial contact to simplify selection and administration of communication services. Brokerage offers parties a choice of communication capabilities at the network edge based on how the party requires the information to be moved in the network and does not force the party to use a specific transport protocol. This function helps integrate networks and their services at network boundaries.

The transformation function allows dissimilar user devices and applications to be accommodated in order to ease service deployment. Currently, applications, devices, or other networks that want to access a network must adapt to the target network. With ICA, parties may use embedded transformation services offered at the network edge to acquire the necessary adaptations. Transformation services are used to modify the information network entry, before it is transported, and on exit, after it is transported, if necessary. Transformation services allow the network to accommodate a greater number of content services, user devices, as well as provide more options for users.

From the user's point of view, the peering interface at the network boundary is of particular interest. The layered approach, together with the user/network horizontal interface, in conjunction with brokerage and transformation functions defines a universal network edge. This can offer a wide selection of transport services, making choice simpler and enabling service customization. Communication networks should satisfy communicating parties' needs by providing communication services seamlessly. To accommodate diversity, to enable service automation, and to make communication access easier and choice simpler, the network needs the above-mentioned edge components.

7 The basic architecture

As an architecture, ICA is a functional decomposition of network control systems into a set of agent-based middleware service components. Each ICA agent provides services that when combined within a collaborative framework creates a universal communication control system. As individual services, each agent is constructed out of a private set of functional components. These components are essential to the successful operation of each agent and are illustrated in Figure 8.

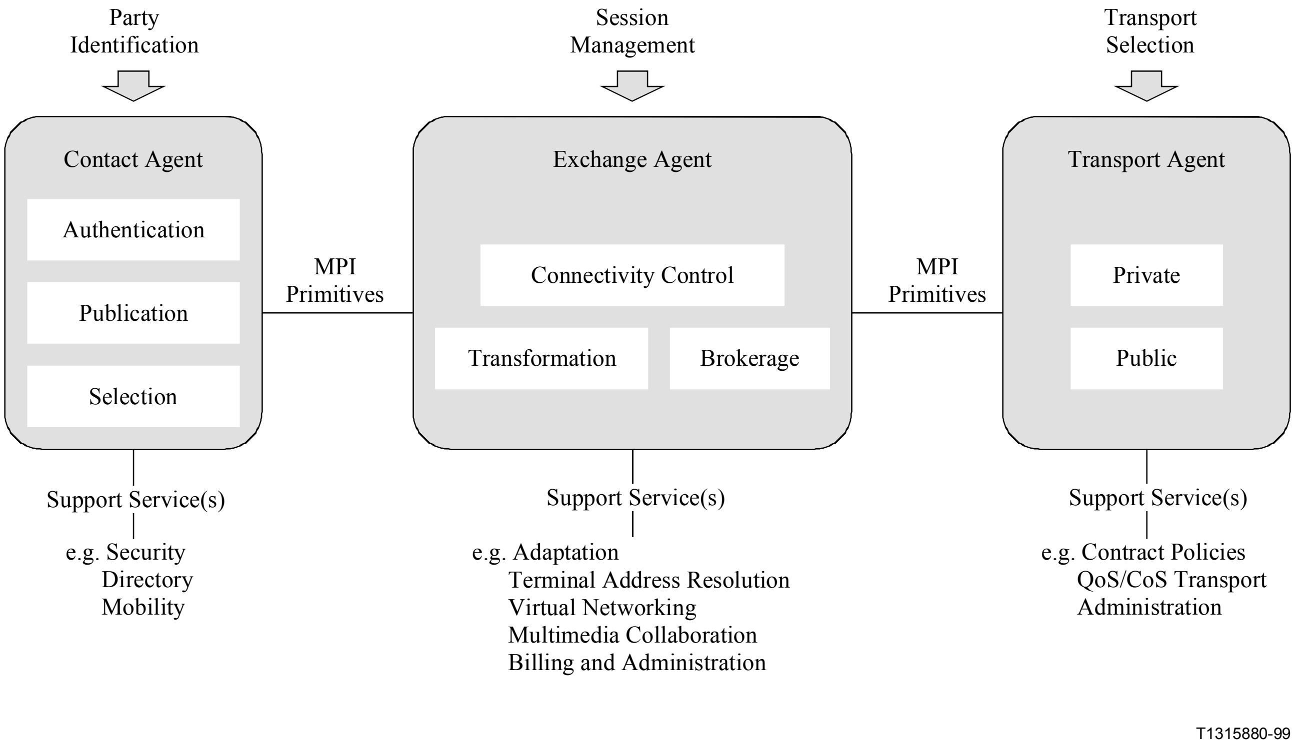

Figure 8/Y.130 – Conceptual architecture

Services residing in the ICA layer will reflect how a network provider has decided to package network services to meet selected market requirements. One of the defining characteristics of an agent is the ability to support flexible interaction and interoperation with other similar software entities. It is expected that the ICA agent framework provides an environment for other specialized applications to be constructed. There are many applications, such as E-commerce, virtual networking and real-time multimedia communications, that continue to challenge the design and performance of networks.

In the industry, agent-based architectures are working towards standard methods and protocols for next generation knowledge-based systems, which will impact how ICA agents are constructed. From an architectural perspective it does not matter how ICA agents are implemented; what matters is what they do and what they are connected to. Engineering and economic considerations will drive agent implementation.

In order to build an agent system that can span the varied requirements of multimedia communication, a functional structure was developed. A system perspective behind how architectural underpinnings of network middleware services operate is presented in Figure 9. Shown in this figure is a generic view of cooperating agents, three agents cooperating within ICA. In practice these agents can be combined, replicated, or divided into a number of further sub-entities, to achieve specific geographical distributions of middleware service components that might be encountered in real-world cases. Users interact with these agents according to the context defined by a session instance established between communicating parties.

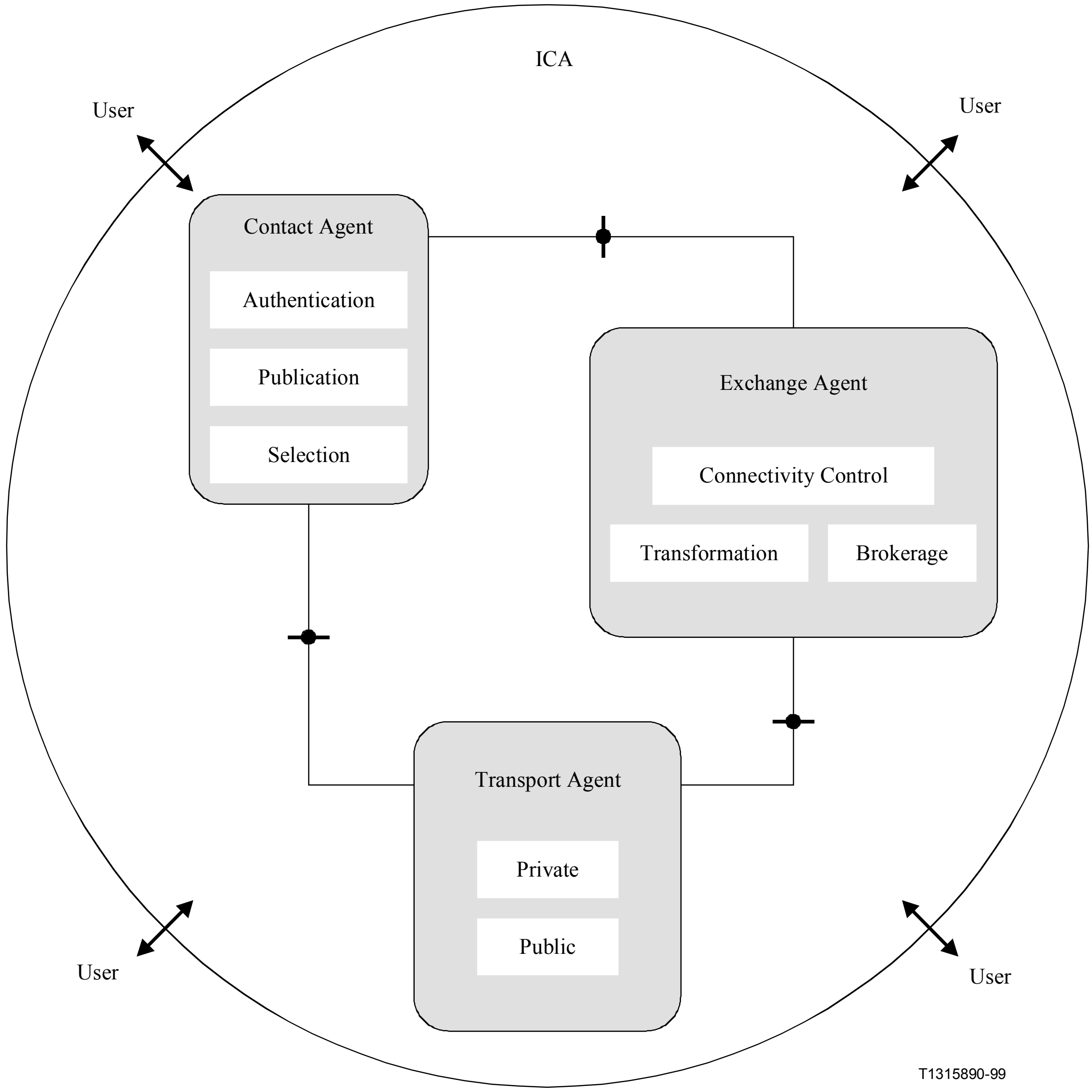

Figure 9/Y.130 – Generic functional structure

This structure illustrates the key relationships between agent-based functional components and their interface points. The black dots represent reference points that will be defined for the ICA. It may be appropriate to specify open interfaces at these points.

This structure is not a complete representation of the architecture, but it does identify major components of the system and their interactions with each other. In general a user, in the form of a party, may interact with any ICA agent provided the appropriate interface is used. In practice this flexibility is not viewed as being desirable. For if a party interfaces directly with the exchange or transport agent the party is no longer decoupled from the operations of infrastructure components and must perform additional management tasks. In the ICA model, an external party presenting their communication requirements to the contact agent starts the whole process.

It is expected that details of the communications between the component parts of ICA will be defined in the form of Service Primitives as defined in ITU-T Recommendation X.210. For the purposes of this ITU-T Recommendation the following X.210 definition of service primitive will be used:

Service Primitive: An abstract, implementation-independent interaction between a service user and the service provider.

For the ICA architecture to be universal and persistent, much like computer middleware, functional composition needs to reflect both network engineering and business considerations. This means achieving a balance between separation of the technical aspects of networking and business aspects of product packaging. In pure architectural terms, this involves determining the right level of granularity, as well as an overall structure that has both desirable engineering and business values. The contact, exchange and transport agents provide the right level of balance.

Within each of the agents reside one or more functional components that define how an agent is constructed. It is the components that form the foundation for middleware product offerings or services. This enables the adoption of technologies as they mature for use in the communication industry.

A description of each agent is presented in the following clauses.

8 Contact Agent

The Contact Agent is responsible for the identification of party communication requirements including the selection of devices to be used during a service instance. Today, communications infrastructures are geared around communication between devices, not parties. People use phone numbers which identify a telephone line, not a person, as the destination of a call. Internet e-mail addresses specify individuals at a specific device. Web surfing specifies pages on specific machines. However, the intention of the parties is generally not to communicate with the device they specify, but to communicate with the party with whom they associate the device. If parties identified the other parties, not devices, with whom they wished to communicate many new opportunities would arise.

The Contact Agent is a middleware service component which, given an originating party, and a mode of communication determines the devices that are capable of achieving this goal. The mode of communication is most likely to be an expression of a business offering (i.e. service) rather than a technological expression, but in the end it should resolve down to a human centric expression of communication, such as talk, read, watch, send mail. Focusing the expression of the service on the human centric form enables the widest possible range of device choices by the network, rather than expressing it using technological terms, such as "phone". However, the exact nature of the communication primitive is not a cornerstone of the Contact Agent's value, as long as the mode can be agreed to by communication consumers and providers alike.

The role of the Contact Agent is to handle all aspects of defining party communication requirements and capabilities. In the ICA, party identifiers are decoupled from devices and parties will specify their multimedia communication requirements. Key to the successful operation of the Contact Agent is the ability to access directory information based on personal identifiers. Parties are known by their personal identifiers which are based on any of the established e-mail, DN, URL standards. In the future, party names can follow unified name standards as they evolve.

Using the party identifier, the contact agent determines the host directory (e.g. DNS or SCP) that holds the target data object or party profile. Requirements for the session are specified either by a party in real-time, by the application the party is using, or as previously entered datafill. Communication requirements are statements of Quality of Service needs in the context of the application and user preferences for the duration of the service instance. Requirements are used to create a communication profile specific to the life cycle of the service instance. The communication profile would be dynamically created as a composite of gathered information relating to user preferences, service preferences/requirements and available network capabilities.

The communication profile also contains information regarding the formatting and movement of information bits for the multimedia session. The formatting information specifies media type and/or device requirements while movement information specifies transport behaviour preferences. By using different personal identifiers or profile attributes a party can define different multimedia behaviours that reflects the many roles of a party (i.e. working role, recreational role, or family role). This enables ICA to support the diversity and changing multimedia requirements of parties.

Communication requirements are converted by the contact agent into a communication primitive. Other agents or vendor products use primitives to invoke standard or propriety services. Depending on how the industry evolves session requirements (world of applications) and communication primitives (world of networks) may one day standardize on the same interface strategy.

In a complete ICA system, the communication primitive would be passed on to the Exchange Agent who would provide the session services required to enable this communication. This is illustrated in Figure 10. However, the Contact Agent's behaviour can be exploited in a system without the other ICA agents, where the existing legacy infrastructure either does not require or enable those components. For example, in today's Internet, DHCP, DNS and LDAP systems provide a limited version of the Contact Agent's behaviour, although today's configurations do not provide the rich capabilities that a complete instantiation of the Contact Agent's behaviour could provide.

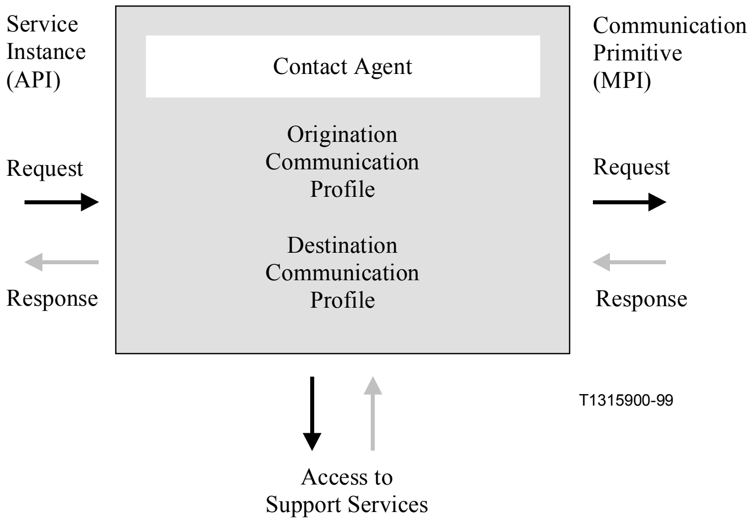

Figure 10/Y.130 – Contact Agent interfaces

In addition to constructing a profile for the origination party, the contact agent also locates and constructs a profile for the destination party(s). The destination party communication profile is modified to reflect the communication primitive issued by the origination party. How profiles are processed is defined by service policies. Both the origination and destination profiles are forwarded to the exchange agent via communication primitives.

In order to perform Contact Agent tasks access to support services may be required. Based on user goals and profile information, the Contact Agent can issue requests that trigger other support services. Examples of other support services include:

- Security Services;
- Directory Services;
- Mobility Services.

The Contact Agent is based on a Publish and Select model. Publication involves registering with the Contact Agent the devices that a party wishes to use for a specific mode of communication. The Contact Agent is made up of three major components:

a) Authentication Engine[10];

b) Publication Engine;

c) Selection Engine.

All of these engines rely on the storage and retrieval of information from databases.[11]

8.1 Authentication Engine

To provide trusted communication between parties, there must be a mechanism for ensuring that the parties are who they say they are when they engage the network. From a network operations perspective, identity of the party is equally important as it will be the principal mechanism for charging for network services.

As an example of a possible authentication approach, one could have every party equipped with a security certificate. The certificate includes the party's private key, which is used to provide secure authentication via digital signatures. Other keys could also be stored in the repository, perhaps representing the party's different roles, or to be used for encryption. The certificate should also provide the address of the hosting Contact Agent, and the identifier by which the Contact Agent knows this party. The party must engage with the Authentication Engine to exchange digital signatures and establish a trusted communications path before the Publication Engine or Selection Engine will provide any service.

8.2 Publication Engine

Given that communication is between parties, but is achieved by using devices, an association between the party and the device or devices a party wishes to use must be made. There exist many forms of this in the telephony world ranging from call forwarding to cell phone roaming. Essentially, when a party wishes to register his use of a device, he updates the device association rules using the Publication Engine. For very simple associations, these rules may simply be a key and value pair in a directory. However, more complex rule-based associations may require a more complex Selection Engine, complete with the capability to dynamically create associations based on rules encapsulated within program logic, and as such the Publication Engine may need to be more complex as well.

Some examples of the capabilities provided by a Publication Engine could be:

- rules that specify sequences of devices, allowing the Selection Engine to deal with situations where the primary device is not available or not responding;
- profiles which encapsulate a number of rules that specify Selection Engine behaviours (e.g. call forwarding and filtering);
- rules that define who pays for the communication (i.e. originator, destination, third party);
- party tracking, which can be used to switch automatically between profiles based on time of day or physical location. Tracking can be based on geographic information determined via mechanisms used to locate wireless devices, or the current location of a wireline device.

Once the party has authenticated himself to the Authentication Engine, it is a matter of informing the Publication Engine of which rules are to be engaged or changed. How this is done will be based on what rule mechanisms are defined.

---

[10] See Glossary of terms in Annex F for definition of "engine".

[11] Scope of the Contact Agent can be expanded to include aspects of GII processing and storage baseware functions.

8.3 Selection Engine

The Selection Engine performs two tasks: locating the hosting Contact Agents for all the destination parties specified by the originating party, and executing the rules provisioned by the Publication Engine to find devices based on the party's communication request.

The Selection Engine will reference a set of directories looking for the addresses of hosting Contact Agents. These directories will be pointed to by the originating party's profile, and may or may not include the party's private directories, directories operated by Contact Agents, or directories provided by third parties. The search may or may not be recursive through directories recommended by other directory providers.

Once the Selection Engine locates all the parties, it sends each of the hosting contact agents a request with all the parties that will be included and the type of communication requested. Hosting Contact Agents consult their own directories, apply appropriate policy rules, and return the list of devices that are suitable. While waiting, the originating Contact Agent does the same for the requesting party, giving preference to the device being used to make the request. Once the hosting Contact Agents return their information, the originating Contact Agent can pass the complete list of devices to either the Exchange Agent, in a complete ICA system, or back to the originating party.

9 Exchange Agent

9.1 General scope and functions of the Exchange Agent

The Exchange Agent is responsible for providing infrastructure services needed to enable successful communication between parties. The Exchange Agent provides communication session management services in order to hide the transport network technology details from users.

It is necessary to provide an understanding of how the term "session" is used in ICA. There are several different types of sessions, each with different capabilities, each at different layers in the network. A clear separation of these sessions must be maintained if one is to breakdown the management of a "call" into various manageable pieces. The term "session" may be used to cover three aspects:

1) Application session that exists between end parties or applications for the duration of a service instance. This session manages Signalling between end parties.
2) Communication session that exists between application and transport layers. This session manages infrastructure resources and connectivity controls for data streams contained in any given application session.
3) Transport session that exists within and between different transport domains. This session manages the physical resources of a particular transport system.

The three aspects are illustrated in Figure 11. Each transport domain may comprise any particular transport technology, e.g. wire, fibre, wireless (including satellite), etc.

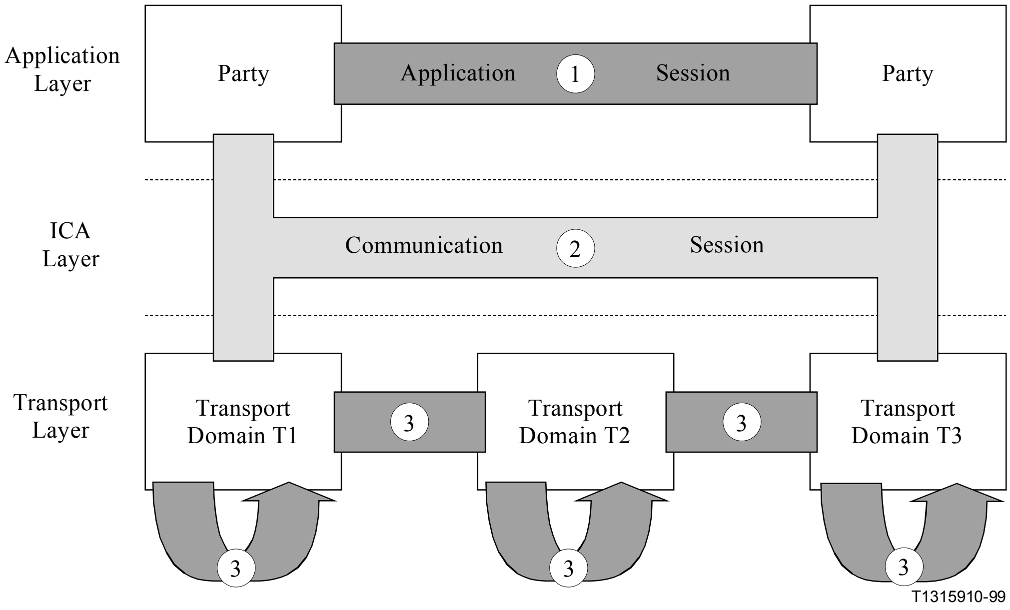

Figure 11/Y.130 – Separation of session into three aspects

The communication session, supported by the Exchange Agent, is intentionally independent of end-user application, and is intended to be a pure middleware service. A communication session must operate across a diverse range of transport systems and be used in conjunction with one or more user applications running either in the user's terminal or in a network element, or both. Potential users of the Exchange Agent communication session services are any applications that require connections from a transport network. In this view, network aspects of session management are middleware services provided by ICA and used by other applications to fulfil their connection management requirements.

The ICA system has been designed with the goal of connecting parties together, not devices. This means that the Exchange Agent uses the multimedia requirements stated in the party profiles to synchronize the events between party devices and transport services. The Exchange Agent contains networking intelligence, control logic and policy algorithms, capable of responding to the communication primitive supplied by the contact agent as shown in Figure 12.

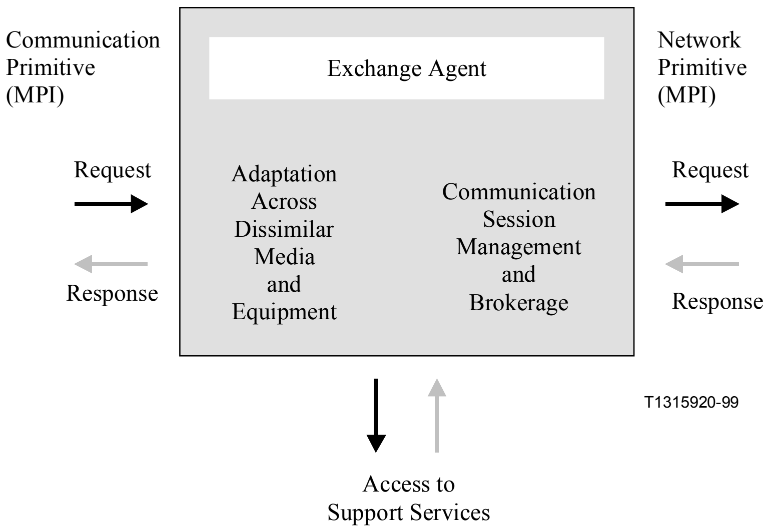

Figure 12/Y.130 – Exchange Agent

It is the exchange agent that specifies how information bits are to be transported. A specific transport primitive is defined for the multimedia session based on resolution of interworking policies across devices and session types. A transport primitive is similar in nature to the communication primitives used to by parties to request communication services. The network primitive articulates delay, throughput, reliability, and connectivity requirements in a generic manner such that negotiation can take place with any network provider offering transport services. The transport primitive provides the flexibility to interwork with any vendor technology and adapt to multimedia processing changes. When required, the exchange agent acts as a proxy for the communicating parties in the process of negotiating or brokering end-to-end transport services.

First there is the selection of suitable devices. Using algorithms designed to recognize text, voice, and video standards, devices are selected from the party profiles that best meet the communication goals defined by both the origination and destination parties. This determines endpoint devices involved in the communication session. When no match can be found, then suitable transformations are identified and applied. Then there is the identification of network routing addresses associated with the selected devices. This may involve mapping virtual names (e.g. e-mail, URL and 1-800) to physical address suitable to the transport technology being utilized.

This is followed by the activation of session controls that are consistent with the device capabilities and overall party requirements reflected in the communication primitive. Communication session management can vary from thin TCP control for message-based flows, to complete call control for circuit-based streams. The exchange agent possesses control policies for management of multimedia data flows. Policies cover a wide range of quality-of-service and connectivity[12] alternatives. A key part of communication session is the ability to adapt between different media, device types, and control protocols.

In order to perform Exchange Agent tasks access to support services may be required. The Exchange agent can issue requests that trigger other support services. Examples of other support services include:

- Security Services;
- Directory Services;
- Adaptation Services;
- Terminal Address Resolution Services;
- Virtual Networking Services;
- Multimedia Collaborations Services;
- Billing and Administration.

The Exchange Agent is made up of three main components:

a) the Connectivity Engine;

b) the Transformation Engine;

c) the Brokerage Engine.

9.2 Connectivity Engine

Applications running in the end systems will request the network to set up a communication session. The connectivity engine is responsible for determining the end-point address that can be used to route traffic. In general no prior assumption is made to the type of devices either the sender or receiver will be using for a particular service instance. The communication profile assembled by the Contact Agent identifies the target devices that the user wishes to use. The device specified in the communication profile may be defined in terms of device type label or device address. When the

[12] Connectivity refers to aspects such as multicast, bridging, uni and bidirectional transport.

device is referenced by an address queries are made by the connectivity engine to determine where in the network the device is currently located and a valid network routing address. This process will more than likely involve interworking with network-based location management databases. When a device label is used, a database query will be required to translate a device type label to standard device name that has been defined by the industry.

The connectivity engine is responsible for determining the network routing addresses that are contained in the network primitive.

The connectivity engine must provide a protocol by which the end systems can request the desired configuration (end point addresses, channel arrangements at each end point, security characteristics of the channels, etc.), and which allows the topology to be modified. In addition, administration policies (admittance, billing, etc.) reflecting the network operator requirements need to be applied. Once the communication session is set up, the applications at the end points transmit data or messages between themselves. In some cases, processing of the data stream in the network is required (e.g. echo cancellation, media transformations), and in other cases no processing or knowledge of any kind is required by the network (as is the case with IP data over dialled up voice connections today).

In order for the network to adequately support services that entail multiple participants, each exchanging multiple data streams on a single call or application invocation, it will be necessary for the network to support a connection topology more general than the point-to-point single channel connection. The term "communication session" will be used to mean an n-way multichannel connection set. It is important to recognize that, unlike the point-to-point single channel connection, the session is defined in such a way that a single session could be instantiated in any one of a variety of connectivity arrangements (i.e. connectionless to connection orientated). The communication session then is a generalized connection concept, capable of representing an unlimited variety of connection topologies. As such, the simple point-to-point IP packet stream can be realized as a session over any transport technology.

The Exchange Agent will control of the overall connectivity associated with a service instance. It decides the device end points and adds in transformation services to provide an overall adaptation service. Since it has knowledge of the overall communication session, it might be best if this agent also collects metering information and passes it on to downstream billing systems.

9.3 Transformation Engine

The transformation engine is used to convert one media type to another. Transformation generally falls into two categories, those that are required in real-time, and those that are non-real-time. Real-time transformations refer to adaptations performed on live data streams such as telephone calls or video feeds. Non-real-time transformations refer to adaptations performed on pre-generated multimedia such as static images, documents and recorded speech.

The multichannel connection approach allows any number of participants to be accommodated in a communication session. Each participant can have any number of individual data streams. Moreover, it is not required that each participant have the same data streams transmitting and receiving at their terminals as the others. For example, on a combined voice and video call, some participants may not be capable of transmitting/receiving video, and hence they participate only in the voice portion of the call. Transformation services expand the possibilities of multimedia interaction. For example, some participants may want a voice call to be converted to a text message or request online language translation services. The combination of transformation engine working in concert with a connectivity engine enables true multimedia communication between parties.

The transformation engine uses data about each parties device capabilities to determine a best match. In general, it is expected that an exact match (e.g. Voice – G.711 encoding at both ends) will occur most of the time. If an exact match is not found, a mediation algorithm can be automatically introduced. The transformation engine, guided by service policies, would ensure that this algorithm was added into the data path or loaded onto the client device.

The transformation engine is responsible for determining the QoS/CoS parameters that are contained in the network primitive. When the transformation engine applies a mediation algorithm it is required to specify the corresponding transport QoS requirements.

9.4 Brokerage Engine

Applications, both network-based and terminal-based, must be able to initiate a transport session. To achieve transport media independence, the brokerage engine utilizes the services of the Transport Agent. The exchange and transport agents interact with each other through the brokerage engine. Brokerage engine provides functions for negotiating and securing end-to-end transport sessions that meet the specific QoS requirements.

The brokerage engine offers users a choice of transport services and does not force the user to utilize a specific transport protocol. In large distributed systems, interaction and dependencies between the various components can be described in terms of services they provide to each other. It usually becomes unnecessary to know the inner workings of a component if its functionality can be described. In fact, to the consumer of a service, the cost and quality of the service become more important than its internal implementation.

Network primitives possess a generic specification of transport Quality of Service and/or Class of Service (CoS) requirements for a specific service instance. A network primitive describes transport needs in terms of bandwidth, duration, and accuracy. Network primitives are necessary for service contact negotiations which yield binding contractual agreements.

In order to achieve the maximum utility, communication sessions must be able to include endpoints that do not reside within the boundaries of one network domain. Thus, a communication session originating in one network domain will need to include connections that extend into other networks. In order to maintain interboundary signalling to the greatest extent possible, it is the responsibility of the brokerage engine to initiate, negotiate and secure a transport contract service instance.

The brokerage model is used to facilitate interworking across network domains. There are several possible approaches to implement interworking control, examples are: Exchange agent can dialog directly with Transport Agents either in a centralized or distributed manner; Exchange Agent can dialog with other peer Exchange Agents either in a centralized or distributed manner.

10 Transport Agent

10.1 General scope and functions of the Transport Agent

The Transport Agent is responsible for acquiring the necessary transport resources that will move data[13]. The Transport Agent hides the transport infrastructure details from the Exchange Agent. Interaction between agents is in the form of contracts. It is the responsibility of the Transport Agent to implement transport-specified policies in response to signalling events received from the Exchange Agent. It manages the transport session services and uses a Quality of Service paradigm to interface with the various transport mechanisms.

[13] Data can be either bearer type or message type.

Today if a user wants QoS for different communication services, it can only be provided as a hard choice of separate transport networks. In future, with ICA, users should have the flexibility of QoS selection automatically provided in an integrated manner by means of an automated process. The Quality of Service dimensions of bandwidth, accuracy and time are used to demand the appropriate connection segment from the underlying transport layer. Different transport layers can implement these requests in different technologies, depending on their economic realities. The underlying transport layer provides basic bit transport services and is not concerned with content. It is the responsibly of the Transport Agent residing in a network domain to use whatever control protocols exist to honour the terms of the negotiated transport contract. Furthermore, to speed delivery of services, instead of the current long provisioning process, all services should be readily available, automatically accessible and instantaneously billable when used. Automating choice of service, providing customized service options and offering these capabilities at the network edge is important to achieving both simplified and quicker service access.

Separation using goal oriented contracts key to successful interworking across a federation of multiple transport domains. Allows mapping of requirements at any particular level in the hierarchy from application processing to data transport. Brokerage enables subcontracting between network providers and enables applications to interwork across any transport technology. When ICA brokerage occurs at the domain boundary the transport session is unaware of the content being transport. Transport session control, as it pertains to the specific resources deployed in the network, is determined within the scope on a network domain. It is brokerage that enables applications and adjoining networks to seamlessly communicate over different types of transport domains. The broker process can be controlled by either the network provider or by an independent company (third party).

The role of the Transport Agent is to secure transport services from one or more network providers. The Transport Agent is the point in the process that involves locating and activating transport service(s)[14] that meet requirements identified by the exchange agent. Here is where the actual setting up of a transport session (e.g. circuit connection or data flows) is performed. One scenario is that for first-time users, new transport service requests and session changes are negotiated between consumer and supplier. Repeat users, using pre-negotiated service packages, need only be authorized or recognized. Key to the negotiation process is the concept of real-time brokering of a transport contract. A contract embodies the terms (money), conditions (policies), and services (network primitives) that apply for the life cycle of the multimedia session. This approach enables the entire process of selecting a transport session service to be automatic and universal. (See Figure 13.)

[14] There are many aspects surrounding the movement of information, such as security, point-to-multipoint, broadcast, QoS/CoS performance, and reliability.

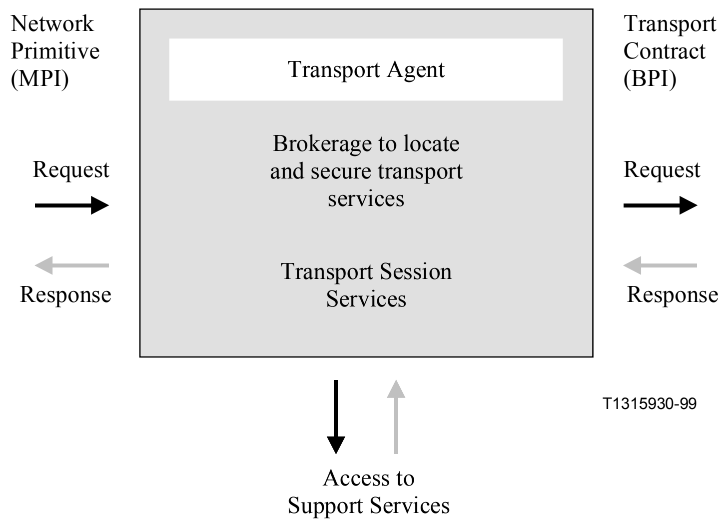

Figure 13/Y.130 – Transport Agent

Negotiation and brokerage are important aspects of next generation networks because of the high probability that future network providers will be offering numerous transport alternatives. In addition, establishing end-to-end interoperability will be difficult to achieve because of a heterogeneous network environment consisting of many operators and transport technologies. Parties will be free to choose between competing network providers and their respective transport session QoS offerings. The Transport Agent enables network providers to offer transport alternatives in a competitive business environment.

In order to perform Transport Agent tasks access to support services may be required. The Transport Agent can issue requests that trigger other support services. Examples of other support services include:

- Security Services;
- QoS or CoS Transport Services;
- Network Administration Services.

The Transport Agent is made up of two major components:

a) the Private Engine;

b) the Public Engine.

10.2 Private Engine

Private engines are envisaged to handle the process of securing network transport services within a given domain. The word private is used to reflect the fact that this broker is inherently owned by and operates on behalf of a network provider. There are three functions associated with private engines. The first function of the private engine is admission control. Here the private engine will match the user policies (such as subscription type, default settings) with the provider policies with respect to admission of new service requests and available resources. The second private engine function will match the service request with the available transport session services. In the ICA, it is assumed that a single transport domain may offer a multitude of transport session services. So the matching functions must be provided by the network operator. The third private engine function is the activation of the requested service. It does this by obtaining the appropriate transport resources for the communication service request. The end result is seamless connectivity initiation across administrative and technological boundaries.

Brokers can be controlled by network provider policies. They dialogue directly with exchange agents. When controlled by a network service provider, the private engine is capable of negotiating a contract, mapping the network primitive to a suitable QoS/CoS transport resource, and supporting contract administration. Negotiation also provides a means of obtaining end-to-end QoS commitment across multiple network providers and heterogeneous transport technologies. When required, the private engine acts as a proxy for network operators in the process of activating or brokering end-to-end transport services. In this case, the private engine provides browsing and location services. It is envisioned that the transport agent containing a private engine may be empowered to negotiate on the network provider's behalf.

10.3 Public Engine

Another type of transport agent engine, referred to as a public engine, can be supported. The public engine acts as an independent mediator between parties and network providers. The public engine can search across all a multitude of transport domains to find the best match. The match is based on cost and product capabilities much the same as a consumer purchasing goods from various retail stores. The result of this function is the identification of network transport providers that meet the criteria defined by Exchange agents.

Public engines will act on behalf of a party to negotiate a service contract with a network operator. A contract is based on the parameters contained in a network primitive and the terms/conditions reflected in policies. Many suppliers will be involved in providing the solution elements and will often be in competition. Communication service providers will support their product brand integrity and solution fit by managing their agent and broker processes.

Fundamental to the negotiation process is making a match between what the consumer requires and the provider is offering. The term "broker" is used to refer to the entire process of locating and negotiating. The benefit of brokers in agent-based architectures can be shown by analogy to how search engines benefit people surfing the Web. A public engine can discover other instances of similar transport services by querying the transport agent for similar service descriptions. Once found, transport session services are negotiated and utilized.

11 Transport Services

The decoupling of application sessions from transport sessions is a key concept of ICA.

The provision of Transport Services is, in itself, a complex issue. Whilst ICA will be designed to be decoupled from, and independent of Transport technologies, an understanding of the basic positioning of Transport services and technologies with respect to ICA is required.

Generally speaking, the technologies and protocol stacks involved in the provision of transport services are many and various. For example, an IP service could be provided over ATM, SDH, Frame Relay (FR), X.25, ISDN, etc. In the case, say of IP over ATM, the ATM itself might be provided over SDH. Similarly FR can be over ATM, or directly over leased facilities, etc. Similarly, X.25 might be over FR, or ATM. Many different combinations of protocols may be encountered in the transport area. Just as IP may be carried over X.25, it is also equally possible to carry X.25 over IP. Thus, ICA must be capable of dealing with a large number of complex cases, and capable of providing service at a number of different layers from instance to instance of communication, with appropriate end-to-end QoS. In some cases some or all of the layers of the protocol stack terminate solely inside the ICA, or terminate between the user and some point within the ICA, or solely between the end users.

To cope with this situation the ICA assumes a general, multi-layered protocol model as previously described in 6.7.

The layers, n and below, each forms a separate "layer network".

A layer network is a collection of functions required to create a logical network. As such it needs to include transport functions such as nodes and links as well as routing and traffic control functions. In addition, a layer network needs to incorporate some form of an addressing scheme to enable traffic to successfully transit the layer network.

NOTE – The term "layer network" as used in this ITU-T Recommendation is identical to that defined in ITU-T Recommendation G.805.

When separated from its physical implementation, the Internet is a layer network, as is the PSTN, a frame relay network, an ATM VC network, an ATM VP network, a SONET STS1 network, etc. These are all separate layer networks.

Three primary features defining a layer network and its associated protocol are:

- A single format for the transport of user data – e.g. the IP packet payload for the Internet, the frame payload for a frame relay network, the STS1-SPE (a continuous bit rate of 48 384 kbit/s) for the STS1 layer network.
- A universal, consistent end point addressing scheme.
- A universal, consistent traffic control and management scheme.

Any layer network contains a number of functional areas. These functions may be implemented in a wide variety of ways including manual processes and network management processes as well as automated processes. These functions include:

- topology design and control;
- attachment of an end point to a layer network together with an address assignment for the end point;
- route table construction;
- resource management;
- transport including forwarding/switching and transport across logical links.

An important property of layer networks is that they can be decomposed into subnetworks ("domains" is an alternative term) and this decomposition can be carried out recursively. An important use of subnetworks is to describe the topology and routing across the topology of the layer network. This property is normally referred to as partitioning of the layer network.

The partitioning of the layer network ultimately reveals its topology in the form of elementary subnetworks and links between them. The links are aggregate flows of traffic of the layer network and are normally supported as an end-to-end traffic flow on another layer network, most often a connection. For example, links in the Internet can be supported by frame relay connections, ATM VC connections, etc.; links in an ATM VP network can be supported by SONET STS3c connections; etc. When this occurs, a layer network becomes the client of another, establishing a set of client/server relationships between layer networks. One layer network can have many server layer networks and one layer network can have many client layer networks. The client/server relationship between layer networks is a many-to-many relationship.

Functional areas which are required in the adaptation between layer networks include the following:

- buffering;
- multiplexing;
- encapsulation;
- client link address to server end address mapping;
- transformations.

These are the two principal recursive properties associated with layer networks – partitioning within a layer network and layering between layer networks.

11.1 Implications for ICA

ICA must adapt, dynamically, to variations in the transport protocol stack architecture, for three primary reasons.

Firstly, the type of transport service requested may be protocol dependent, e.g. IP when IP service has been explicitly requested. This is particularly true for the primary protocol layer.

Secondly, the ICA may select the underlying layers on the basis of QoS, cost, security and other parameters.

Thirdly, the transport agent may be required to offer service at any and all of the layers n-k to n, depending on the type of service requested by the user and that offered by the particular provider concerned.

12 Target values

ICA network-based agents address the business and technological opportunities that exist between the world of content applications and the world of networking. The architecture being proposed addresses the challenges of party identification, device independence, and adaptive transport that are expected to be key enablers for personal multimedia communication. ICA provides a methodology for rolling out intelligent network-based services that solve the challenges of establishing end-to-end multimedia communication across heterogeneous networks.

ICA does not require a single implementation architecture for all networks, just as an IP-based network does not require a single implementation architecture for all the computers that are connected to it. As a network-based middleware proposal, it is designed to fit between content applications and transport providers. The associated benefits of this approach are:

- Decouples users from the complexities of establishing communications.
- Adapts the network to user's capabilities and needs, rather than forcing the user to adapt to network's offerings.
- Allows network and service providers to utilize multiple technologies to meet user requirements in the most cost-effective fashion.
- Separation of content applications from transport isolates applications from the impact of technology changes.
- Creates a path towards an open network-based platform capable of supporting diverse range of multimedia based content applications.
- Self-service network approach (agent collaboration) enables greater collaboration across services.
- Requirement driven interface between applications and networks future proofs the overall of design of network services.
- Creates a universal framework for multivendor interoperability.
- Provides a vehicle for network vendors to deliver value-added features (e.g. encryption, authentication, flexible billing, and virtual networking policies).
- Provides an opportunity to move away from the inherent difficulty of supporting a large number of vertically integrated products.
- Simplifies the design and support of network equipment.

ICA is targeted at providing network-based middleware services that intelligently bridge the gap between smart edge and smart networks. The ICA proposal addresses the key challenge of providing diversity of network services while maintaining simplicity of use.

Building communication solutions based on a functional approach has the advantage of allowing flexibility in how agents are packaged. It is expected that ICA will not be implemented as a complete system; rather, it will be realized though the evolution of several products. These products will reflect the challenge of delivering party identification, device independence, next generation session management and multivendor interworking. With this in mind, each agent has been positioned to possess values reflecting these challenges and can be deployed as a standalone product offering.

Contact agent values:

- Decoupling device from communicating party.
- Recognition based on personal or party Ids.
- Freedom to select alternative devices.
- Profiles allow customization of communication requirements.
- Permits various forms of addressing, e.g. individual, group, etc.
- Permits the definition of personalized profiles based on temporal and geographic information, which can be processed in different ways according to context of their usage and applicability.

Exchange agent values:

- Streamlines the selection of devices.
- Enables communication between devices and networks that cannot normally interoperate (i.e. by employing transformations and proxies).
- Primitives enable increased interoperability across transport technologies.
- Allows bridging of value-added services by either edge or network-based solutions.
- Automatically adapts to user differences and session changes.
- Permits the provision of mediation services.

Transport agent values:

- Transport activation process appears simple, familiar and straightforward.
- Customizes movement of information behaviour for end parties.
- Enables end-to-end interoperability across any networking technology or operator domain.
- Enables automation of administrative information related to users/customers.

Content providers will be able to build content solutions that can operate over any communication infrastructure. ICA network-based middleware services will enable content providers to deploy multimedia based content applications with greater simplicity and ubiquity.

Network providers will be able to build network solutions that consist of products from multiple vendors. ICA network-based middleware services will enable network providers to build competitive solutions over heterogeneous computing and networking environments. It also offers an evolution path for both IP and TDM networks.

13 Implementation model

The functions provided by ICA need to be distributed in order to meet real-time and scalability requirements. Since ICA is an intermediary control system for network services, most functionality resides between end applications and transport path technologies. This implies that in next generation networks ICA agents will reside, in any or all of:

- local intelligence associated with edge devices;
- network gateways;
- communication servers.

Within the ICA there is an element in the network called the Communication Server. The Communication Server contains higher-level network services. In principle this is similar to IN SCP present in narrow-band networks. However, there is one significant difference. The Communication Server does not rely on switch-based service triggers to make decisions. Rather, contact agents direct end party signals to the Communication Server. Exchange agents residing in the Communication Server signal to network elements requests to move information based on network primitives. This will have a significant impact on the design of emerging data-centric networks as switching and transport equipment will be relegated to path connectivity controls specific to moving information based on network primitives and policies.

With such a flexible implementation capability it is difficult to draw an implementation picture. However, Figure 14 provides a simplistic view of the logical positioning of the agent functionality with respect to access and core networks.

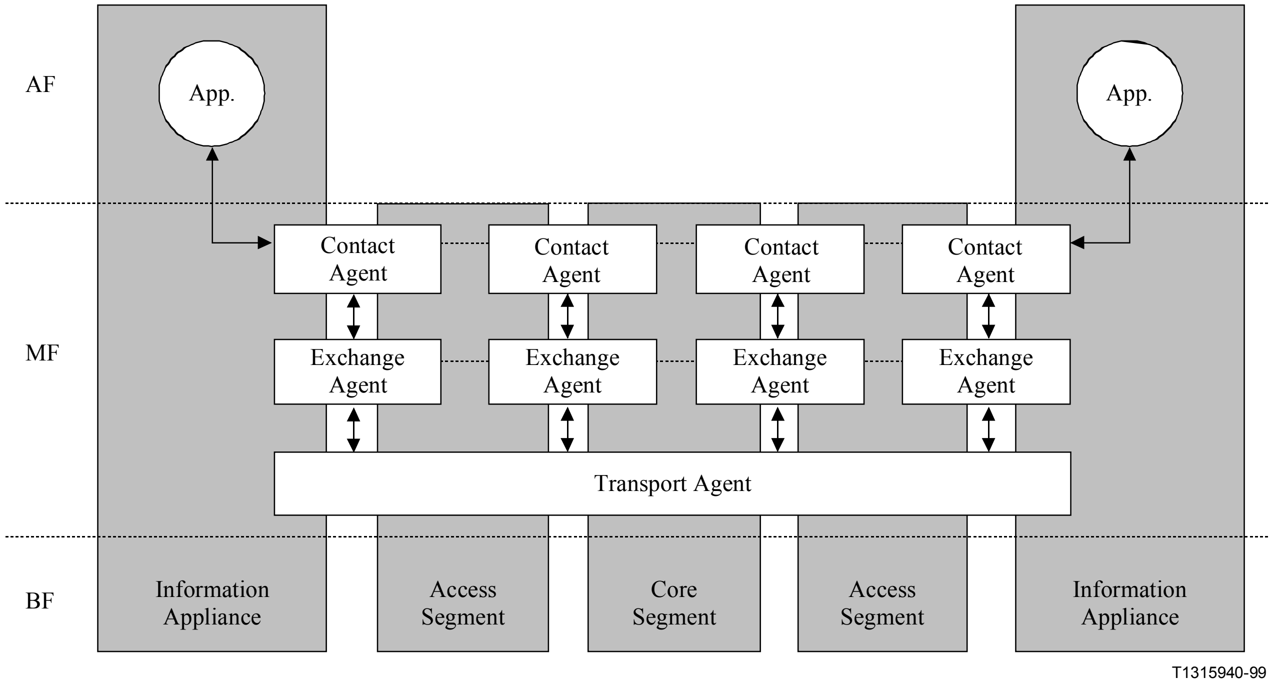

Figure 14/Y.130 – Basic implementation aspects

Agents residing in client devices and servers will involve aspects of party discovery. Agents residing in gateways and communication servers will involve aspects of device alignment, connectivity control, and network engagement. Each provider, whether they are enterprise or carrier specific, will possess their own Communication Server and associated databases. This means that contact and exchange agents will need to have controlled access to various databases in order to determine the appropriate user profiles and transport policies.

The use of middleware has been firmly embraced by the computing industry and it is easy to see that ICA-like functionality may first appear in content servers and end devices. However, commercial computing platforms have traditionally not been adequate for supporting mission-critical real-time applications such as those inherent in large scale networks.

The Internet and object technology industries are two key driving forces that are shaping middleware infrastructures in today's networks. Many existing middleware providers are moving towards such cohesive architectures, but from different points of view. For example:

- Database systems are moving towards open distributed transactions.
- Agent solutions are evolving towards mobile and cooperative frameworks.

- CORBA and Java solutions provide component brokers as Internet middleware for distributed objects.
- Standards groups (IEEE PIN, TINA and OMG) are focusing on generic network services.
- Computer vendors are developing architectures that create seamless environments for developing multi-tiered distributed computing applications.

It is commonplace knowledge that the development of distributed applications from scratch (i.e. with only protocol messages defined) requires an expert's effort, and the resultant applications are costly to maintain. This is further complicated by the necessity to develop different parts of applications in different programming and execution environments. Still added to this complexity is the cumbersome task of porting, and often rewriting, these applications to suit different platforms. Network Operating Systems address these network heterogeneity-related issues.

Further information on the relationship of ICA to possible implementation architectures, such as TINA, CORBA, Network Operating Systems, etc. are given in the annexes.

Annexes:

ANNEX A

TINA and the ICA

The primary purpose of the ICA is to define a partitioning of the major functional elements between a user (or a user's application/equipment) and the information transport facilities, such that the services and associated interfaces presented to the user may be developed as independently as possible from the technology of the transport facilities, and vice versa. The partitioning into three layers constitute a separation of functional concerns.

The TINA architecture has similar objectives. In order to simplify and speed the introduction of new telecommunication services, the TINA architecture proposes that the functions associated with services [the Service Architecture (SA), and Network Resource Architecture (NRA)] be both logically and physically separated from the network elements that constitute the information transport facilities. The TINA architecture also addresses the ODP computing and enterprise viewpoints by specifying a specific implementation technology (TINA DPE – presently a superset of OMG CORBA), and by defining a business model. The business model recognizes that several stakeholders may need to cooperate in providing a user with a telecommunication service. The architecture then defines the replication and partitioning of the SA and NRA among the various stakeholders in order to support such cooperative services. The points of connection between the stakeholders' parts of the system are known as reference points and work on defining and refining these reference points is ongoing in the TINA consortium.

From these brief descriptions it is apparent that the TINA architecture meets the major objectives of the ICA, that of decoupling the services presented to the users from the technology (and business arrangements) of the underlying transport infrastructure. Nevertheless, the two architectures should be viewed as complementary, not competitive.

The ICA, unlike TINA, does not address the engineering viewpoint. Although the physical separation of the SA and NRA from the Network Elements (NEs) offers significant benefits, it places enormous demands upon the hardware and software platforms used to host the SA and NRA. Presently, the specialized and optimized hardware and software platforms of the NEs host major parts of the "service logic", and they will continue to do so for at least the next few years. However, it should once again be noted that the ICA does not depend upon, or specify, the physical location of these functions.

An implication of the separation of service logic from NEs is that the information transport function is controlled from outside of the NEs. At present, the transport function is typically controlled through signalling, and the NEs often include specialized hardware and software features to support such control mechanisms. A key aspect of the ICA is the independence of the users data traffic from the control of the transport infrastructure. In effect the ICA is saying that the means to control the transport infrastructure is whatever it takes to control the available transport infrastructures, and that the nature of the user data traffic and the control mechanisms are decoupled. If the transport infrastructure is controlled by a TINA mechanism, so be it, but it is certainly not a requirement of the ICA.

Thus we see that in these two major respects, the ICA is significantly less radical than the TINA architecture. This is a natural consequence of what is the major driver behind the ICA, namely to ensure that the existing, and near future, transport infrastructure of Public Network Operators offer as much value as possible in the face of rapidly evolving services. However, it should be noted that there is no overt conflict between the two architectures at this high level. An ICA system could be built making use of the TINA architectural principles.

When we consider the details of the SA we see considerable convergence. User profiles and session concepts are well defined in the TINA SA specifications, and will be similarly important in the ICA. Indeed, as the ICA work advances it would be inappropriate not to follow the TINA work to the greatest extent possible. Nevertheless, there will, inevitably, be differences in emphasis or substance, which will require either extensions or modifications to the TINA work. An example of the former may lie in the Broker function. The driver in the TINA architecture is the business model, and the brokerage function emphasizes selection of a carrier or service provider. In the ICA the Transport Broker deals with the selection of a transport technology. Of course, in the final analysis, both are necessary, and it is to be hoped that a single definition will suffice for both the choice of carrier/technology and hence serve both TINA and the ICA.

Another example is that the TINA architecture separates the sessions associated with requesting a service and the session associated with a service. In the ICA, legacy applications and user behaviours may not distinguish between the two. Given the importance of avoiding discontinuities in the evolution of services, the ICA will need to accept and deal with this situation. Similarly, a service request might also be interpreted as an implicit service subscription request by the ICA. Of course, in the actual implementation of the ICA the various functions of subscription, service request and service will be clearly separated and may well draw upon much of the TINA service architecture.

Yet another example is the issue of technology-independent communication primitives. To some extent this is inherent in the TINA architecture; however, given that a prime *raison d'être* for the ICA is technology independence, it is anticipated that in this matter the requirements of the ICA may be more stringent than those of the TINA architecture.

ANNEX B

The ICA Broker function and the OMG CORBA broker

The Broker function in the ICA selects a network, or combination of networks, to carry a user's data from source to destination. Various criteria will be applied, starting with the networks ability to meet the user's needs. It is generally anticipated that a fundamental aspect of the choice will be the selection of transport technology, in combination with the carrier's policies. The Broker function in the TINA architecture is generally expected to choose between carriers who each offer the same transport technology.

Within OMG software functions have been defined which select computational objects based upon a set of criteria. The Broker function of CORBA (Common Object Resource Broker Architecture) searches for computational objects that fully match criteria. The Trader function searches for computational objects that partially match criteria.

Clearly, if in the implementation of the ICA the network capabilities are represented by an appropriate computational object, the OMG Broker and/or Trader capability may provide a simple implementation of the ICA Broker function. However, other engineering considerations may preclude such an approach, in which case the ICA Broker function would need to be "custom programmed". The ICA architecture, which eschews the computational viewpoint, allows for either approach.

ANNEX C

Network Operating Systems as platforms for the ICA

There is little doubt that an implementation of the ICA will have to draw heavily upon the advances in the computing software industry, most notably that of Network Operating Systems.

The development of distributed applications, such as a practical implementation of the ICA, from scratch (i.e., with only protocol messages defined) requires an expert's effort, and the resultant applications are costly to maintain. This is further complicated by the necessity to develop different parts of applications in different programming and execution environments. Still added to this complexity is the cumbersome task of porting, and often rewriting, these applications to suit different platforms. Network operating systems are to address these network heterogeneity-related issues.

The network operating system is a comparatively new phenomenon, and there is not yet a consensus in the industry on its definition.

One such definition (Tanenbaum) is that a network operating system provides a single computer image of the network while permitting a high degree of autonomy of each computing node. It is a layer of functional capabilities residing above operating system and low-level networking software used by distributed applications to enable inter-operation and resource sharing. This kind of network operating systems is sometimes referred to as *middleware,* which is, however, an overloaded term. Open Software Foundation's Distributed Computing Environment (DCE) and Object Management Group's Common Object Request Broker Architecture (OMG CORBA) are examples of this type of middleware. The availability of this type of software provides the ICA with *ready-made* locational transparency.

However, this type of middleware does not offer assurance of portability to different platforms. Further, on each particular machine, the middleware needs to be supported by an operating system, which makes it impossible for middleware to run on some rudimentary user terminals such as set-top boxes. Continual research on removing these limitations is leading to the advent of full network operating systems that add platform independence to Tanenbaum's definition.

Running either standalone on inexpensive hardware or as an application under conventional operating systems, a full network operating system provides a complete environment for portable distributed applications through multiple levels of abstraction:

A Virtual Machine (VM), serves as the common execution environment for all applications. Applications built on the full network operating system are compiled into its instruction set, which ensures portability across hardware platforms. Further, the instruction set of the virtual machine matches well with those of the modern machines so that applications are executed with comparable performance characteristics.

A Virtual Operating System (VOS), serves as both an extended machine and a resource manager. In the former role, it provides an Application Programming Interface (API) shielding the application developer from the low level hardware and network and unpleasant business dealing with, for example, inter-process communications, networking, and security. In the latter role, it performs the actual task of resource management, including the support of transparent secure sharing of resources among processes throughout the network.

A Virtual Network Interface (VNI), provides a common network interface to the applications. Applications therefore can be developed independent of the underlying network or transport. (This network, which supports the communication necessary to a distributed implementation of the ICA may, or may not be part of the network resources which carry the user traffic. In the TINA architecture, it is referred to as the kernel transport network. In present day voice networks the situation is analogous to the existence of a SS7 Signalling network, separate from the Voice network.)

This extended view of the network operating system meets many of the implementation objectives of the ICA. Therefore, in later phases of ICA work, it will be appropriate to use this definition as the basis for developing related interfaces in the ICA.

ANNEX D

Global Mobility and the ICA

D.1 ETSI framework for Global (Terminal) Multimedia Mobility

Global Mobility denotes the mobility aspects, both personal and terminal, resulting from the convergence of telecommunications, information technology and entertainment services. A basic GM assumption is that future terminal equipment should be able to connect to several types of access systems. The choice of access will be made dynamically and will depend on a variety of factors such as the application service requested by the user, the service subscription, and the access systems available locally. A variety of access systems can be identified which include 3rd Generation Mobile Systems, GSM-BSS, cordless access, satellite (S-PCN) and fixed access. GM indicates the dynamic use of multiple access systems that will enable high bit rate services to be introduced gradually according to market demand. GM envisages several core networks and one of prime importance is the 3rd Generation Mobile System that will be optimized for radio access.

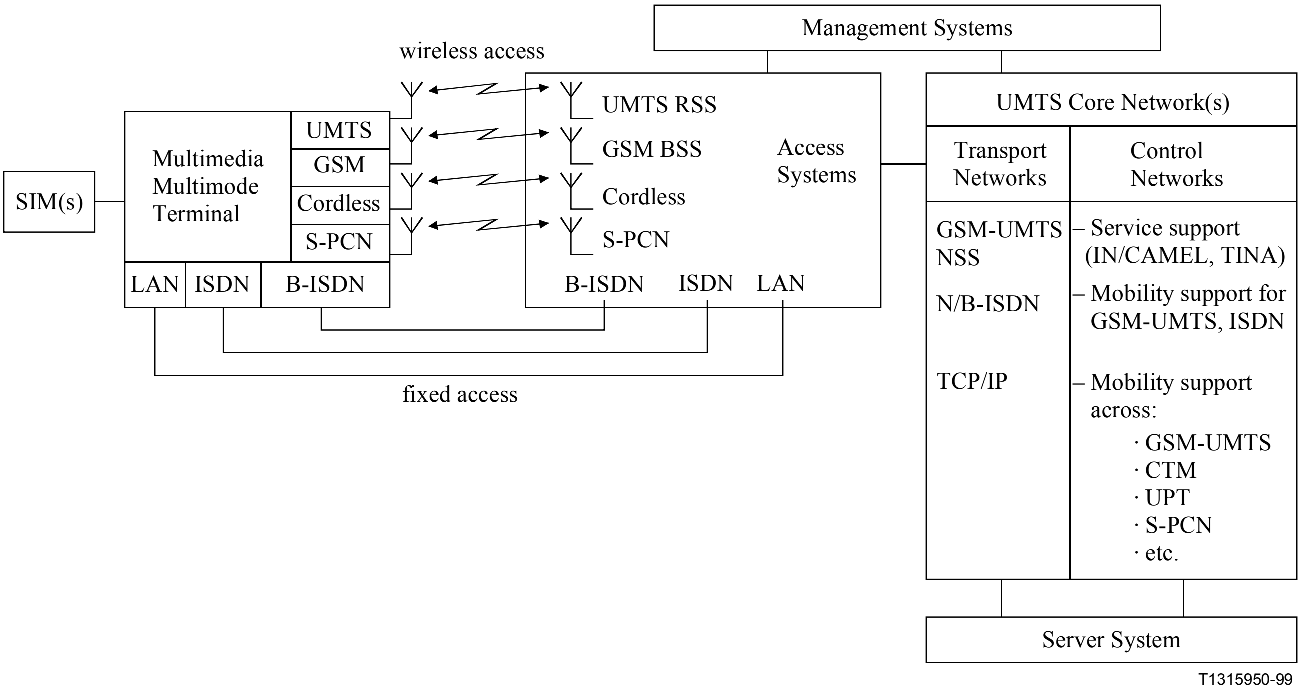

Figure D.1/Y.130 – ETSI Framework for Global (Terminal) Multimedia Mobility

There are two pole approaches to support global mobility in the ICA:

- the Top Down approach; and
- the Bottom Up approach.

In the Top Down approach global mobile communication services are viewed as any other information communication service. In this approach, the assumed underlying baseware involves both mobile and fixed transport networks deprived of intelligence to control and manage resources. Such intelligence is considered to be part of the Middleware provided by the ICA.

In the Bottom Up approach, intelligence for the support of mobile communication services is considered as an available part of the baseware. In this sense the mobile systems, for example GSM, and IMT-2000, are assumed to provide the functions needed in support of mobility; the Middleware will include an interface to these systems as it does to other basic information communication services.

D.2 Impact of Mobility on the ICA in the Bottom Up Approach

In the SG 13 preferred *Bottom Up* approach assumes the existence of a mobility-supporting system and relieves the Middleware to include an interface to mobility services as well as other basic information communication services.

The existing mobile system is assumed to provide terminal mobility and to support personal mobility in its most general standardized concepts.

It is a well-established and widely accepted position that third-generation mobile systems, like IMT-2000, will support the provision of a unified seamless presentation of Personal Communication Services to the user in fixed and mobile environments. In the context of applications, this means that the same services can be provided by both fixed and mobile networks through a user-network interface that does not bother users with mobility aspects. The main restriction that hinders these systems from offering services identical to those of fixed networks is the 2 Mbit/s local area, and 155 Mbit/s wide area, maximum capacity of the IMT-2000 air interface. A fixed environment is assumed to be aligned with the B-ISDN capabilities and, therefore, services in this environment can exploit the high bandwidth provided by such fixed networks.

By taking the above considerations into account, the Bottom Up approach addresses IMT-2000 as the mobile supporting system existing in the baseware. The main reasons for this choice are the following:

- IMT-2000 will be the major mobile system at the time of B-ISDN deployment, which will boost the design of value-added services;
- IMT-2000 will be integrated with B-ISDN; and
- IMT-2000 will provide a B-ISDN compatible interface.

IMT-2000 providing a B-ISDN compatible interface is of special importance for Middleware design, as it minimizes the influence of mobility on the Middleware provided that it is capable of dealing with a B-ISDN fixed network infrastructure.

In accordance with the above assumptions, and considering that mobility functions are already provided in the baseware that includes third-generation mobile systems, the main impact on the ICA is that the Middleware has to deal with the interface provided by the underlying mobile system and has to enhance its quality. In other words, the Middleware only has indirect control over mobile system functions. In particular, the Middleware only has control over the quality parameters of the information communication service provided at the interface between the Middleware and the mobile system.

The main task of the Middleware is, therefore, to protect the quality of the services it supports by properly handling the quality and performance parameters associated with the information communication services provided by the underlying mobile system, according to what has been stated in this field by the current standards for the provision of advanced information communication services set by the B-ISDN community.

Concluding, in the SG 13 preferred Bottom Up approach the factors to be taken into account in evaluating the impact of mobility on the ICA are:

- The user control over functions and features supporting terminal mobility is filtered and therefore limited by the user-network interface, which will hide most of the mobility-supporting internal functions of a mobile network.
- The only possibility to interact with the mobile system is provided by the protocols supported by the IMT-2000 user-network interface.
- A mobile system is not always capable of providing the same quality of service as a fixed network, because of the low transmission rate supported by the IMT-2000 air interface.

ANNEX E

Abbreviations

BPI	Baseware Programming Interface
CF	Control Functions
CORBA	Common Object Request Broker Architecture
CTN	Core Transport Network
DCE	Distributed Computing Environment
GII	Global Information Infrastructure
HCI	Human Computer Interface

HCIF	Human Computer Interface Functions
ICA	Information Communication Architecture
IN	Intelligent Network
KTN	Kernel Transport Network
LOS	Local Operating System
LTA	Long Term Architecture
ManF	Management Functions
MF	Middleware Functions
MPI	Middleware Programming Interface
NE	Network Elements
NF	Network Functions
NOS	Network Operating System
ODP	Open Distributed Processing
ODP-RM	Open Distributed Processing Reference Model
OMG	Object Management Group
P&SF	Processing and Storage Functions
RP	Reference Point
SCF	Service Control Function
SN	Service Node
SPI	System Programming Interface
STN	Signalling (Kernel) Transport Network
TAEE	Telecommunication Architecture for Evolving Environment
TF	Transport Function
TINA	Telecommunication Information Network Architecture
TMN	Telecommunications Management Network
TN	Transport Network
TRP	Telecommunications Reference Point
VM	Virtual Machine
VNI	Virtual Network Interface
VOS	Virtual Operating System

ANNEX F

Glossary of terms

F.1 **agent**: An agent is an element that performs some task on behalf of some party (i.e. user, machine, application, or another agent), rather than have the party, itself, perform the task. The term "party" refers to either a client (user) or server application involved in communicating with others.

F.2 **engine**: An engine is the realization and mechanization, in software or hardware, of one or more functions dedicated to performing a specific task. Well-known examples of engines include search engine, encryption engine, etc.

F.3 **service primitive**: An abstract, implementation-independent interaction between a service user and the service provider.

F.4 **service provider**: An organization that provides services for consumption by some third party (user).

F.5 **user (service-user)**: A consumer of services. In this context "user" covers both "end user" as well as "user" as a client in any client server situation. In addition, content providers and service providers may also be considered to be users.

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series B	Means of expression: definitions, symbols, classification
Series C	General telecommunication statistics
Series D	General tariff principles
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Construction, installation and protection of cables and other elements of outside plant
Series M	TMN and network maintenance: international transmission systems, telephone circuits, telegraphy, facsimile and leased circuits
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks and open system communications
Series Y	**Global information infrastructure and Internet protocol aspects**
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2001