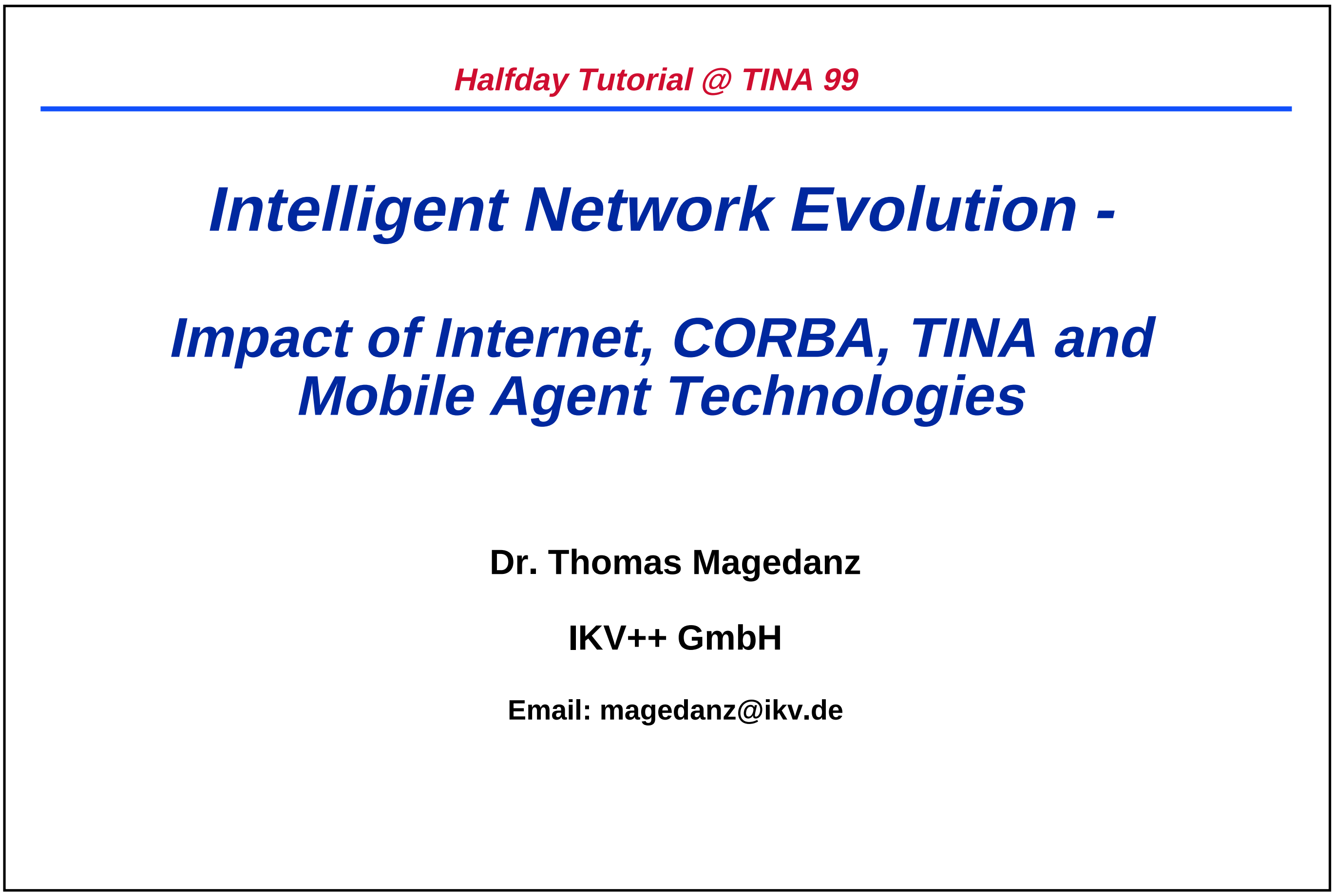


Halfday Tutorial @ TINA 99

Intelligent Network Evolution -

Impact of Internet, CORBA, TINA and Mobile Agent Technologies

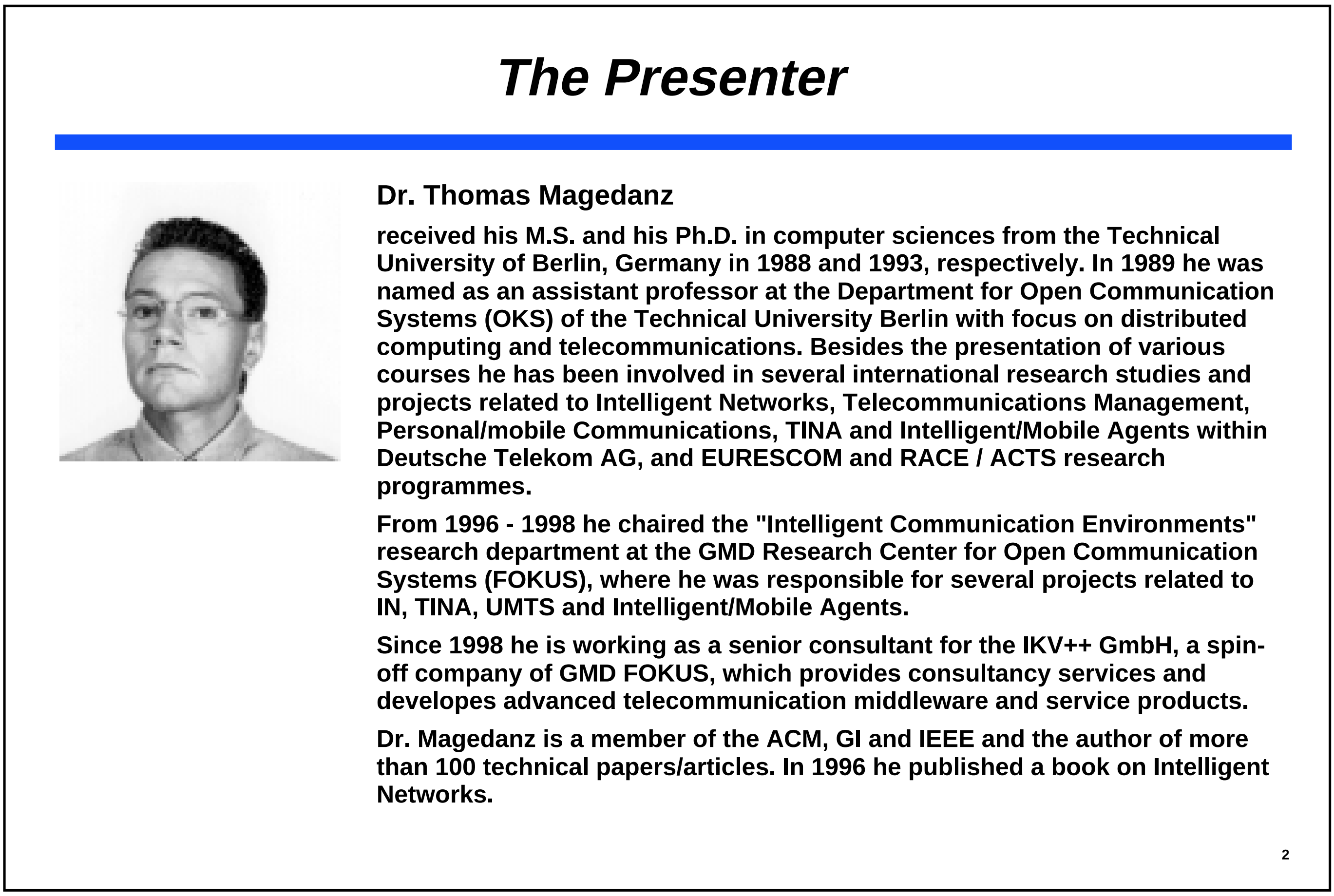
Dr. Thomas Magedanz

IKV++ GmbH

Email: magedanz@ikv.de

Abstract

The main concern of this tutorial is the evolution of the "Intelligent Network (IN)" which is in principle designed as an evolvable telecommunications middleware solution. After a brief review of the IN principles, this tutorial looks at the main impacts on IN evolution derived from new bearer network environments / application domains (e.g., Voice over IP networks) and emerging middleware technologies (i.e., distributed object technologies and mobile agent technologies).



The Presenter

Dr. Thomas Magedanz

received his M.S. and his Ph.D. in computer sciences from the Technical University of Berlin, Germany in 1988 and 1993, respectively. In 1989 he was named as an assistant professor at the Department for Open Communication Systems (OKS) of the Technical University Berlin with focus on distributed computing and telecommunications. Besides the presentation of various courses he has been involved in several international research studies and projects related to Intelligent Networks, Telecommunications Management, Personal/mobile Communications, TINA and Intelligent/Mobile Agents within Deutsche Telekom AG, and EURESCOM and RACE / ACTS research programmes.

From 1996 - 1998 he chaired the "Intelligent Communication Environments" research department at the GMD Research Center for Open Communication Systems (FOKUS), where he was responsible for several projects related to IN, TINA, UMTS and Intelligent/Mobile Agents.

Since 1998 he is working as a senior consultant for the IKV++ GmbH, a spin-off company of GMD FOKUS, which provides consultancy services and developes advanced telecommunication middleware and service products.

Dr. Magedanz is a member of the ACM, GI and IEEE and the author of more than 100 technical papers/articles. In 1996 he published a book on Intelligent Networks.

2

Dr. Thomas Magedanz received his M.S. and his Ph.D. in computer sciences from the Technical University of Berlin, Germany in 1988 and 1993, respectively.

In 1989 he was named as an assistant professor at the Department for Open Communication Systems (OKS) of the Technical University Berlin with focus on distributed computing and telecommunications. Besides the presentation of various courses he has been involved in several international research studies and projects related to Intelligent Networks, Telecommunications Management, Personal/mobile Communications, TINA and Intelligent/Mobile Agents within Deutsche Telekom AG, EURESCOM and RACE/ACTS.

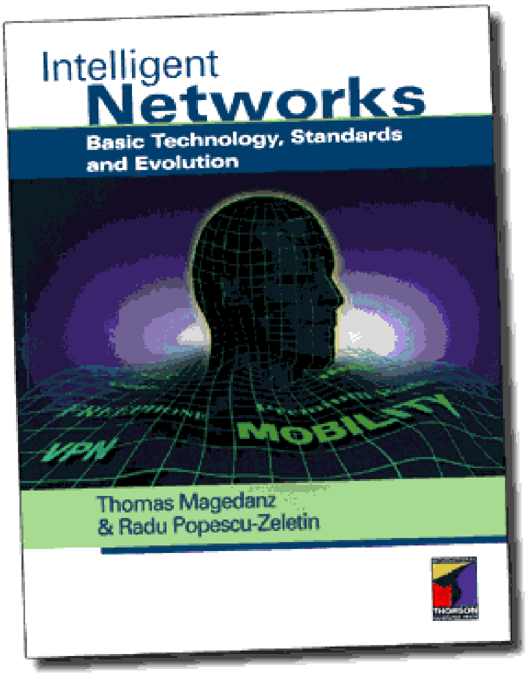
From 1996-1998 he chaired the "Intelligent Communication Environments" research department at the GMD Research Center for Open Communication Systems (FOKUS), where he was responsible for several projects related to IN, TINA, UMTS and Intelligent/Mobile Agents. In addition was one of the founding director sof the FOKUS Competence Center for Intelligent Mobile Agents (IMA).

In 1998 he joined IKV++ GmbH, a spin off company of GMD FOKUS, as senior consultant, where he is leading multiple international projects related to fixed and mobile telecommunications middleware.

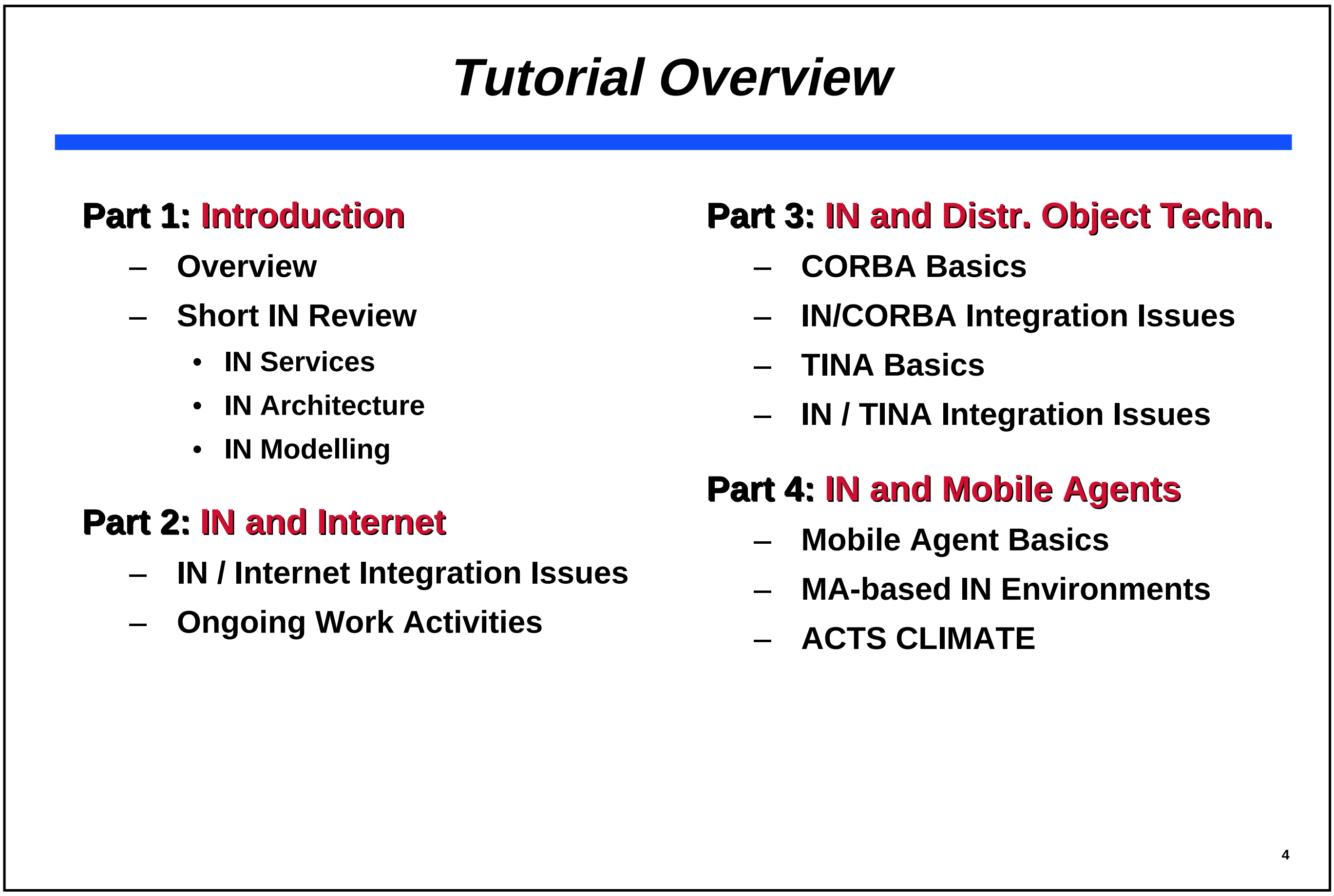
Dr. Magedanz is a member of the ACM, GI and IEEE and the author of numerous technical papers/articles. In 1996 he published a book on Intelligent Networks. Since two years he giving professional seminars related to IN, TINA and Intelligent Mobile Agents.

The Background Book to this Tutorial

- **Title:**
  - **"Intelligent Networks - Basic Technology, Standards and Evolution"**
- **Authors:**
  - **T. Magedanz, R. Popescu-Zeletin**
- **Publisher:**
  - **International Thomson Computer Press,**
  - **ISBN: 1-85032-293-7,**
  - **London, 1996**
- **Price: 69.- DM / 25 £**



The above book presents the information given in this course in a compact form particularly to newcomers. However, the course material you have in hand is more recent.



Tutorial Overview

Part 1: Introduction

- – Overview
- – Short IN Review
  - • IN Services
  - • IN Architecture
  - • IN Modelling

Part 2: IN and Internet

- – IN / Internet Integration Issues
- – Ongoing Work Activities

Part 3: IN and Distr. Object Techn.

- – CORBA Basics
- – IN/CORBA Integration Issues
- – TINA Basics
- – IN / TINA Integration Issues

Part 4: IN and Mobile Agents

- – Mobile Agent Basics
- – MA-based IN Environments
- – ACTS CLIMATE

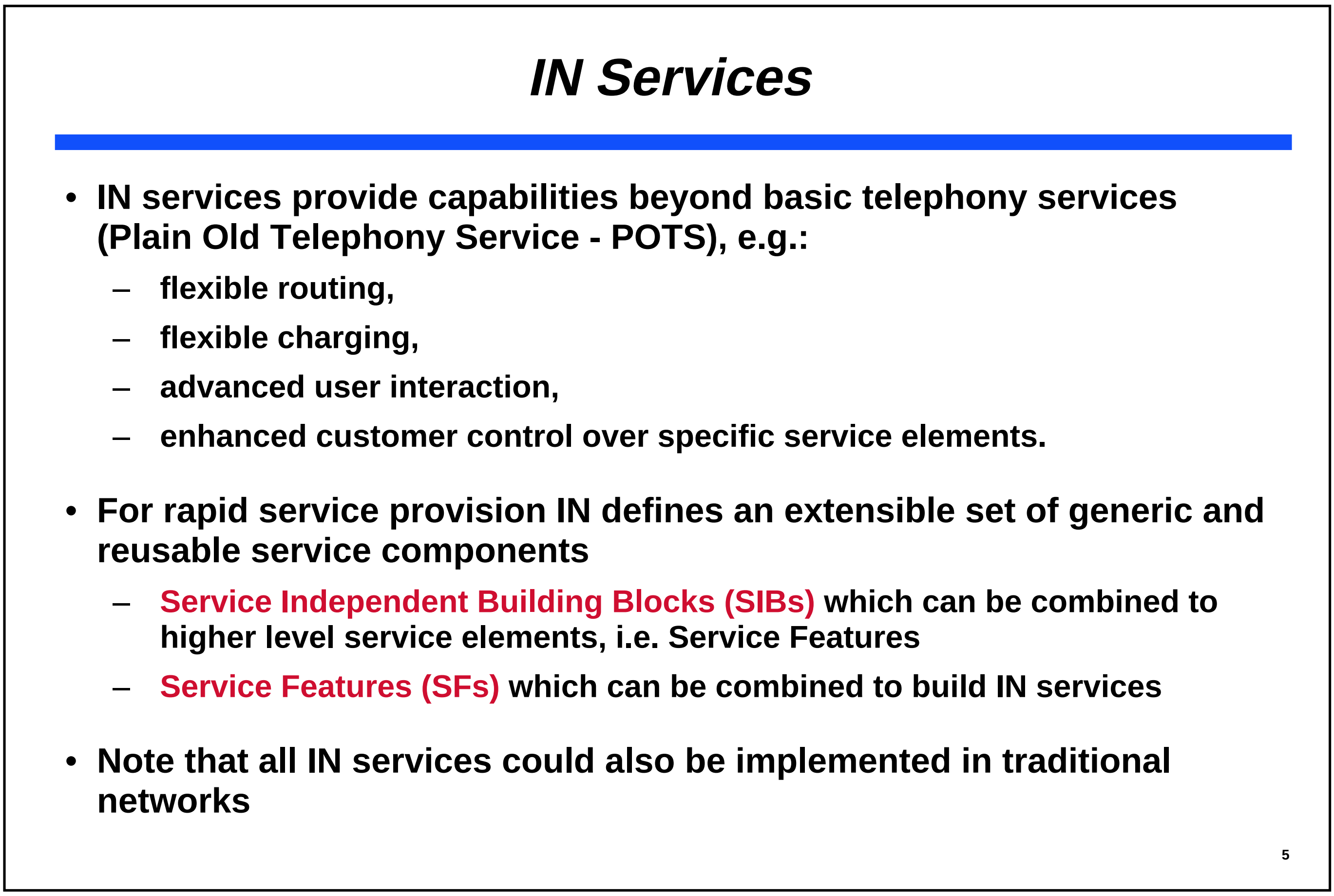
4

Part I provides a short review of the main IN principles and standards.

Part II looks at the emerging integration of IN and Internet, covering an overview of potential integration issues, such as Web-based IN Management, Web-initiated Telephony services and Web-IN, and the usage of IN principles for provision of value added services within Voice over IP networks. Furthermore this part provides a overview of related standards fora, such as ETSI TIPHON, IETF PINT, Parlay, JAIN, etc.

Part III investigates the impact of new object-oriented middleware technologies and new open service architectures on IN. In this context the Open Management Group’s (OMG’s) Common Object Request Broker Architecture (CORBA) is becoming the de facto standard for distributed object computing middleware and is gaining importance in telecommunications. We provide an overview of CORBA and outline the current IN/CORBA integration activities being undertaken by the OMG. Furthermore, we provide an overview of the Telecommunications Information Networking Architecture (TINA), gaining importance as a universal open service platform for the near future, and address potential migration steps from IN-based toward TINA-based platforms looking particularly at the work performed within the TINA-IN Working Group.

Part IV introduces the emerging paradigm of Mobile Agents (MAs) and introduces related technologies and standards. An outline of emerging MA-based IN environmentscompletes the course.



IN Services

- **IN services provide capabilities beyond basic telephony services (Plain Old Telephony Service - POTS), e.g.:**
  - **flexible routing,**
  - **flexible charging,**
  - **advanced user interaction,**
  - **enhanced customer control over specific service elements.**
- **For rapid service provision IN defines an extensible set of generic and reusable service components**
  - **Service Independent Building Blocks (SIBs) which can be combined to higher level service elements, i.e. Service Features**
  - **Service Features (SFs) which can be combined to build IN services**
- **Note that all IN services could also be implemented in traditional networks**

Generally, it has to be stressed that it is not within the scope of IN to define any specific IN services, since the IN aims to provide a service-independent network platform.

However, one may identify specific functional commonalities, i.e. common service elements, within IN services, providing:

• flexible routing

• flexible charging

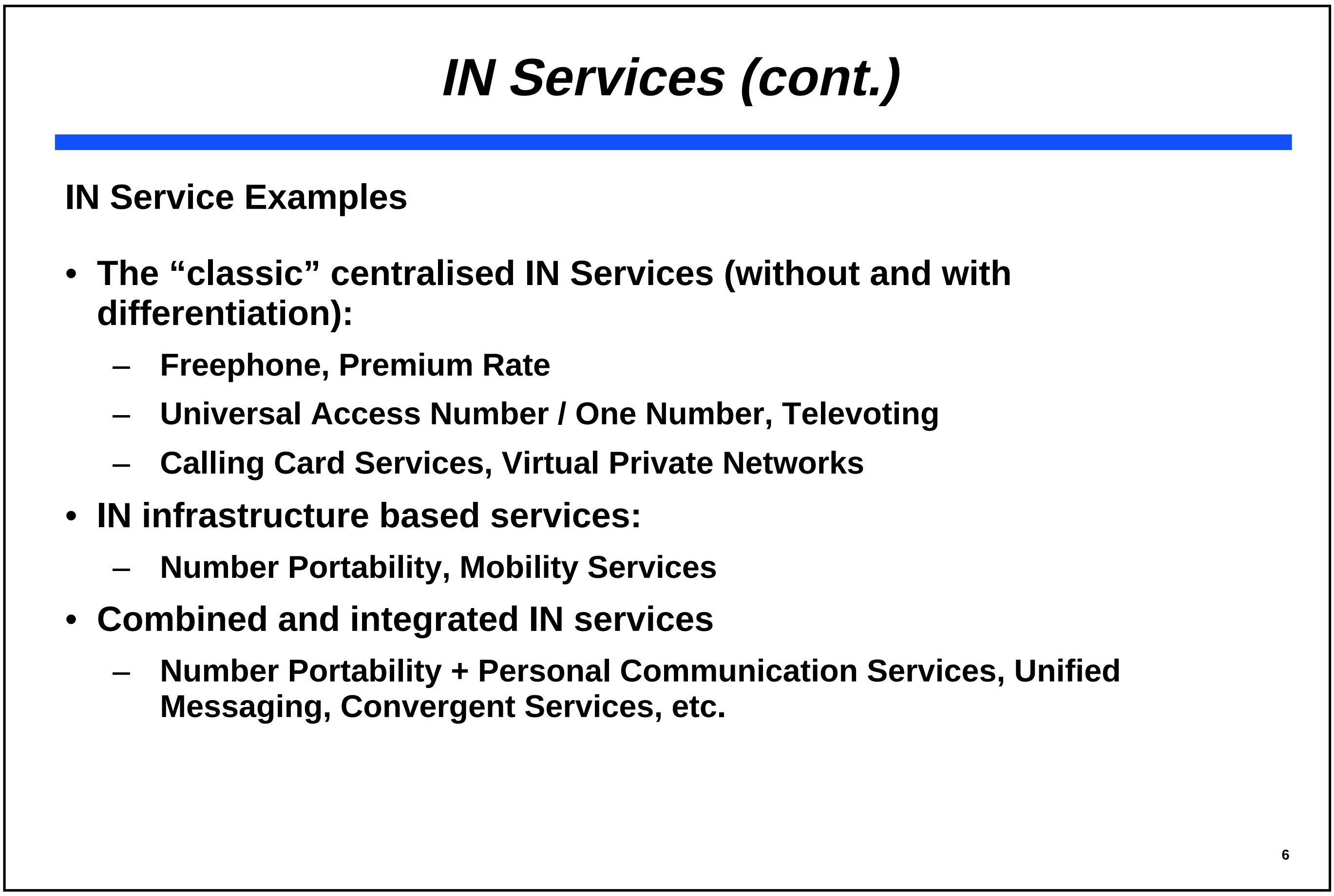
• advanced user interaction, and

• enhanced customer control (customer profile management) capabilities.

Although today's most promoted IN services are related to advanced telephony, i.e. voice applications, future IN services are considered to include also mobility, broadband and multimedia applications.

In order to support the provision of an open set of services, the IN defines an extensible set of generic and reusable service components, known as "Service Independent Building Blocks (SIBs)". These can be combined to higher level service elements, known as "Service Features", in order to construct rapidly new services within the scope of the service components' capabilities. This allows service providers to respond quickly to specific market demands.

Notice that a service user of an IN service does not see or care how the service is realized in the platform, but just how it is provided to him. That means that any of these envisaged IN services can also be implemented in a conventional network, but the IN-based service implementation represents the most economic.



IN Services (cont.)

IN Service Examples

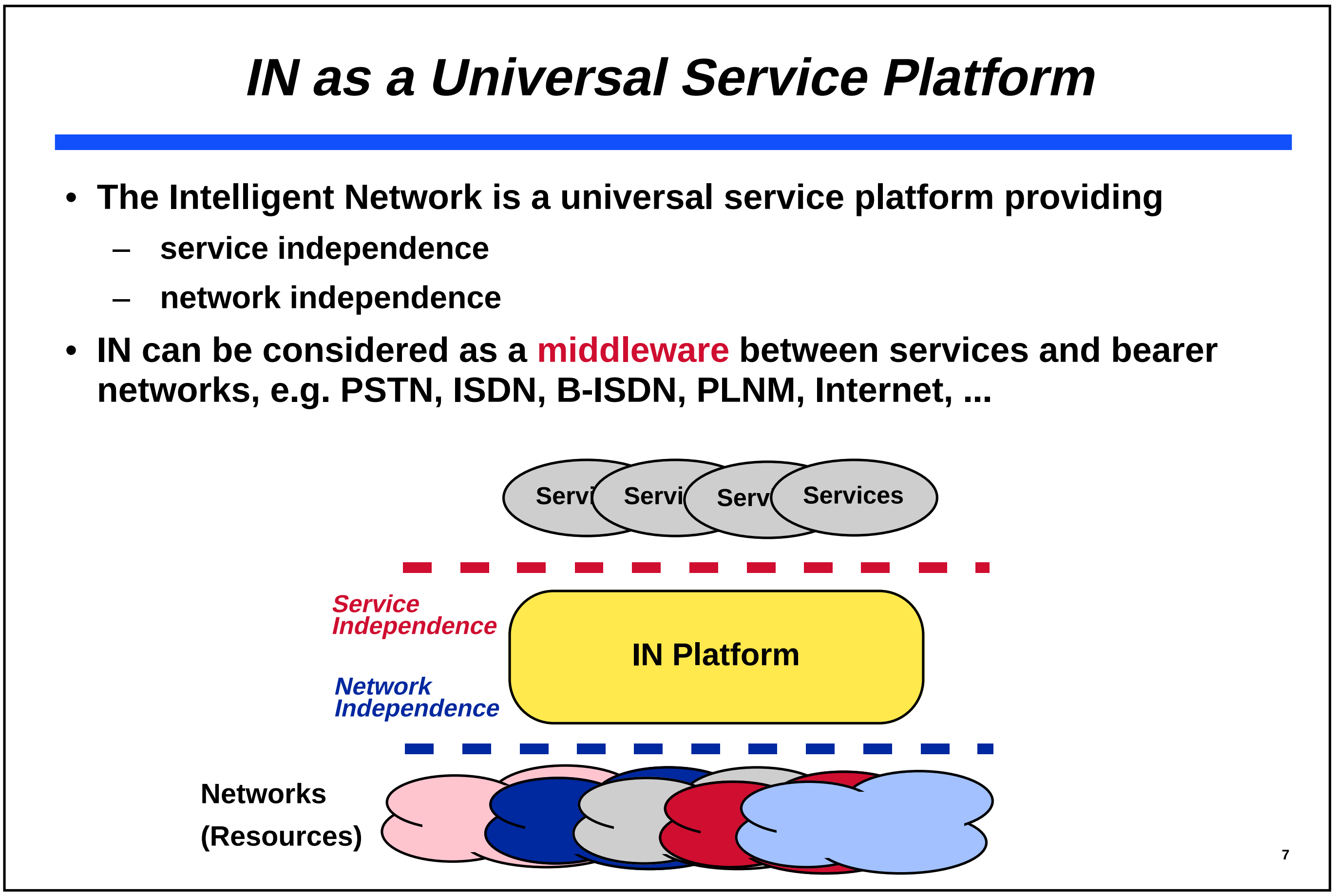
- The “classic” centralised IN Services (without and with differentiation):
  - Freephone, Premium Rate
  - Universal Access Number / One Number, Televoting
  - Calling Card Services, Virtual Private Networks
- IN infrastructure based services:
  - Number Portability, Mobility Services
- Combined and integrated IN services
  - Number Portability + Personal Communication Services, Unified Messaging, Convergent Services, etc.

Examples for IN services defined and implemented by means of reusable service components are Freephone, Televoting, Premium Rate, Card Calling, Virtual Private Networks, Universal Personal Telecommunications and many more. A detailed description of these is given later.

However, it has to be noted, that these are the “Classic” standard IN services, representing the 1st generation of IN services, which offer limited opportunities for differentiation between operators. Therefore the next generation of “fully differentiated” IN services allowed for higher sorphistication of the services by a rising share of pre- and after sales services and changing marketing approaches of operators.

In addition, some new IN service emerged on the basis of the IN architecture, such as Number Portability and financial calling, which provide limited opportunities for value generation but which where dectated by regulators or have a potential of 100% penetration.

Finally, combined and integrated IN services represent the most recent and most promising category of IN services, which provide significant opportunities for operators, service providers and subscribers. Examples are Personal Communication Services, Unified Messaging and convergent services.



IN as a Universal Service Platform

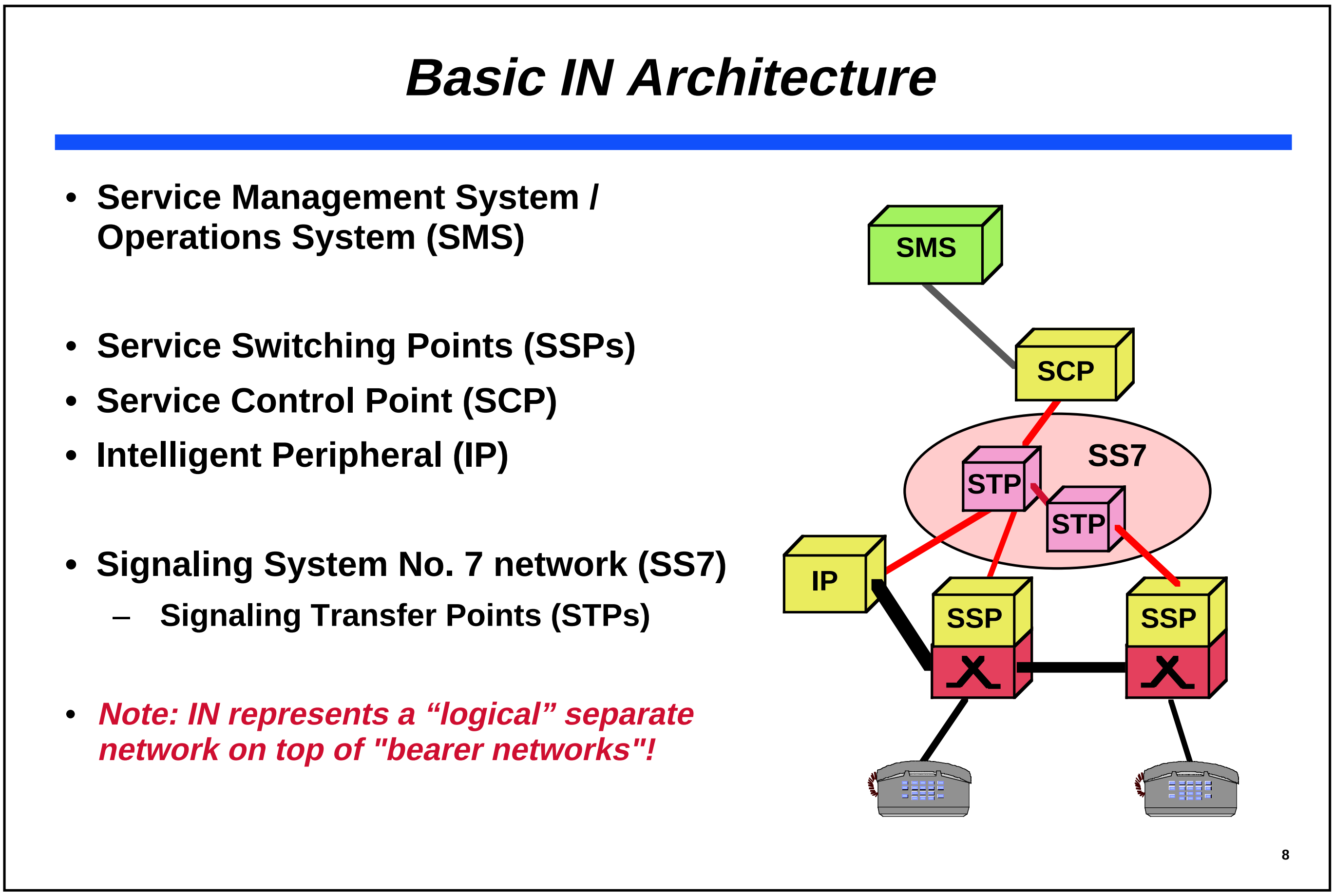
- **The Intelligent Network is a universal service platform providing**
  - **service independence**
  - **network independence**
- **IN can be considered as a middleware between services and bearer networks, e.g. PSTN, ISDN, B-ISDN, PLNM, Internet, ...**

7

In general, the IN can be considered as an additional (network) layer on top of any bearer network, such as Public Switched Telephone Network (PSTN), Integrated Services Digital Network (ISDN), Broadband-ISDN (B-ISDN), Public Land Mobile Networks (PLMNs) and recently the Internet. In this respect the IN provides a service-oriented network architecture, which separates in principle service control functions from service switching functions, with typically both types of functions being implemented in different physical equipment. This is supported by a clear definition of the relationships between these functions, thus providing for network and vendor independence. Due to this separation of functions it is possible to introduce new services rapidly without having to change the functionality of the switches. This means the IN architecture allows operators to deploy and provide new services more quickly than in traditional switch-based service implementations, which is essential in a liberalizing telecommunications market of ever increasing competition.

However, another major target of IN is service independence. During the development of many advanced telecommunication services it became clear, that all of these services contain similar functionality, i.e. are based on a set of "service components". Hence the idea was to identify a generic sets of reusable service components, which could be (re)used for the construction of a new service. Examples for such service components are authentication, screen, user interaction, number translation, charge, etc. In this context there is often the notion of an IN programming interface which can be used for easy service creation. What is meant is that a service designer can make use of these service components, by combining them in order to "implement" a new service. These service components are referred to as "Service Independent Building Blocks (SIBs)". The resulting "service (logic) program" could be loaded into the network, i.e. into one or more Service Control Point (SCPs), and the new service is instantly available.

Putting these aspects together it can be recognized, that the IN is more than just a new network architecture, since the IN aims to support both service and network independence.



Basic IN Architecture

- **Service Management System / Operations System (SMS)**
- **Service Switching Points (SSPs)**
- **Service Control Point (SCP)**
- **Intelligent Peripheral (IP)**
- **Signaling System No. 7 network (SS7)**
  – **Signaling Transfer Points (STPs)**
- ***Note: IN represents a "logical" separate network on top of "bearer networks"!***

SMS
SCP
SS7
STP
STP
IP
SSP
SSP

8

Service Switching Points (SSPs) are stored-program control switches, either local exchanges or access tandem exchanges, that are able to interface with the SS7 signaling network. These switches also contain a limited "service (access) logic" required to suspend calls that require special handling. The Service Switching Point recognizes IN service calls and routes the corresponding queries to the Service Control Point via the SS7 network, which consists of Signaling Transfer Points (STPs). Service Control Point commands will be used by the Service Switching Point to further process the call.

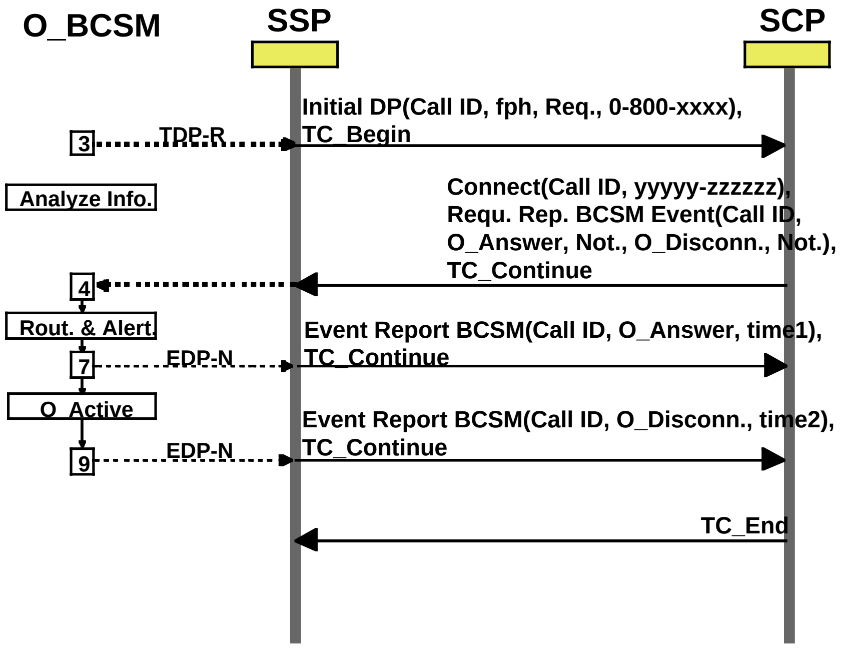
The Service Control Point (SCP) is typically an on-line, fault-tolerant, transaction-processing data base which provides call handling information as response to Service Switching Point queries. Service Control Points are high-capacity systems handling several hundred transactions per second or more than 100.000 calls an hour. It is a high capacity system with a response time requirement of less than half a second, and an availability of less than three-minutes-per-year downtime for a mated Service Control Point pair. Service Control Points are designed to support multi-service operation.

In order to manage IN platforms a Service Management System (SMS) containing the reference service databases is required. Supervision, remote operations and maintenance of Service Control Points, and (coordinated) software downloading are part of the Service Management System features. The Service Management System is integrated in an Operations Support System which supports network operation, administration and maintenance functions, and normally resides in a commercial host computer.

An additional Intelligent Peripheral (IP) may be connected to an Service Switching Point, providing enhanced services/functions, such as announcements, speech synthesizing, speech recognition, etc. under the control of a Service Switching Point or Service Control Point. The motivation for the introduction of this network element is an economical aspect, because it might be better for several users to share an Intelligent Peripheral, when the capabilities of the Intelligent Peripheral are too expensive to be implemented at all Service Switching Points.

Freephone Service Processing Example

Information Flows between SSP and SCP



O_BCSM

SSP

SCP

Initial DP(Call ID, fph, Req., 0-800-xxxx), TC_Begin

3 TDP-R

Analyze Info.

Connect(Call ID, yyyyy-zzzzzz), Requ. Rep. BCSM Event(Call ID, O_Answer, Not., O_Disconn., Not.), TC_Continue

4

Rout. & Alert.

Event Report BCSM(Call ID, O_Answer, time1), TC_Continue

7 EDP-N

O_Active

Event Report BCSM(Call ID, O_Disconn., time2), TC_Continue

9 EDP-N

TC_End

Freephone Example : SSF - SCF Interworking

The usage of the Freephone service commences with the originating party going off-hook and dialling a freephone number. On reaching Detection Point 2 (DP 2) "Collected Information" in the originating BCSM, which has been armed in order to indicate that a freephone number is dialled by the originating party, a Trigger Detection Point-Request (TDP-R) is detected. This TDP-R results in sending an Initial DP IF to the SCF and suspending call processing.

On receiving the Initial DP-IF the SCF uses the IE Service Key to determine the appropriate steps. In the case of the Freephone-service this would be to translate the information contained in the IE Dialled Digits into the according directory number and to adjust the charging, i.e. to charge the called party for this call by consulting the SDF (not indicated). The determined directory number is send with an Connect-IF to the SSF.

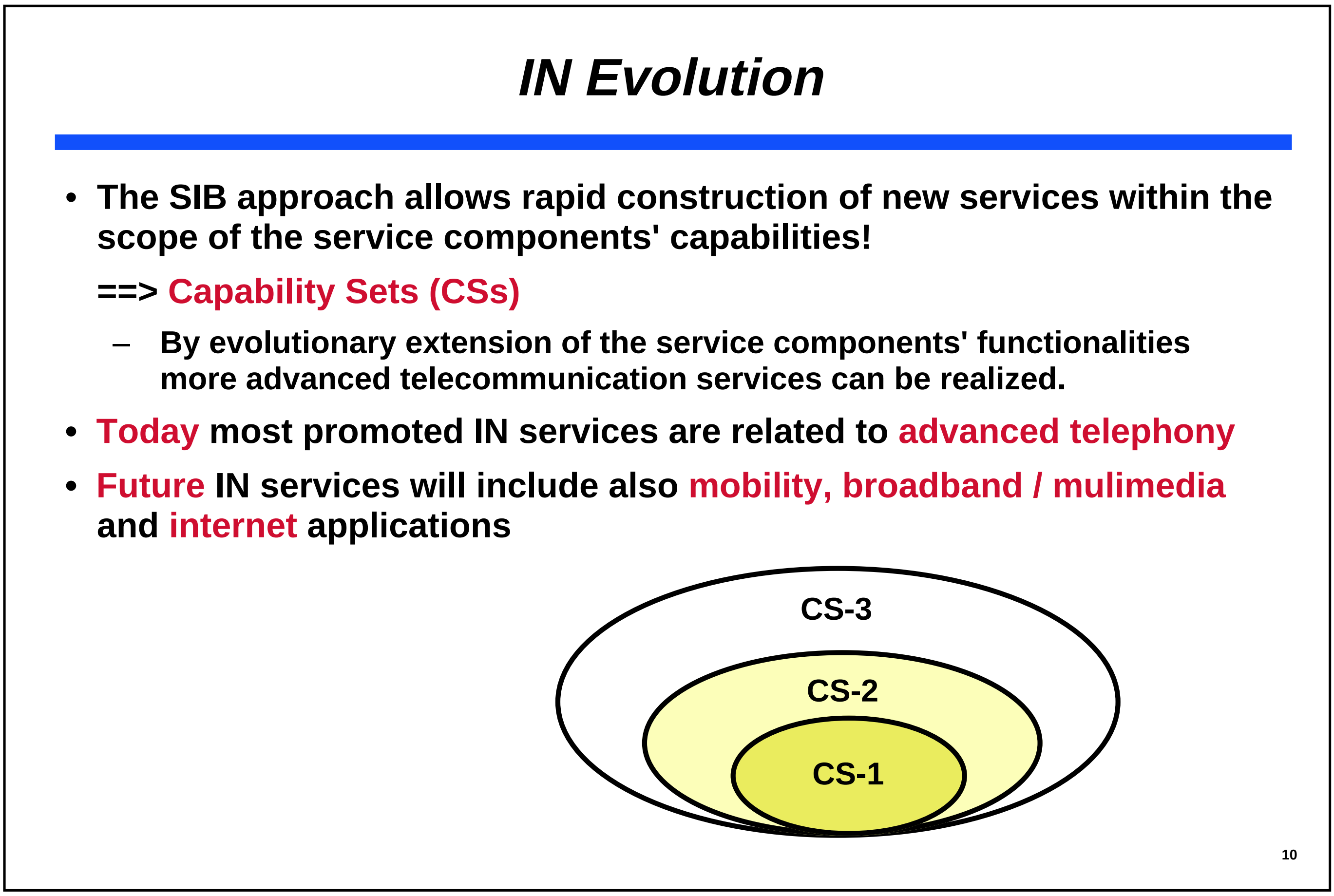
Together with the Connect-IF a Request Report BCSM Event-IF is sent to the SSF. In this scenario the IF Request Report BCSM Event is used to arm DP 7 "O_Answer" and DP 9 "O_Disconnect" dynamically.

In the SSF the desired DPs are both armed as Event Detection Points-Notification (EDP-N), i.e. their Monitor Modes take on the value „Notify & Continue”.

On receiving this IF from the SCF, the SSF resumes call processing in PIC 4 "Routing & Alerting". The provided directory number is used to route the call to the terminating side. In case DP 7 "O_Answer" and DP 9 "O_Disconnect" are reached throughout call processing the SSF sends a Event Report BCSM-IF to the SCF. In both cases call processing is not suspended, since the mentioned DPs have been armed as EDP-Ns.

The SCF will use the directory number of the called user and the information contained in the Event Report BCSM-IF to determine the charge for this call after its completion.

Finally, a TC_END-message is sent from the SCF to the SSF. This message indicates that the relationship between the SCF and the SSF is terminated.



IN Evolution

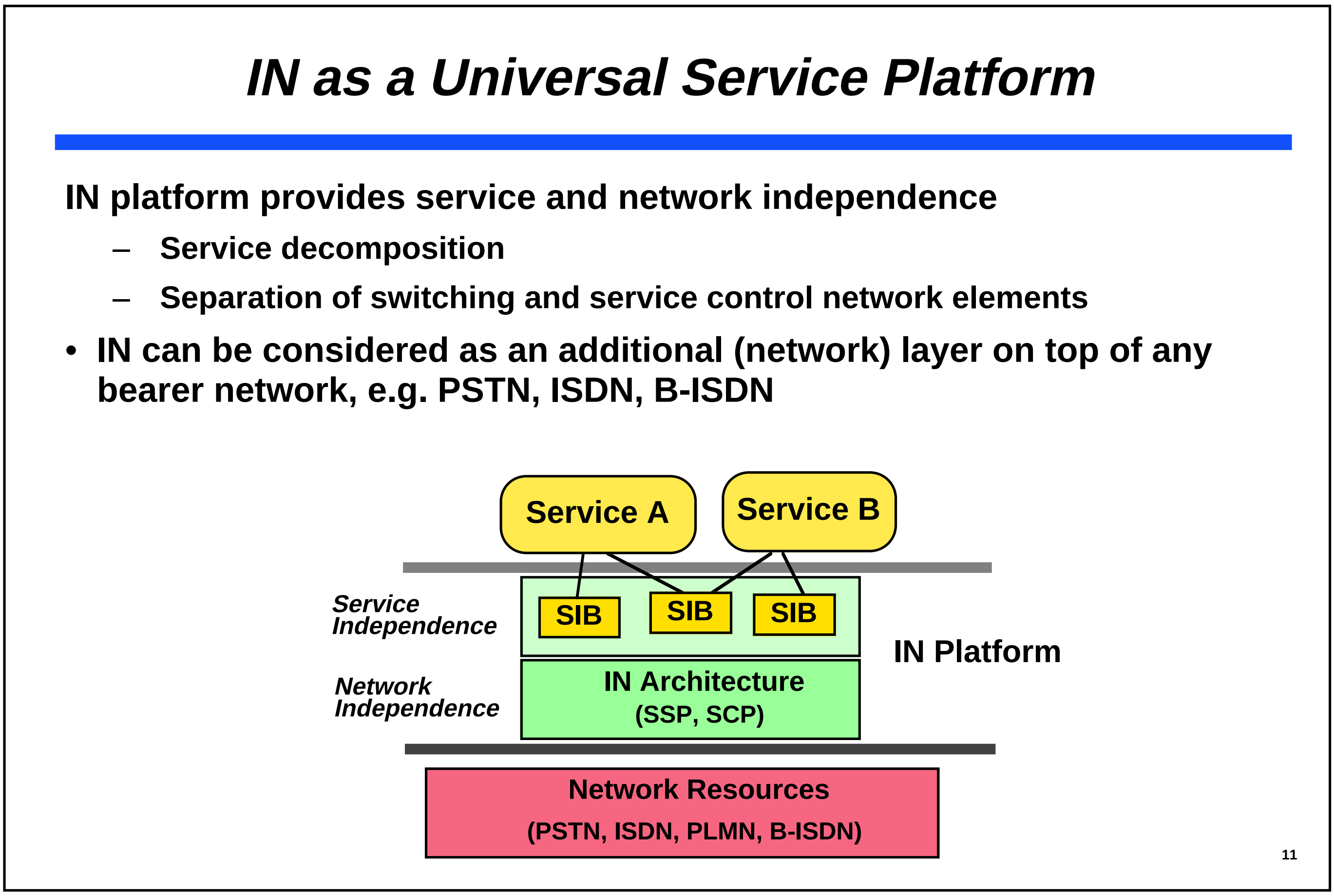
- **The SIB approach allows rapid construction of new services within the scope of the service components' capabilities!**

  ==> Capability Sets (CSs)

  - **By evolutionary extension of the service components' functionalities more advanced telecommunication services can be realized.**

- **Today most promoted IN services are related to advanced telephony**
- **Future IN services will include also mobility, broadband / mulimedia and internet applications**

By evolutionary extension of the service component's functionalities (and corresponding enhancement of the IN architecture) more advanced telecommunication services can be realized, while still supporting existing services. Although today most promoted IN services are related to advanced telephony, i.e. voice applications, future IN services are considered to include also mobility, broadband and multimedia applications. That is, the IN concept is considered to provide the necessary openness in order to enable the realization of future services.



IN as a Universal Service Platform

IN platform provides service and network independence

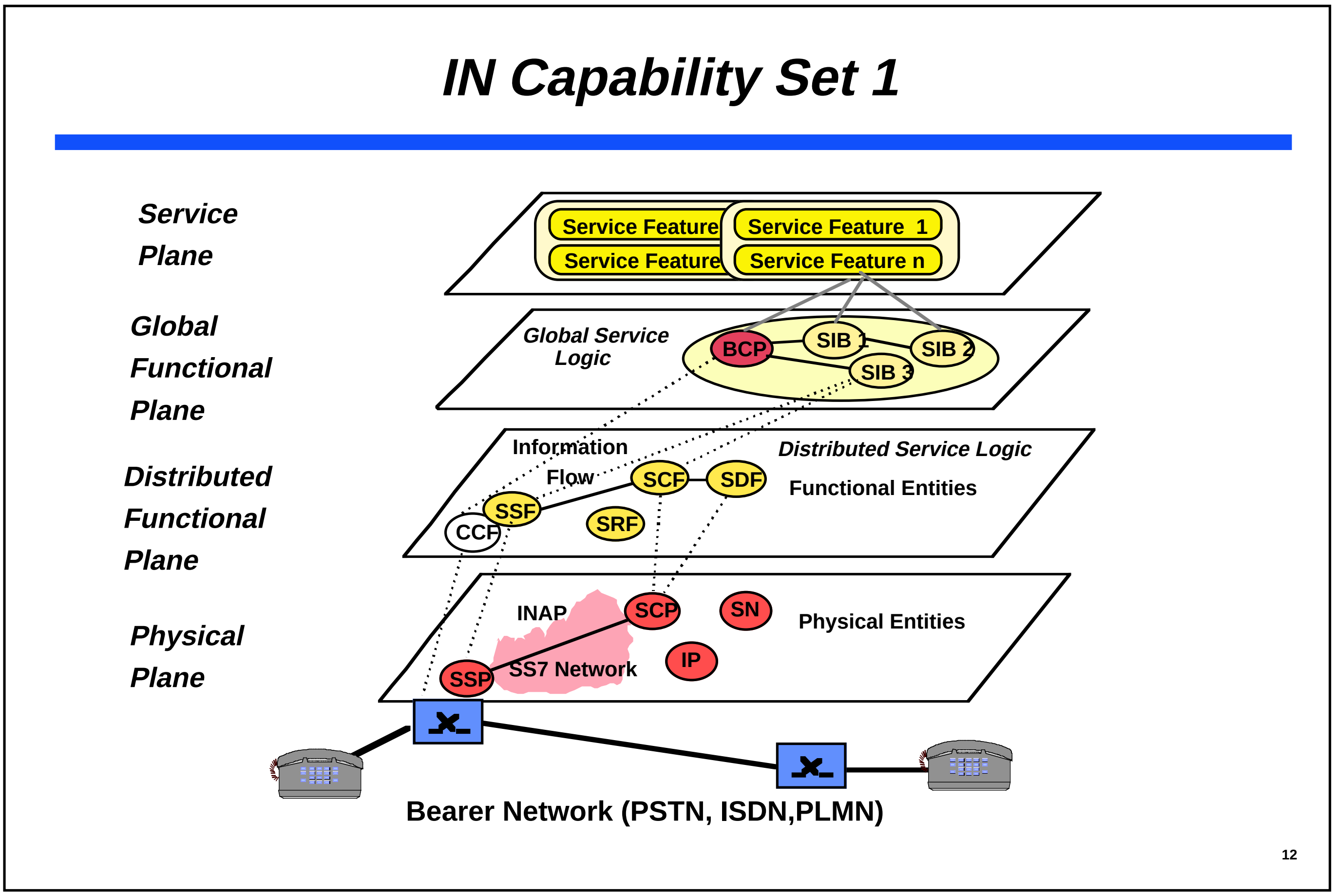
- – Service decomposition
- – Separation of switching and service control network elements

- IN can be considered as an additional (network) layer on top of any bearer network, e.g. PSTN, ISDN, B-ISDN

In general, the IN can be considered as an additional (network) layer on top of any bearer network, such as Public Switched Telephone Network (PSTN), Integrated Services Digital Network (ISDN), or Broadband-ISDN (B-ISDN). In this respect the IN provides a service-oriented network architecture, which separates in principle service control functions from service switching functions, with typically both types of functions being implemented in different physical equipment. This is supported by a clear definition of the relationships between these functions, thus providing for network and vendor independence. Due to this separation of functions it is possible to introduce new services rapidly without having to change the functionality of the switches. This means the IN architecture allows operators to deploy and provide new services more quickly, which is essential in a liberalizing market of ever increasing competition.

However, another major target of IN is service independence. During the development of many advanced telecommunication services it became clear, that all of these services contain similar functionality, i.e. are based on a set of "service components". Hence the idea was to identify a generic sets of reusable service components, which could be (re)used for the construction of a new service. Examples for such service components are authentication, screen, user interaction, number translation, charge, etc. In this context there is often the notion of an IN programming interface which can be used for easy service creation. What is meant is that a service designer can make use of these service components, by combining them in order to "implement" a new service. These service components are referred to as "Service Independent Building Blocks (SIBs)". The resulting "service (logic) program" could be loaded into the network, i.e. into one or more Service Control Point (SCPs), and the new service is instantly available.

Putting these aspects together it can be recognized, that the IN is more than just a new network architecture, since the IN aims to support both service and network independence.



IN Capability Set 1

IN CS-1 defined within the ITU Recommendations Q.1210-Q.1219 represents the first stage of a standardized IN, supporting a first range of IN services on top of different bearer networks.

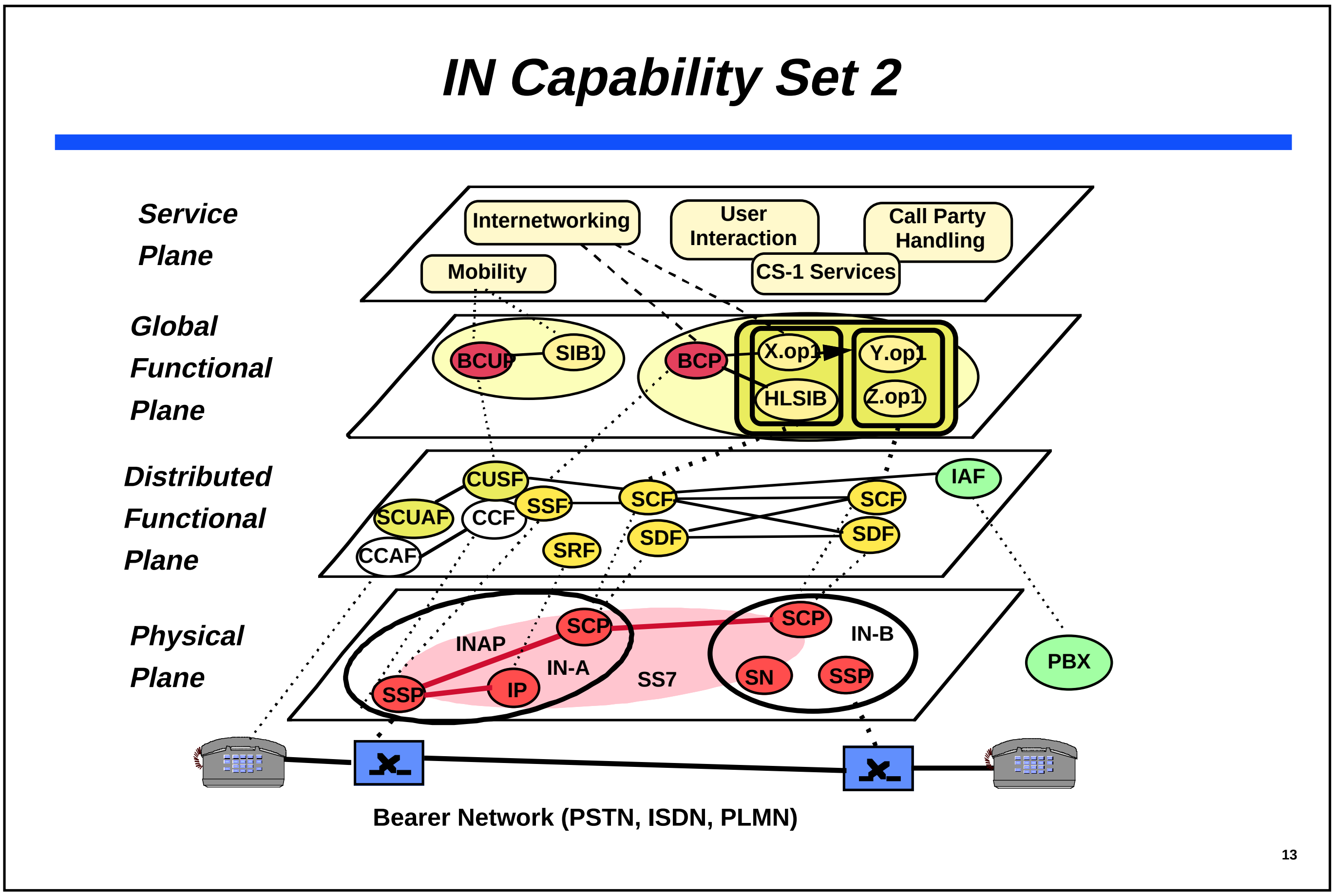
CS-1 is based on the IN Conceptual Model (INCM), that means that a set of IN CS-1 benchmark services and service features has been defined, which has guided the definition of the lower INCM planes for CS-1.

In the Global Functional Plane a set of 15 appropriate SIBs required for the CS-1 benchmark services has been defined, including a Basic Call Process (BCP) SIB, which models the processing of a basic call within a switch. The BCP SIB interacts at dedicated points in call processing with the SIB-based IN (global) service logic.

The Distributed Functional Architecture defines a corresponding functional architecture supporting the distributed realization of SIBs by appropriate information flows between the defined functional entities and functional entity actions. A Basic Call State Model, which is a projection of the BCP SIB onto the DFP, identifies the detection points for IN service logic invocation.

The Physical Plane defines the possible grouping of functional entities within physical elements and in particular the physical IN application protocol used for the communication between the IN network elements.

CS-1 excludes many aspects of INs, particularly service creation and management, as well as IN interworking (for global IN services).



IN Capability Set 2

Bearer Network (PSTN, ISDN, PLMN)

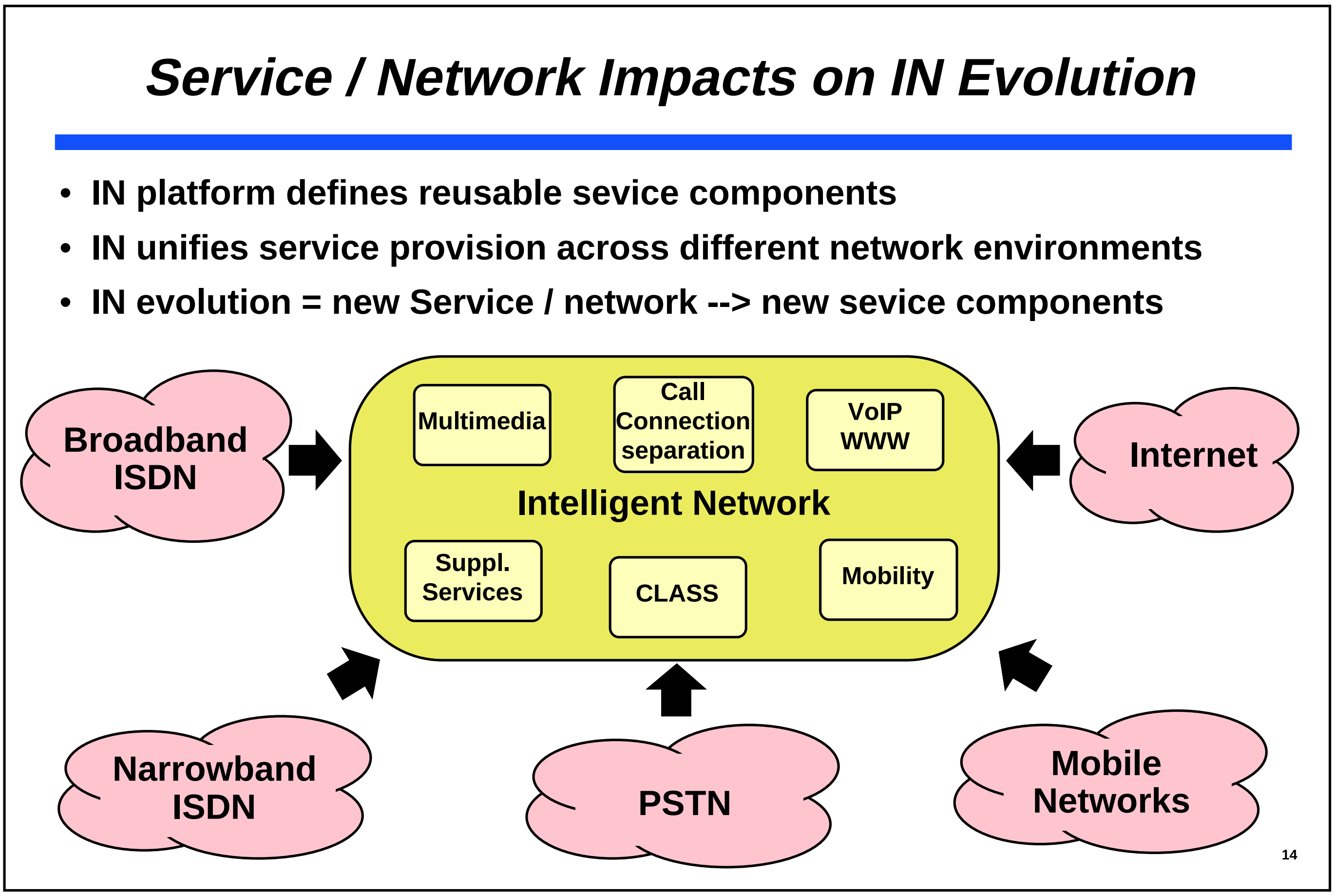
IN CS-2 defined within the ITU Recommendations Q.1220-Q.1229 represents an extension of IN CS-1 removing the basic limitations. Also CS-2 is based on the INCM.

That is the development of the CS-2 IN architecture started with the identification of envisaged service capabilities in the Service Plane. The basic service enhancements comprise IN internetwork services (e.g. pan-European IN services requiring IN interconnection), mobility service and corresponding call unrelated services (e.g. user registration), multiparty services requiring multi party handling and enhanced user interaction capabilities.

This enhanced service scope had impacts on all lower planes of the INCM. In the Global Functional Plane new SIBs have been identified in order to support the implementation of the new features and the Basic Call Process SIB has been enhanced. Particularly a new Basic Call unrelated Process (BCUP) SIB has been introduced. Furthermore SIB programming has been enhanced by new concepts, such as the introduction of service processes, high level SIBs, SIB operations, etc.

In the Distributed Functional Plane the basic call state models have been appropriately enhanced in order to enable the interaction with external IN service logic. Also a Basic Call Unrelated State Model has been defined for non-call related services. Additionally, the identified functional entities have been enhanced (e.g. for the purpose of IN internetworking) as well as new entities have been defined for call unrelated interactions. Also the SRF has been enhanced in order to enable enhanced user interaction capabilities. Consequently new information flows have been defined.

In the Physical Plane the IN application protocol has been enhanced in order to accommodate the new information flows. Additionally, new physical entities have been identified.



Service / Network Impacts on IN Evolution

- IN platform defines reusable sevice components
- IN unifies service provision across different network environments
- IN evolution = new Service / network --> new sevice components

Today different network environments, such as Public Switched Telephone Networks (PSTNs), Narrowband and Broadband Integrated Digital Services Networks (ISDNs), Public Land Mobile Networks (PLMNs), and even the Internet, define their own range of services. These services, covering sometimes similar capabilities but sometimes also quite diverging capabilities, are realized in specific ways.

This results in different service environments, which makes service development quite complex. However, the IN is aiming to integrate these different ways of service provision by definning a uniform service middleware, applicable to any underlying network technology.

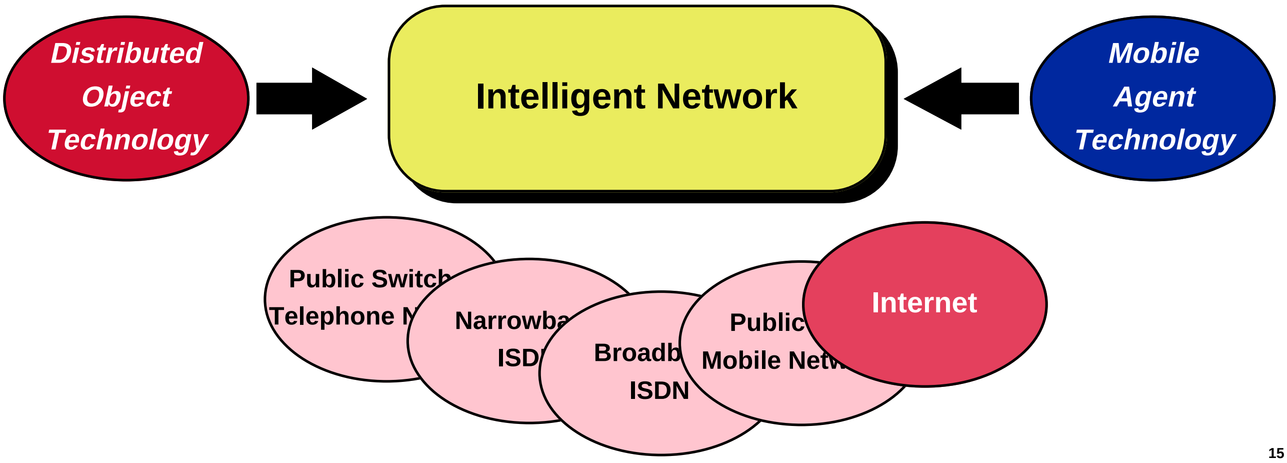
However, the IN is aiming to integrate these different ways of service provision by definning a uniform service middleware, applicable to any underlying network technology. The basic idea is to use the same service modelling principles and to define new service features/building blocks (ideally by reusing some already existing ones) as required by a particular network environment.

As depicted above, in addition to the provision of advanced telephony service capabilities for PSTN and ISDN, specific mobility support capabilities have to be defined in order to make use of IN capabilities within PLMNs. B-ISDN requires additional capabilities required for the provision of multimedia, multiparty services. Furthermore the usage of IN for Internet requires interworking capabilities to support advanced email services and web applications.

We will address these aspects within this tutorial.

Technology Impacts on IN Evolution

- **IN is a Middleware, introducing advanced Information Technology (i.e. distributed computing) into telecom environments**
- **IN evolution is also driven by new Computing / Software Technologies**
- **In contrast to centralisation today the keyword is distribution**



We have jut seen that IN evolution is driven by the requirement to provide new service capabilities for specific network environments.

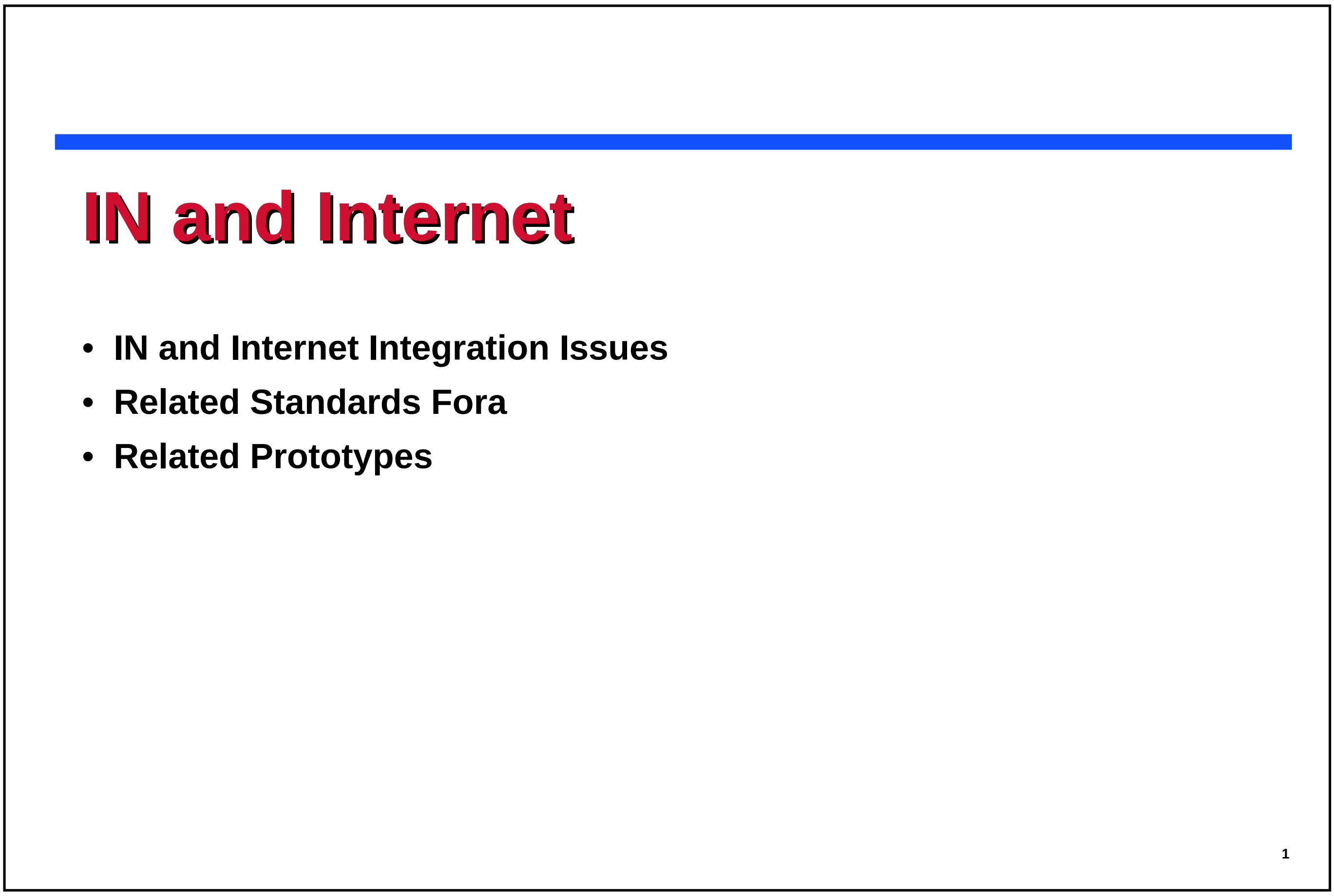
However, another fundamental principle of IN is to take advantage of new computing technologies. This has led to the introduction of computing nodes providing the service logic into the telecommunications network interacting via Remote Procedure Calls with the switches.

Today two basic new computing paradigms are gaining momentum:

- distributed object technology and
- intelligent mobile agent technology.

These technologies promise to provide more flexibility in software development and distribution and thereby may be more adequate for developing solutions for the emerging open telecommunications service market.

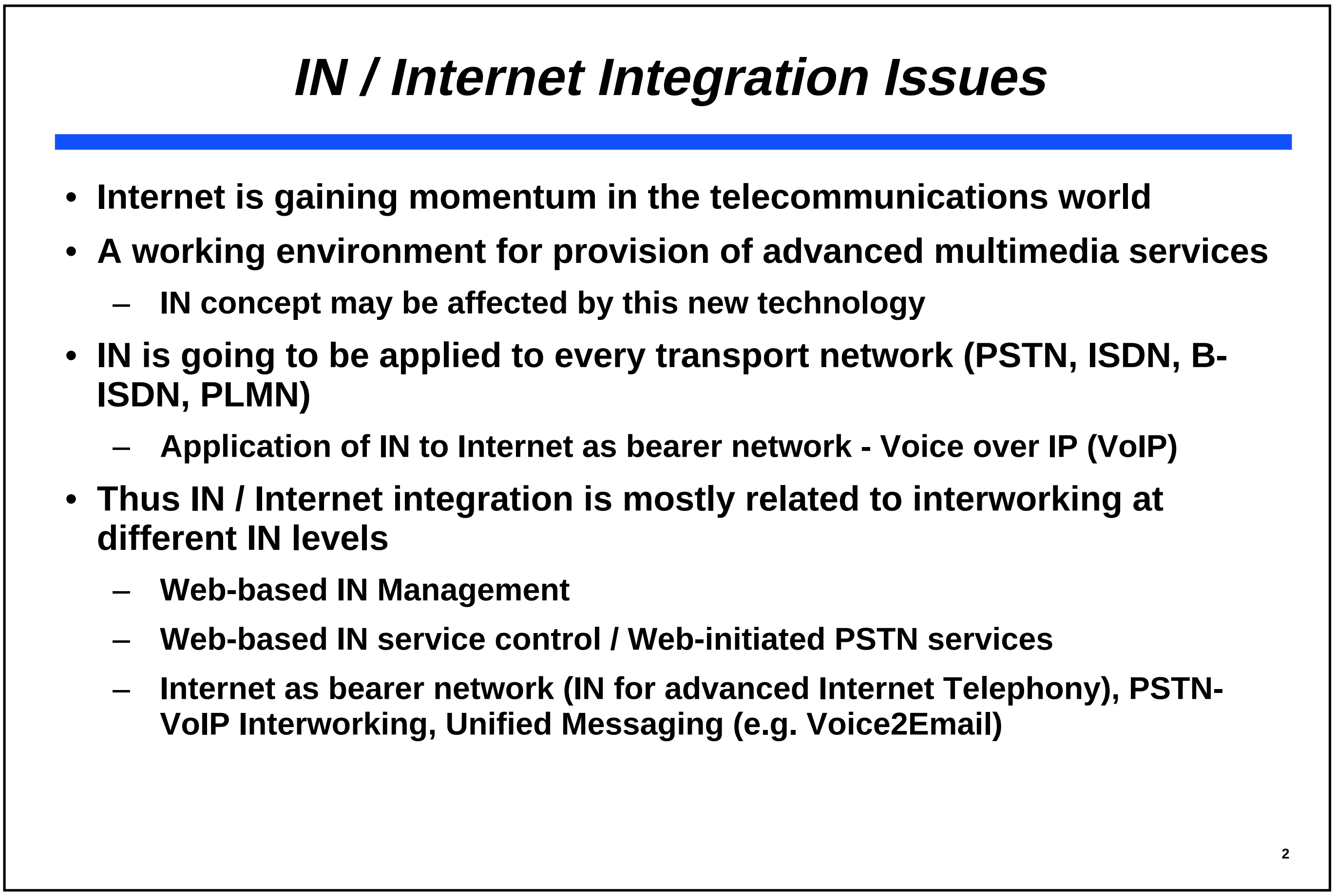
Within this tutorial we look also at these evolution trends.



IN and Internet

- IN and Internet Integration Issues
- Related Standards Fora
- Related Prototypes

This part addresses the emerging issue of IN/Internet integration, looking at the aspects how the Internet, i.e. the technology, could be used to enhance current IN environments (e.g. web-based IN management), and how on the other hand, considering the internet as just another transport network, how internet telephony environments can be enhanced by IN concepts, i.e. IN-based service for advanced internet telephony. Besides examples for the different IN/INternet integration issues we look briefly at the related standards fora and related Prototypes in this rapidly emerging field of research.



IN / Internet Integration Issues

- Internet is gaining momentum in the telecommunications world
- A working environment for provision of advanced multimedia services
  - IN concept may be affected by this new technology
- IN is going to be applied to every transport network (PSTN, ISDN, B-ISDN, PLMN)
  - Application of IN to Internet as bearer network - Voice over IP (VoIP)
- Thus IN / Internet integration is mostly related to interworking at different IN levels
  - Web-based IN Management
  - Web-based IN service control / Web-initiated PSTN services
  - Internet as bearer network (IN for advanced Internet Telephony), PSTN-VoIP Interworking, Unified Messaging (e.g. Voice2Email)

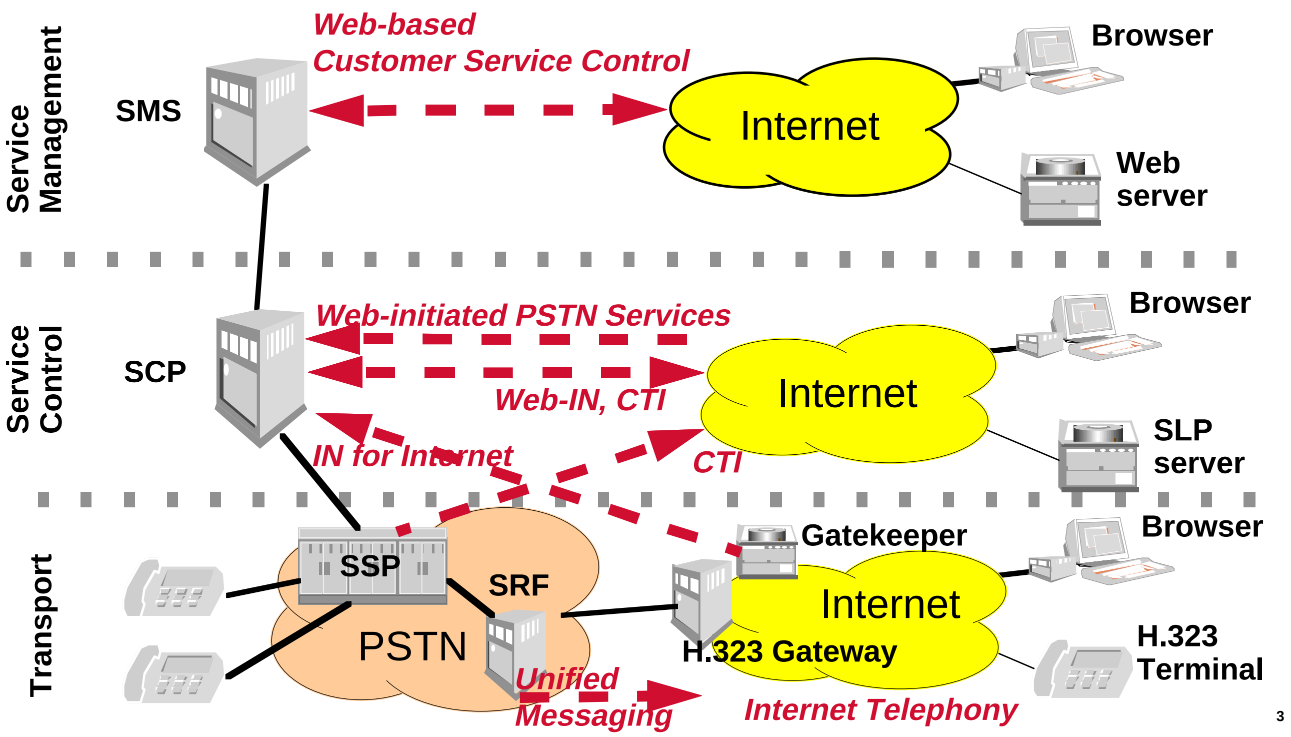
The internet is gaining momentum in the context of telecommunications. The reasons are the increasing deployment of Personal Computers and the fast pace of technological progress in internet service developments, which makes internetwing interesting for an increasing number of people. Today many people in the telecmmunications reserach labs are convinced, that the internet represents the biggest thread for the evolution of the existing telecommunications systems.

In the following we present the current existing views of IN / Internet integration taking into account the evolution of IN in terms of new technology impacts and new bearer network support. Particularly the internet stands for both domains as it represents an environment where advanced multimedia services are provided in a distributed nature (i.e. based on distributed intelligence in the network edges - Browsers and Web servers) to advanced end systems (multimedia PCs) as well as a connectionless (!!!) transport network for telephony (Voice over IP - VoIP).

The last issue is probably the most important driver for considering the IN as a means to provided advanced services (e.g. the same services available in PSTN/ISDN) in VoIP networks. This means that in the end, IN services can be provided uniformly across Switched Cirucuit Networks (SCNs) and VoIP networks.

IN the following we look at these issues in an integrated manner, taking into account the different levels of an IN infrastructure, i.e service management, service control and transport.

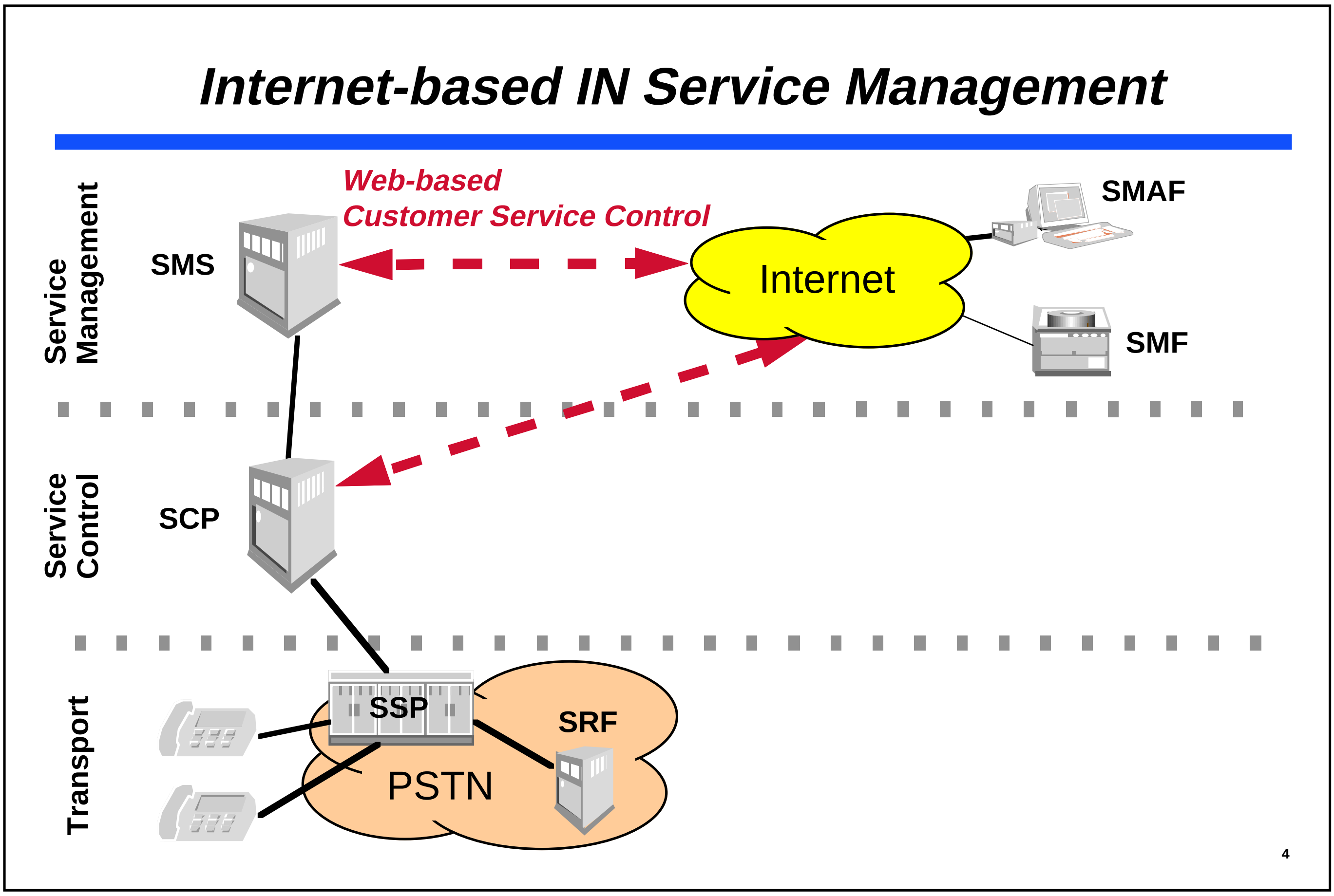
IN-Internet Integration Levels



Based on the previous considerations, three major levels of IN/Internet integration can be identified:

- Integration at service management level,
- Integration at service control level, and
- Integration at transport level, which means bearer service interworking/integration.

In the following slides we provide more details!



Internet-based IN Service Management

This slide depicts the mainstream trend towards using web browsers for accessing and manipulating SMS information. Particularly for Customer Service Control (Customer Profile Management) applications this is the ideal solution allowing service customers uniform and mobile access to their service data.

Java applets may be used to download the service management client software (GUI) to any web server.

The service processing is as follows:

1. The IN service customer modifies his Customer Profile via a web browser.

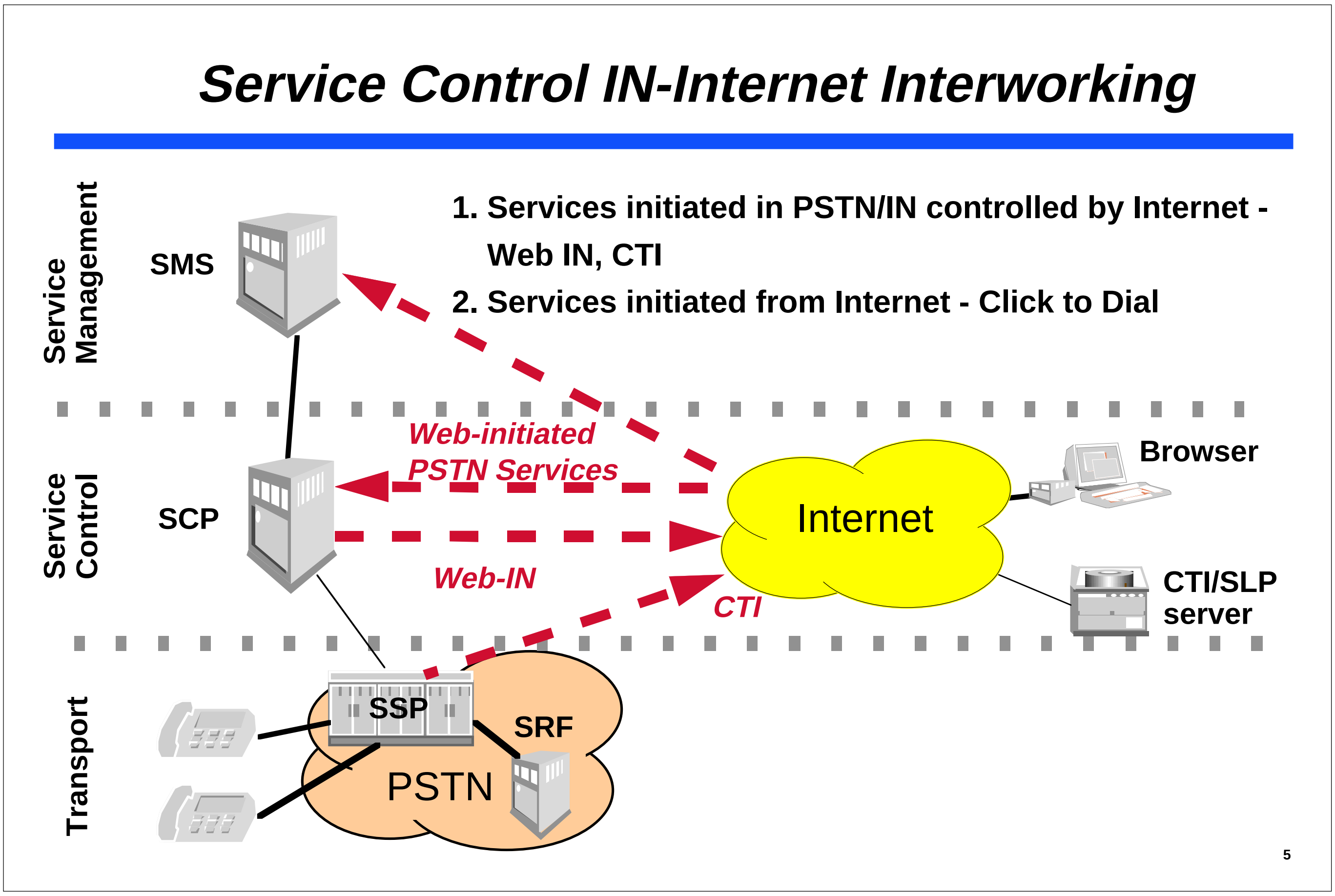
2. The modifications are processed in the webserver.

3. The webserver transmits the modifications to the IN SMS containing the master profile. Optionally the modifications may be directly transmitted to the SCP in case the webserver belongs to the service provider.

4. The SMS propagates the modifications to the SCP.

5. Subsequently users may call the called party.

6. The SCP handles the service requests in accord to the modified Customer Profile.



Service Control IN-Internet Interworking

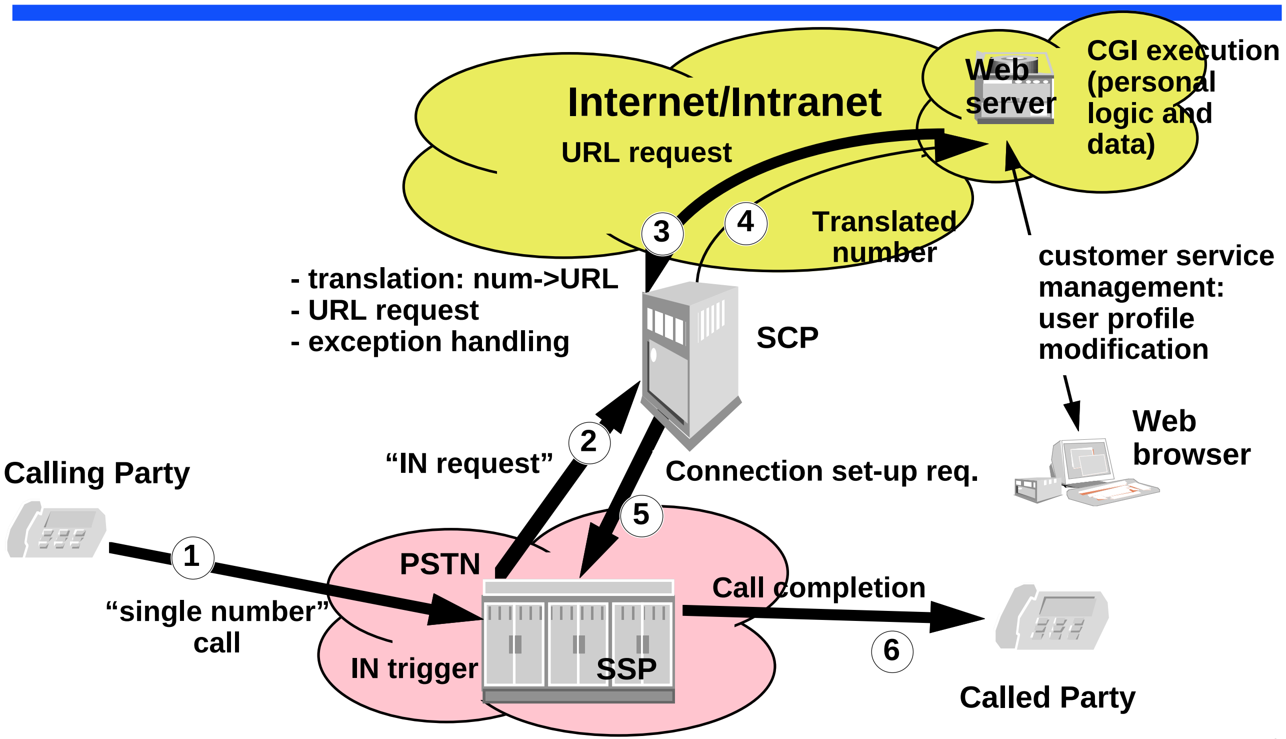
1. Services initiated in PSTN/IN controlled by Internet - Web IN, CTI
2. Services initiated from Internet - Click to Dial

At the control level two options exist:

1. Placing the service control logic/data into the Internet domain, which means that during service control an SCP or even SSP may consult a web server hosting the service logic (known as WebIN concept).

2. Using IN service control capabilities (i.e. the Initiate Call INAP operation) to establish bearer connections, iniated from the internet (known as Web-initiated services as defined by the IETF PINT working group).

Example WebIN



Internet/Intranet

URL request

Web server

CGI execution (personal logic and data)

3

4

Translated number

customer service management: user profile modification

- translation: num->URL
- URL request
- exception handling

SCP

Web browser

Calling Party

“IN request”

2

Connection set-up req.

5

1

PSTN

“single number” call

IN trigger

SSP

Call completion

6

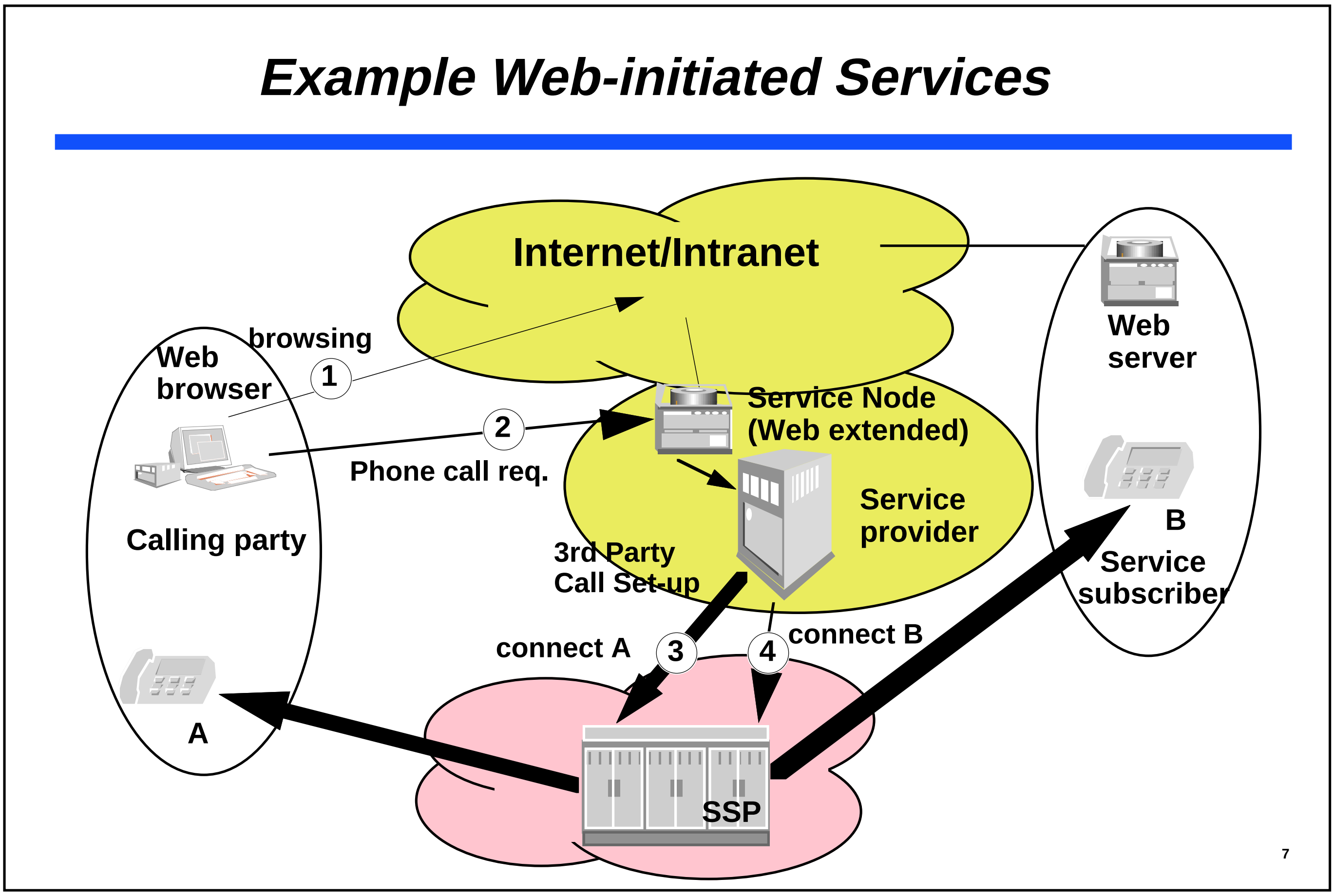
Called Party

6

This slide depicts the signalling required for implementing a “Single Number” service provided by a webserver (as CGI script or defined by a dedicated Call Processing Syntax as currently under development in the IETF IPTEL working group).

This means that potential customers/users can easily define their own services and modify these in an Internet style.

The service program hosted on a web server will be consulted by an SCP (3) for corresponding calls originating in the PSTN recognised by an IN SSP (1 + 2). Baseed on the result of the query (4) the SCP can use the obtain physical number to advice the SSP to set up the right connection (5 + 6).



Example Web-initiated Services

Internet/Intranet

Web server

Web browser

browsing

1

2

Phone call req.

Calling party

Service Node (Web extended)

Service provider

3rd Party Call Set-up

B

Service subscriber

connect A

3

4

connect B

A

SSP

7

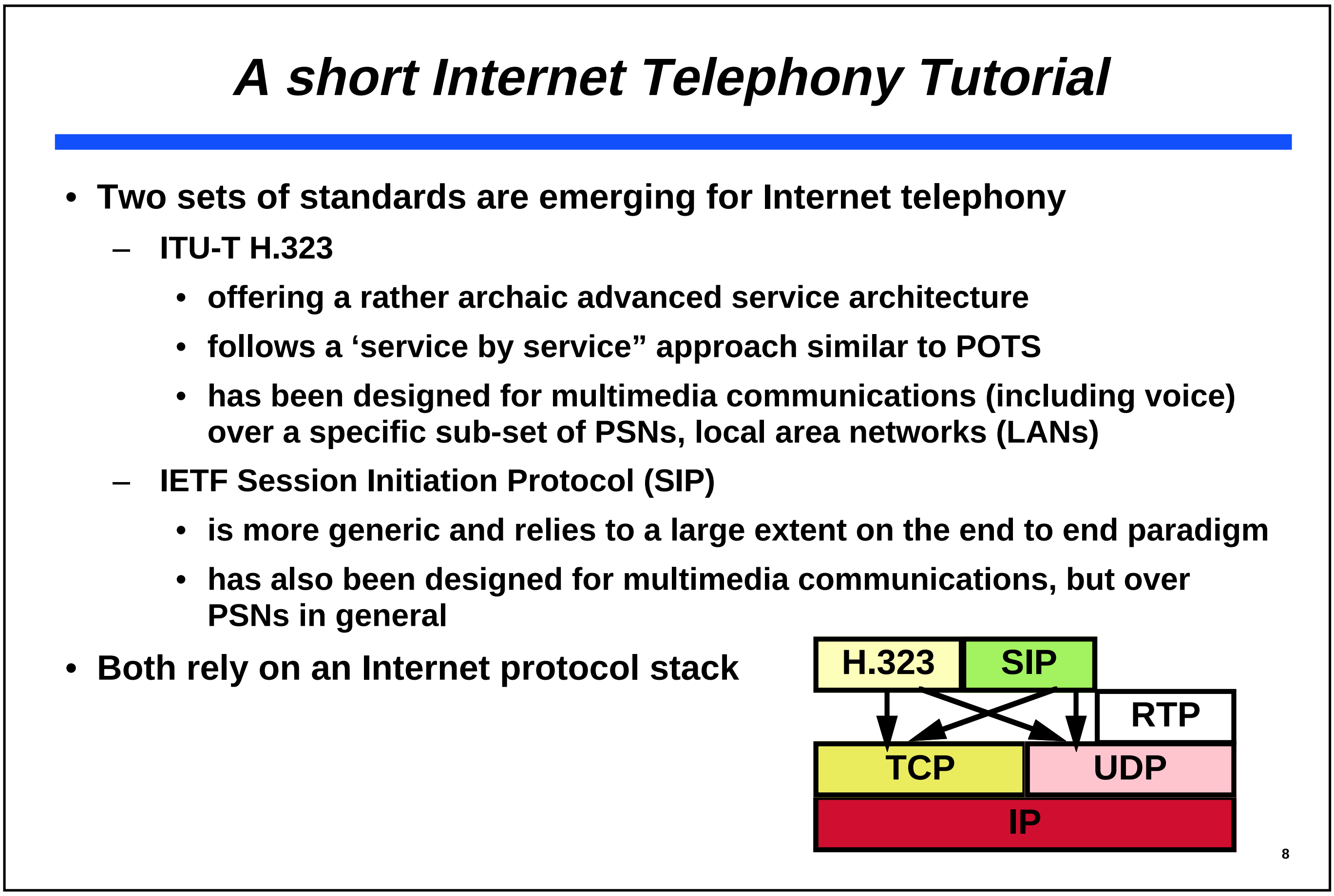
This slide illustrates a typical web-iniated service (click-to-dial), as defined by the IETF PINT WG, enable the initiation of telephone calls via a web browser. However, the call will only be established inside the PSTN (i.e. no internet telephony is envisaged so far).

Service processing is as follows:

1. The user A clicks a specific button on a web page to be called at his telephone (the number has to be provided somehow, e.g. by entering it into a web form.

2. The webserver contacts an SCP via a specific interface in order to initiate the establishment of an end-to-end connection between the users phone and the call party related to the web page, e.g. a call center indicated as service subscriber B.

3. The SCP instructs an SSP to realise the connection via the iniate call operation.



A short Internet Telephony Tutorial

- Two sets of standards are emerging for Internet telephony
  - ITU-T H.323
    - offering a rather archaic advanced service architecture
    - follows a ‘service by service” approach similar to POTS
    - has been designed for multimedia communications (including voice) over a specific sub-set of PSNs, local area networks (LANs)
  - IETF Session Initiation Protocol (SIP)
    - is more generic and relies to a large extent on the end to end paradigm
    - has also been designed for multimedia communications, but over PSNs in general
- Both rely on an Internet protocol stack

8

Two sets of standards are emerging for Internet telephony: the first from ITU-T and the second from IETF. The ITU-T set offers a rather archaic advanced service architecture which follows a ‘service by service” approach reminiscent of classical telephony early days. The IETF architecture is more generic and relies to a large extent on the end to end paradigm. It however has few pitfalls. We are indeed still far from the comprehensive architectures needed for creating and managing advanced services in Internet telephony.

H.323 recommendation is the principal ITU-T standard for Internet telephony. It is an umbrella standard which refers to many other standards. It was first released in 1996, then subsequently in 1998. A third release is planned for 1999. As customarily done in the industry, we use the term “H.323” to refer to the set of ITU-T standards for Internet tele-phony, including the H.323 recommendation itself. It is important to state that H.323 has been designed for multimedia communications (including voice) over a specific sub-set of PSNs, local area networks (LANS).

The Session Initiation Protocol (SIP) [3] is the principal IETF draft for Internet telephony. It has also been designed for multimedia communications, but over PSNs in general. It allows the establishment, modification and termination of multimedia calls. SIP relies on a host of Internet protocols including the real time protocol (RTP) for data transport. The same actually applies to H.323 as shown above which depicts a simplified Internet telephony protocol stack. In this course, as customarily done in the industry, we also use the term “SIP” to designate the set of IETF specifications for Internet telephony, including the SIP draft itself..

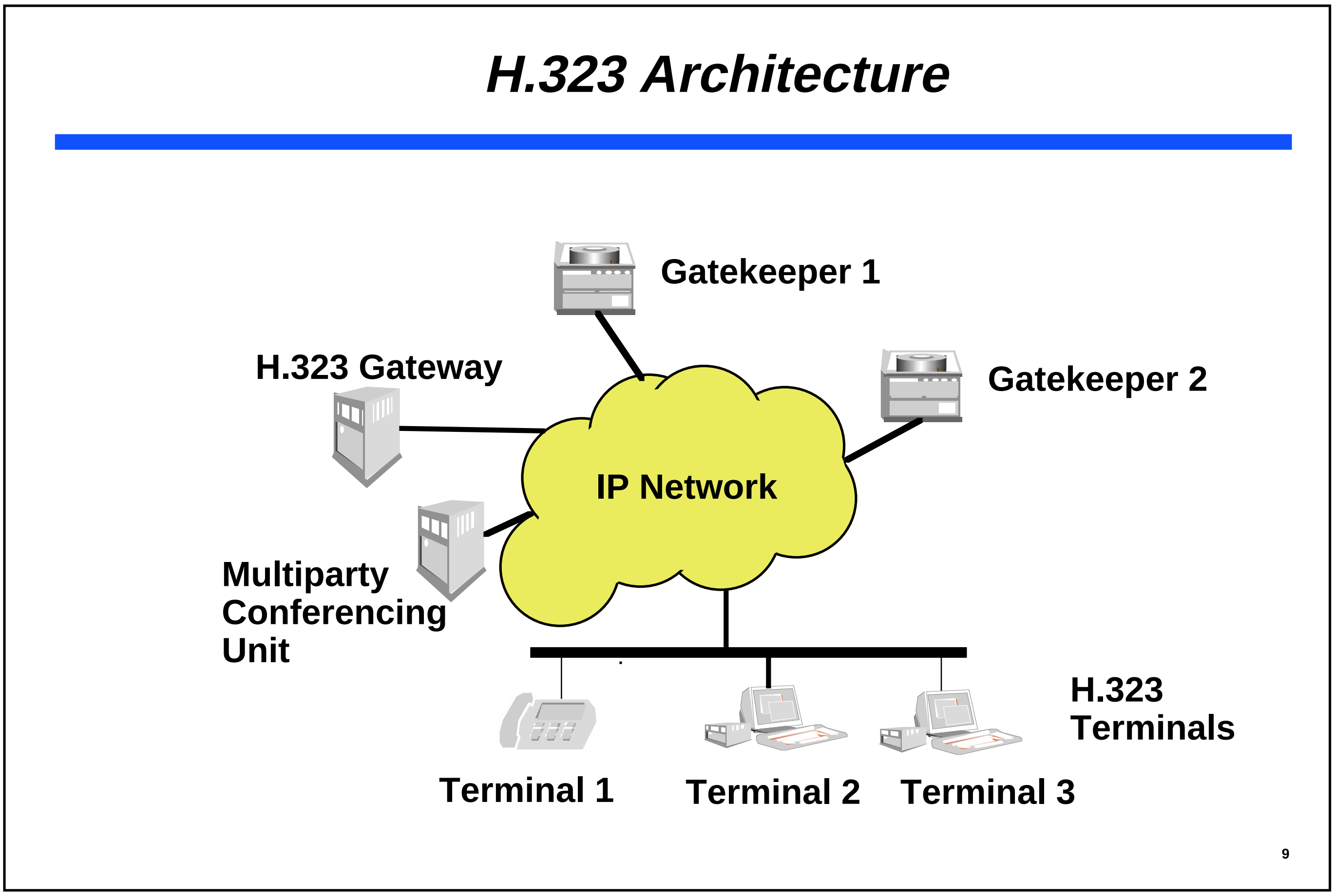
References

ITU-T Rec. H.323, Visual Telephony Systems and Terminal Equipment for Local Area Networks which provide a Non-Guaranteed Quality of Service, 1998

G.A. Thom, H.323: The Multimedia Communications Standard for local Area Net-works, IEEE Communications Mag. Vol.34, No5, December 1996, pp.52-56

D. Handley et al., SIP: Session Initiation Protocol, Internet Draft, Internet Engineering Task Force, Nov. 1998, Work in Progress

H.Schulzrinne et al., RTP: a transport protocol for real time application, RFC 1889, Internet Engineering Task Force, Jan. 1996



H.323 Architecture

Internet Telephony a la H.323

This slides shows a simplified H.323 network comprising three terminals, two gatekeepers, a gateway and a multipoint conference unit (MCU). It is used in the rest of the paper to illus-trate various H.323 features.

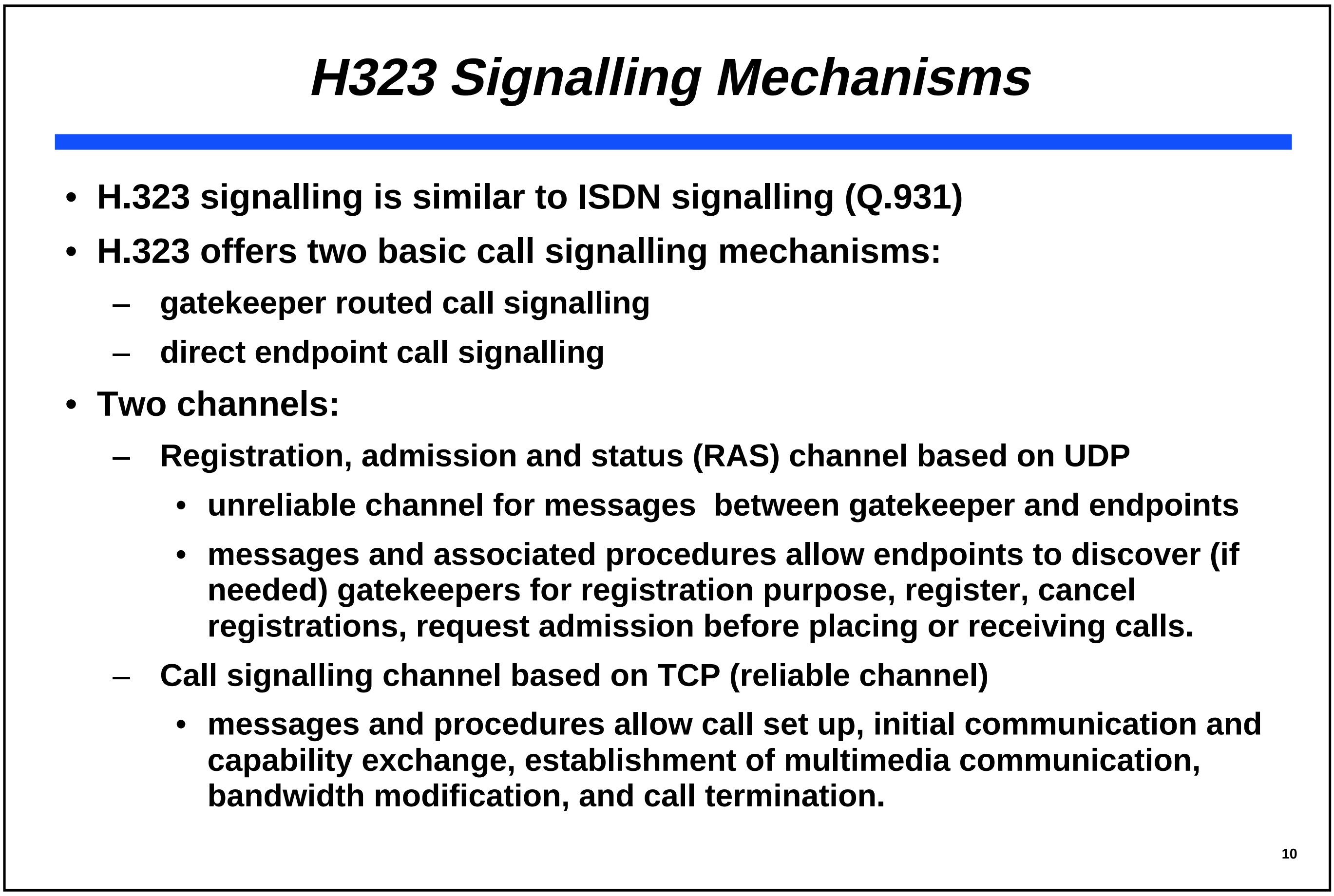
The H.323 entities are defined below.

• Terminal: Provides real time bidirectional multimedia communication including audio. H.323 specifies neither multimedia equipment nor multimedia application. It however specifies a variety of media coding options and mandates a set of capabilities in order to ensure a minimal level of interoperability.

• Gateway: Its key objective is to provide interoperability between H.323 terminals and ITU-T terminals on the circuit switched networks (CSNs) including both narrowband and broadband ISDNs. On the H.323 network side, a gateway is seen as an H.323 ter-minal. It can call and be called. H.323 entities which can call and be called are termi-nals and gateways. They are known as endpoints.

• Gatekeeper: Provides functions including address translation and admission control. Thanks to the address translation, external numbers such as telephone numbers and aliases can be used to address H.323 endpoints. The admission control function helps in administrating the network by controlling the amount of traffic. Endpoints interested in using these functions must explicit register to a gatekeeper. However gatekeepers are not mandatory. Furthermore, even if present, endpoints might select not to use their ser-vices.

• Multipoint Conferencing Unit (MCU): Is used for multipoint conference purpose



H323 Signalling Mechanisms

- **H.323 signalling is similar to ISDN signalling (Q.931)**
- **H.323 offers two basic call signalling mechanisms:**
  - **gatekeeper routed call signalling**
  - **direct endpoint call signalling**
- **Two channels:**
  - **Registration, admission and status (RAS) channel based on UDP**
    - **unreliable channel for messages between gatekeeper and endpoints**
    - **messages and associated procedures allow endpoints to discover (if needed) gatekeepers for registration purpose, register, cancel registrations, request admission before placing or receiving calls.**
  - **Call signalling channel based on TCP (reliable channel)**
    - **messages and procedures allow call set up, initial communication and capability exchange, establishment of multimedia communication, bandwidth modification, and call termination.**

10

H.323 signalling mechanisms

H.323 signalling messages are specified in H.225.0 while the procedures are specified in H.323 recommendation itself. The mechanisms draw very heavily on ISDN signalling as specified in recommendation Q.931. As endpoints may use gatekeeper functions as part of the process leading to call set ups, we introduce the mechanisms pertinent to this use before presenting the actual call signalling mechanisms.

Messages pertinent to the use of gatekeeper functions by endpoints are transported on an unreliable channel known as the registration, admission and status (RAS) channel. From the protocol stack point of view, user datagram protocol (UDP) is used. The messages and associated procedures allow endpoints to:

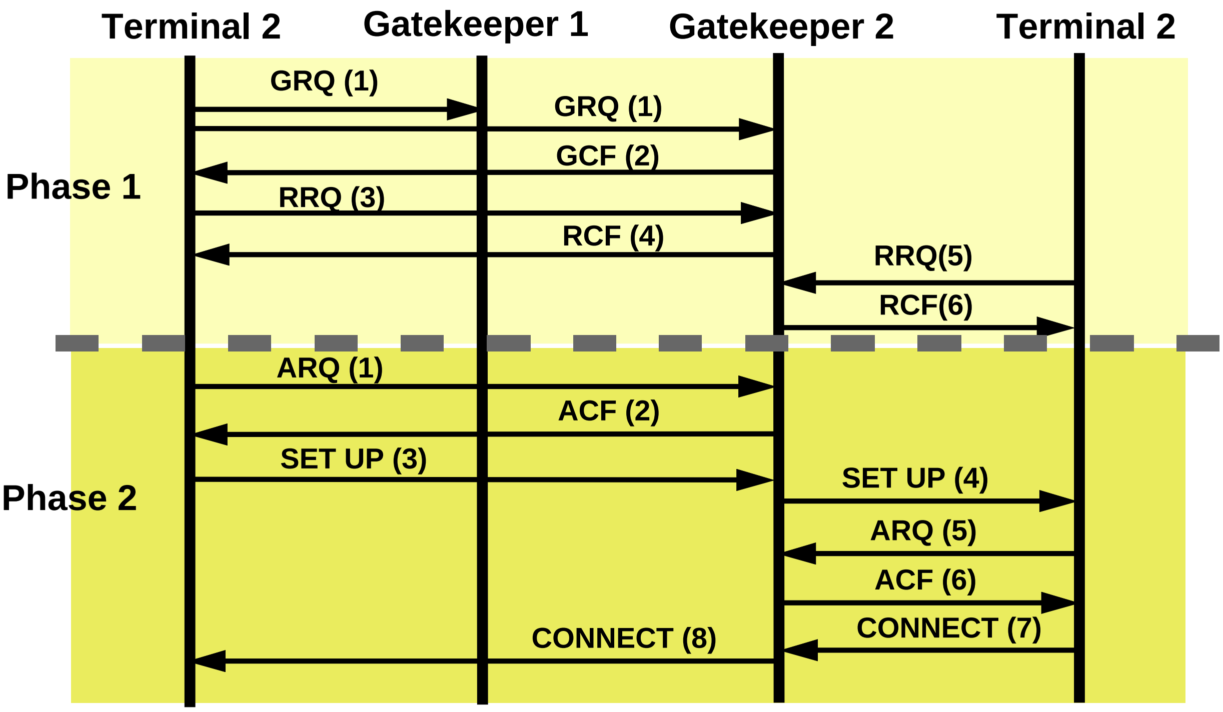
- discover (if needed) gatekeepers for registration purpose
- register by sending their addresses to gatekeepers
- cancel registrations
- Request admission before placing or receiving calls.

The amount of bandwidth needed for the call is specified in the request. Gatekeepers can initiate registration cancellations by sending appropriate messages to endpoints. They can also request endpoints status in order to figure out whether registered endpoints are alive or dead.

Call signalling messages and procedures allow call set up, initial communication and capability exchange, establishment of multimedia communication, call services such as increase or decrease of bandwidth, and call termination. The messages are exchanged using a reliable channel known as call signalling channel. From the protocol stack point of view, transport control protocol (TCP) is used.

H.323 offers two basic call signalling mechanisms: gatekeeper routed call signalling and direct endpoint call signalling. In the first case, all call signalling messages go via the gate-keeper while in the second case they are passed directly between endpoints. While endpoints which select to use gatekeepers functions might express preferences, gatekeepers have the last say when it comes to which method to use for a specific call. The question does not arise for unregistered terminals since gatekeepers do not even know they exist.

A Basic H.323 Call Set Up Scenario



Terminal 2

Gatekeeper 1

Gatekeeper 2

Terminal 2

Phase 1

GRQ (1)

GRQ (1)

GCF (2)

RRQ (3)

RCF (4)

RRQ(5)

RCF(6)

Phase 2

ARQ (1)

ACF (2)

SET UP (3)

SET UP (4)

ARQ (5)

ACF (6)

CONNECT (7)

CONNECT (8)

A basic H.323 call set up scenario

This slide depicts in two phases a basic call set up scenario with gatekeeper routed signal-ling. It is based on the simplified network from figure 2. This scenario like the other sce-narios which will be presented in the rest of the paper is simplified in the sense that only essential messages are shown. We assume that terminal 1 and terminal 2 are registered to the same gatekeeper, gatekeeper 2. We further assume that terminal 2 knows a priori that it is associated with gatekeeper 2. This is known as manual discovery.

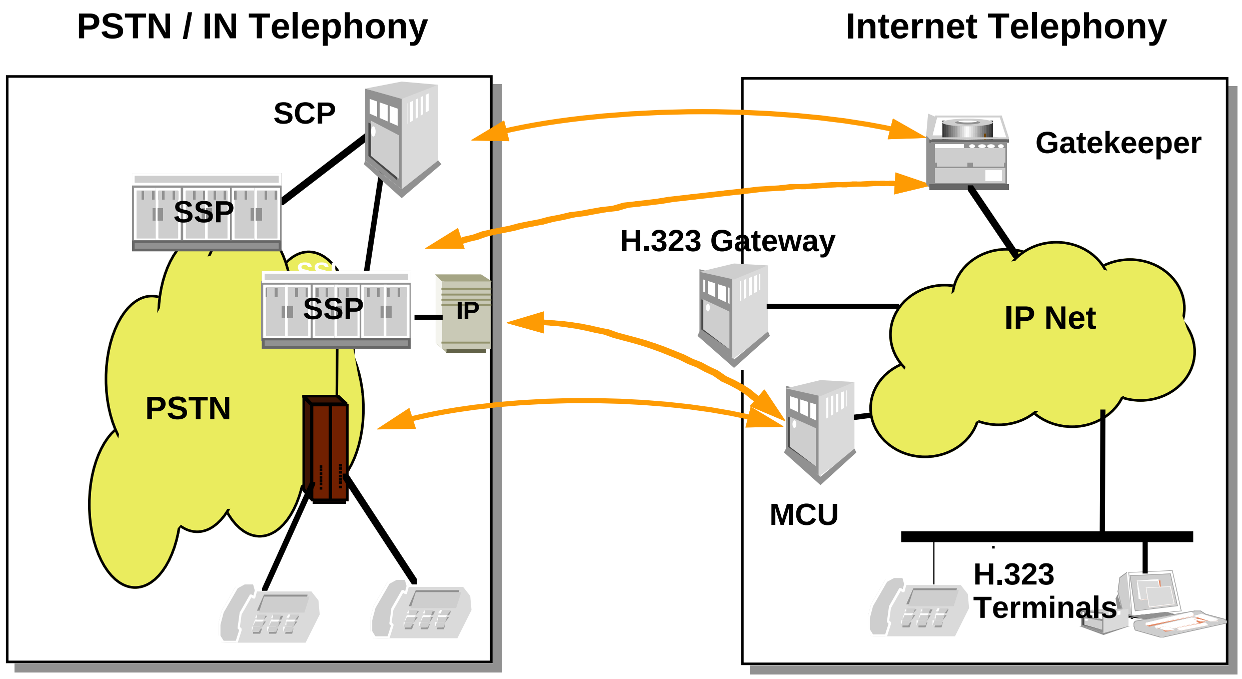
In the first phase, terminal 1 starts by multicasting a gatekeeper request (GRQ) message to the two gatekeepers in the network (i.e. gatekeeper 1 and gatekeeper 2). Gatekeeper 2 replies by a gatekeeper confirmation (GCF) message indicating that it can be terminal 1' gatekeeper. Terminal 1 then asks to register by sending a registration request (RRQ) message and gets a confirmation by receiving the registration confirm (RCF).

Let us assume that terminal 2 powers up just after. It registers directly to gatekeeper 2 without going through the gateway discovery process.

In the second phase, terminal 1 decides to call terminal 2. It starts by sending a admission request (ARQ) to gatekeeper 2. The gatekeeper confirms by sending an admission confirm (ACF) and indicates that the call shall be gatekeeper routed. Terminal 1 then sends a call set up message to the gatekeeper which forward it to terminal 2. Terminal 2 replies by a call proceeding then proceed to request admission to the gatekeeper. When admitted it sends a connect message to the gatekeeper to indicate that the call has been connected.

The gatekeeper forwards the message to terminal 1.

A Comparison



12

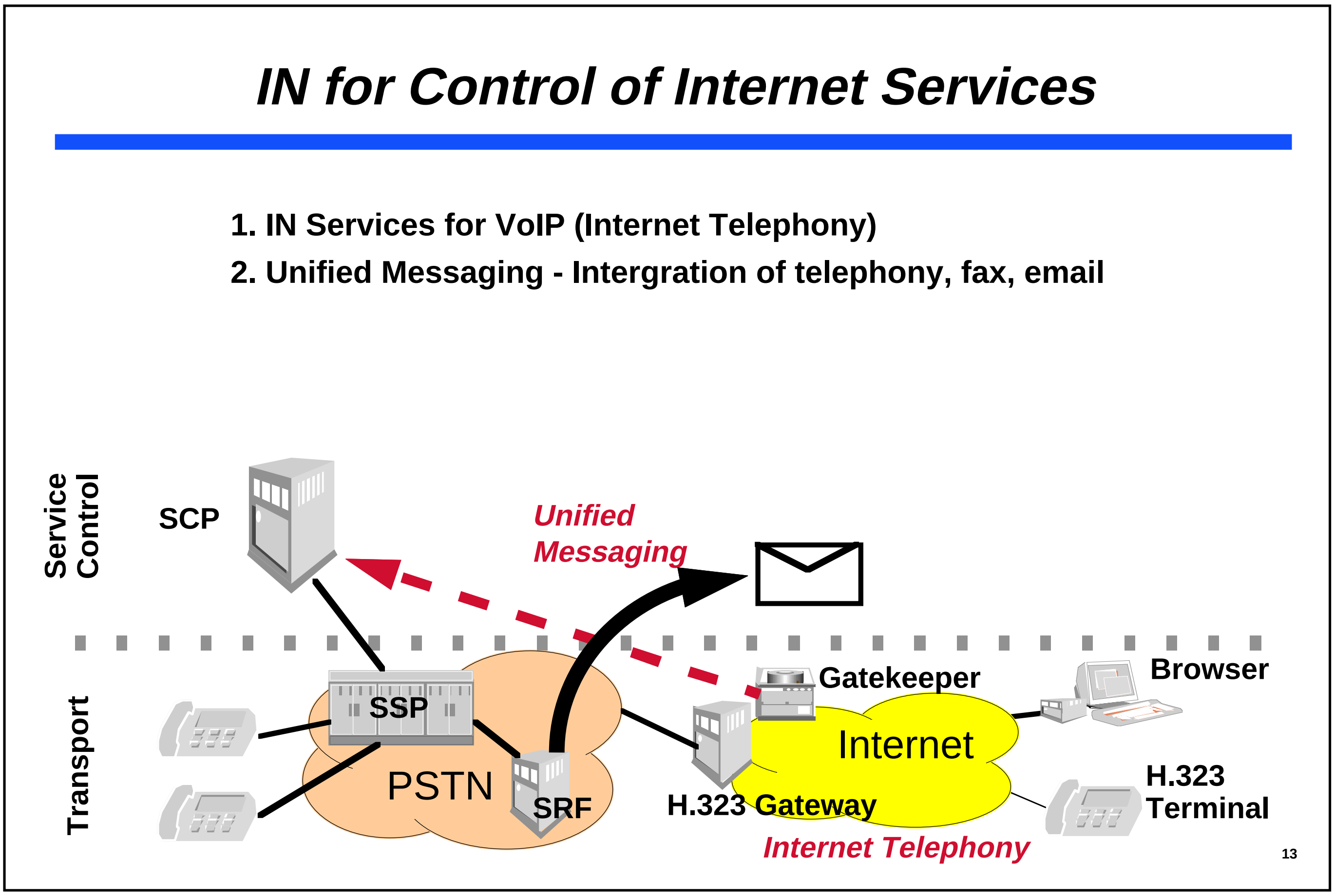
This slide depicts acomparison between the IN components and the Internet Telephony components defined in H.323.

Note that at this point in time it is not foreseen, that the Gatekeeper is in charge for providing value-added / supplementary services. However, it is in charge for routing and therefore provides theoretical this opportunity.

There are some views considering the Gatekeeper also as pseudo-SSP, which may interact for the provision of advanced routing services with an IN SCP.

The MCU represents an optional resource, and therefore compares to the IP in the IN domain.

Also the Gateway may be seen as an IN IP.



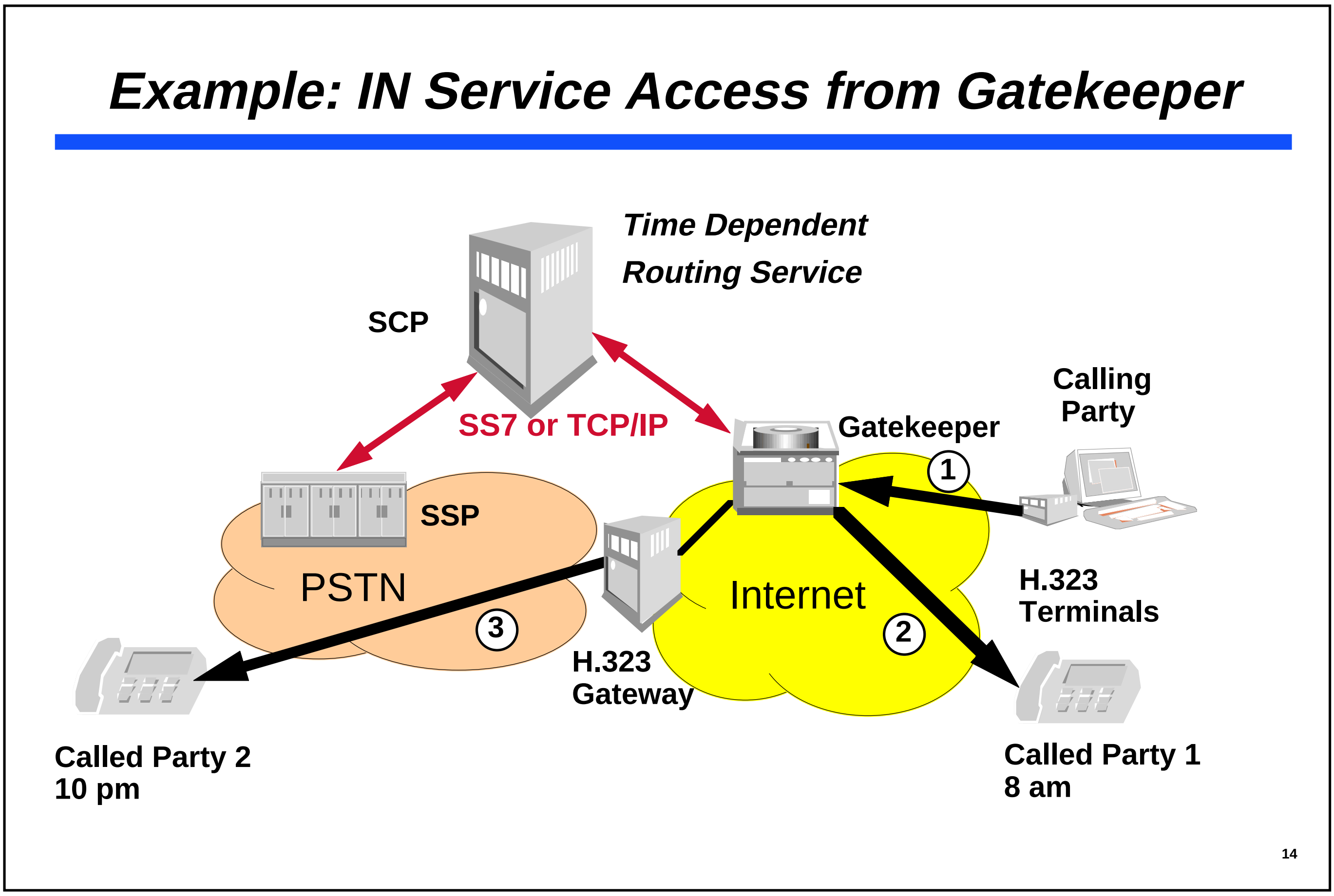
IN for Control of Internet Services

1. IN Services for VoIP (Internet Telephony)
2. Unified Messaging - Intergration of telephony, fax, email

This slide depicts the integration at transport level, which includes on the one hand the usage of IN services for providing advanced services on top of Voice over IP (VoIP) networks, as well as the provision of integrated services, such as Unified Messaging (e.g., voice mails received by an IN SRF) may be delivered as email to the called party.

Note that in this context the SRF will play a important role, since this very IN element provides the chanc to introduce inside the IN domain new capabilities. It will act as the gatewy to the internet, as it can be an internet web server.

Furthermore it can act as gateway to VoIP networks!



Example: IN Service Access from Gatekeeper

Time Dependent Routing Service

SCP

SS7 or TCP/IP

SSP

PSTN

Gatekeeper

Calling Party

Internet

H.323 Terminals

H.323 Gateway

1

2

3

Called Party 2
10 pm

Called Party 1
8 am

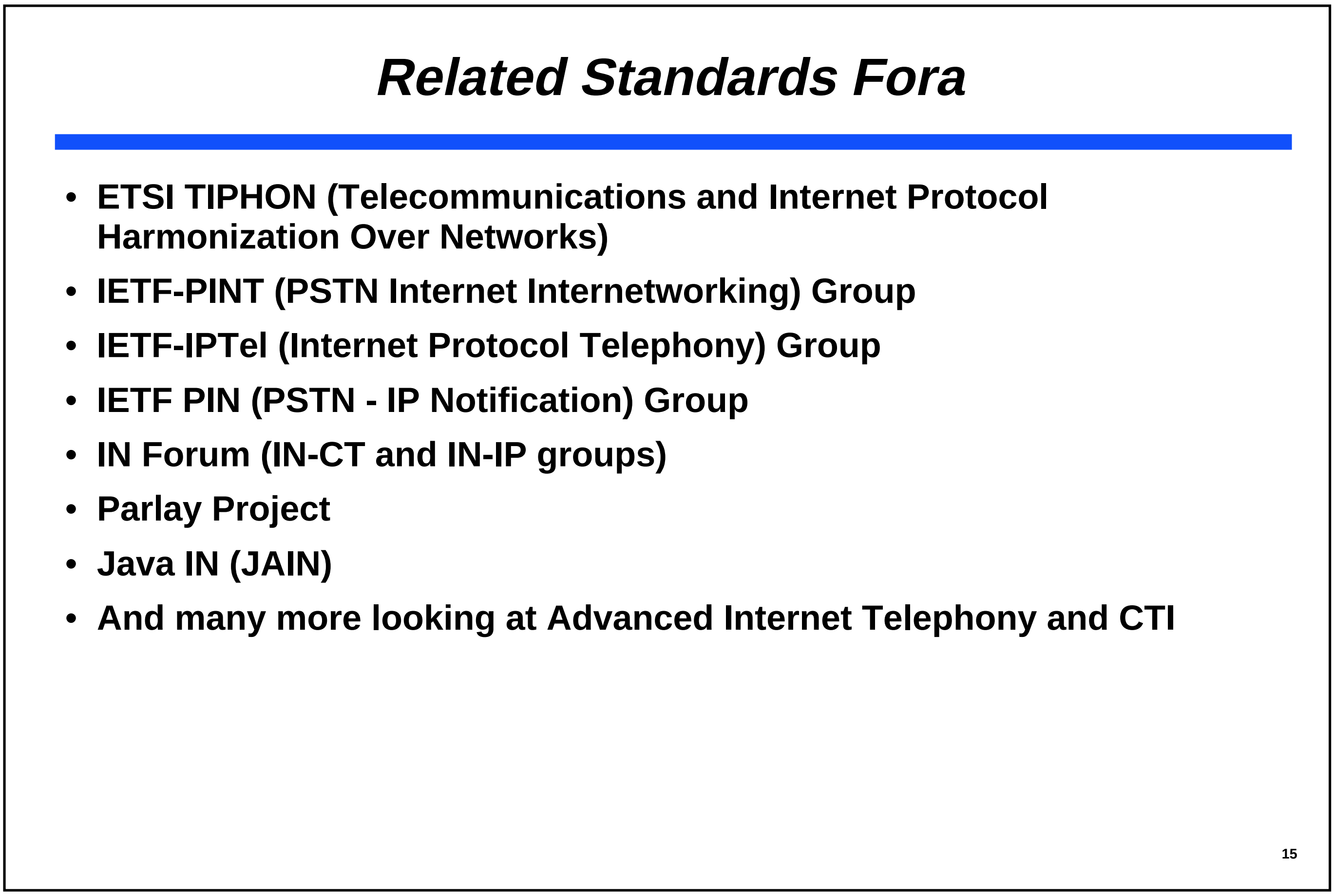
14

The scenario above depicts the provision of a time dependent routing service, which is provided across PSTN and VoIP networks. The user uses both a VoIP network during the day and the PSTN in the evening and night.

In the example above the calling party sits at the VoIP network. Making a call at 8 am the gatekeeper either may find a registration of the user locally or (more likely) asks an IN SCP how to treat the call. The SCP passes back as the result of this query the appropriate internet address of the called party to the gatekeeper, which subequently establishes the internet telephony call.

Assuming that the calling party makes again a call during the evening (10 pm), the SCP query of the gatekeeper will result in getting back a PSTN number and the address of a gateway able to deliver the call to the PSTN.

Note that the calling party may also be a PSTN user, resulting in a morning call to be routed from the IN via the SRF acting as an PSTN/VoIP gateway (not illustrated).



Related Standards Fora

- ETSI TIPHON (Telecommunications and Internet Protocol Harmonization Over Networks)
- IETF-PINT (PSTN Internet Internetworking) Group
- IETF-IPTel (Internet Protocol Telephony) Group
- IETF PIN (PSTN - IP Notification) Group
- IN Forum (IN-CT and IN-IP groups)
- Parlay Project
- Java IN (JAIN)
- And many more looking at Advanced Internet Telephony and CTI

15

Related Standards Fora

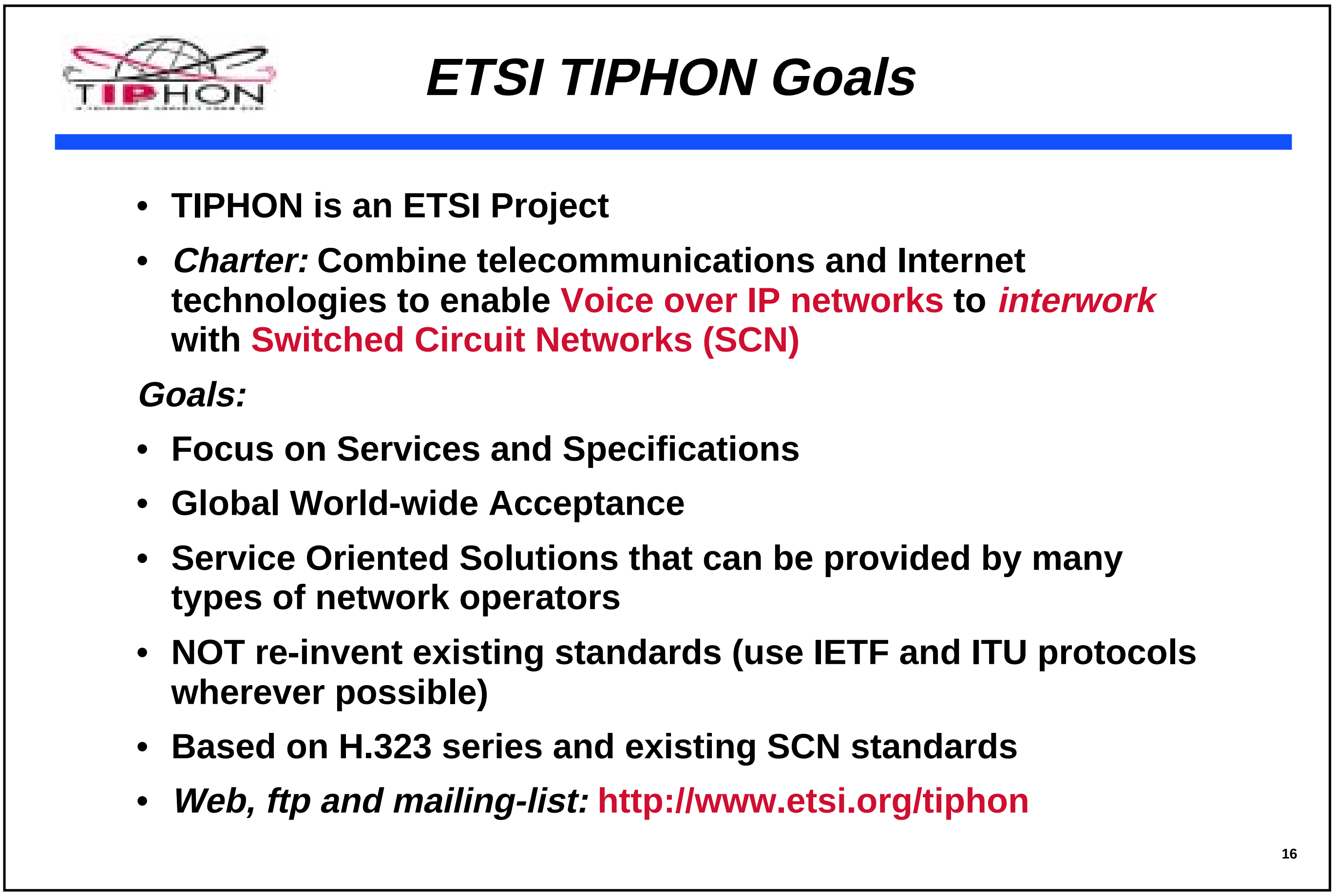
1) TIPHON: It’s an ETSI activity that focuses on how to "combine telecommunications and internet technologies to enable voiceband communications over IP networks to work with Circuit Switched Networks”. Look at “http://www.etsi.org/tiphon/”

2) Multiple Internet Engineering Task Force (IETF) Working Groups, such as PINT, IPTEL, Megaco, SigTrans, PIN, and many more. More details are given in the following!

3) IN Forum (INF) CT-IN Integration WG and IN/IP Integration WG. During 1998, the IN/CT workgroup has been developing a positioning paper to define the opportunities and the services enabled by the convergence of IN and Computer Telephony. The mission of the IN/SS7-Internet Protocol Working Group is to facilitate the broader application of Intelligent Network (IN) capabilities in the Public Switched Telephone Network (PSTN) to access Internet functionality and vice-versa for providing synergistic arrangements for existing and new services. The Working Group will define interoperability requirements, identify any protocol shortfalls, strive to resolve any such shortfalls through influencing appropriate standards, and develop implementation agreements that meet industry opportunities for inter-networking IN/SS7 and Internet capabilities. See http://WWW.INF.ORG/techoverview.htm and http://WWW.INF.ORG/library.htm for details.

4) BT, DGM&S Telecom, Microsoft, Nortel Networks and Siemens have formed a new working group, called the Parlay Group, to develop a specification for an open network interface that will widen the horizons of the intelligence in networks. This specification will enable service providers, ISVs and other developers in the IT and telecommunications industries to generate a new range of applications which access and use functionality resident in public and/or private communications networks while maintaining the integrity, performance, and security of those networks. See http://www.parlay.org.

5) Java IN (JAIN) is an activity initiated by Sun in order to allow the use of Sun’s Java technology for providing IN services. See “http://www.sun.com/smi/Press/sunflash/9806/sunflash.980609.9.html” for more details.



ETSI TIPHON Goals

- TIPHON is an ETSI Project
- *Charter:* Combine telecommunications and Internet technologies to enable Voice over IP networks to *interwork* with Switched Circuit Networks (SCN)

Goals:

- Focus on Services and Specifications
- Global World-wide Acceptance
- Service Oriented Solutions that can be provided by many types of network operators
- NOT re-invent existing standards (use IETF and ITU protocols wherever possible)
- Based on H.323 series and existing SCN standards
- *Web, ftp and mailing-list:* http://www.etsi.org/tiphon

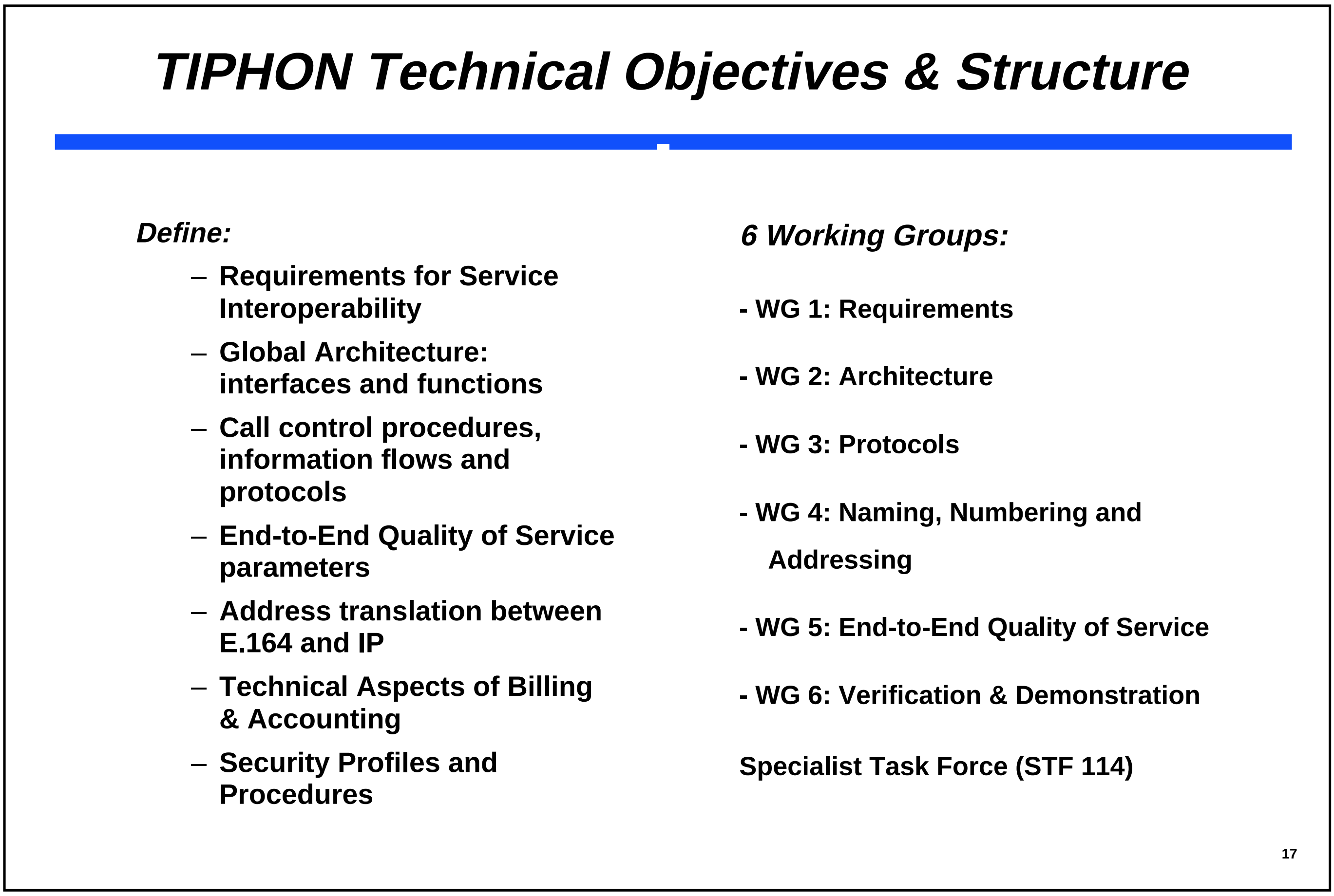
16

Recognizing the urgent need for common solutions, the European Telecommunications Standards Institute, ETSI, has established Project TIPHON - "Telecommunications and Internet Protocol Harmonization Over Networks".

The project's objective is to support the market for voice communication and related voiceband communication (such as facsimile) between users. It will ensure that users connected to IP based networks can communicate with users in Switched Circuit Networks (SCN - such as PSTN[1] /ISDN[2] and GSM[3]), and vice versa. As well as between users in SCN, where IP-based networks are used for connection/trunking between the SCN involved. The support comes in the production of appropriate ETSI deliverables: technical specifications and reports. In addition, the activity will include validation and demonstrations, in order to confirm the appropriateness of the solutions proposed.

Given the universal nature of IP networks, the prime goal is to produce global standards. As ETSI is essentially a European body, it recognizes that co-operation with relevant groupings in ITU-T[4] and IETF[5] is essential. Specifically, ETSI believes that it has a role in opinion leadership and in helping to build consensus between all the major market players. The Institute co-operates closely with relevant Fora, especially the IMTC[6] VoIP[7] Activity Group.

The initiative for Project TIPHON was a joint one between a number of ETSI members and the ETSI Secretariat. Since its creation, support for the Project has continued to grow and, at the time of writing, more than 40 ETSI members and other companies have committed their support. All of those members are major players in telecommunications and information technology, and most of them are companies with substantial global business. Other companies are most welcome to join the Project under the normal terms of ETSI membership.



TIPHON Technical Objectives & Structure

Define:

- Requirements for Service Interoperability
- Global Architecture: interfaces and functions
- Call control procedures, information flows and protocols
- End-to-End Quality of Service parameters
- Address translation between E.164 and IP
- Technical Aspects of Billing & Accounting
- Security Profiles and Procedures

6 Working Groups:

- WG 1: Requirements
- WG 2: Architecture
- WG 3: Protocols
- WG 4: Naming, Numbering and Addressing
- WG 5: End-to-End Quality of Service
- WG 6: Verification & Demonstration

Specialist Task Force (STF 114)

Much of the specification work will be carried out by ETSI'snormal Technical Organization: its Technical Committees and their Working Groups. But where the timescales forbid this, or where work is only relevant to the Project, Working Groups within Project TIPHON will take responsibility. Close liaison is being maintained between the Project and the relevant elements of the Technical Organization as part of the project management process. At present the following Working Group themes have been identified:

Requirements for service interoperability, technical aspects of charging/billing and security;

Architecture and reference configurations;

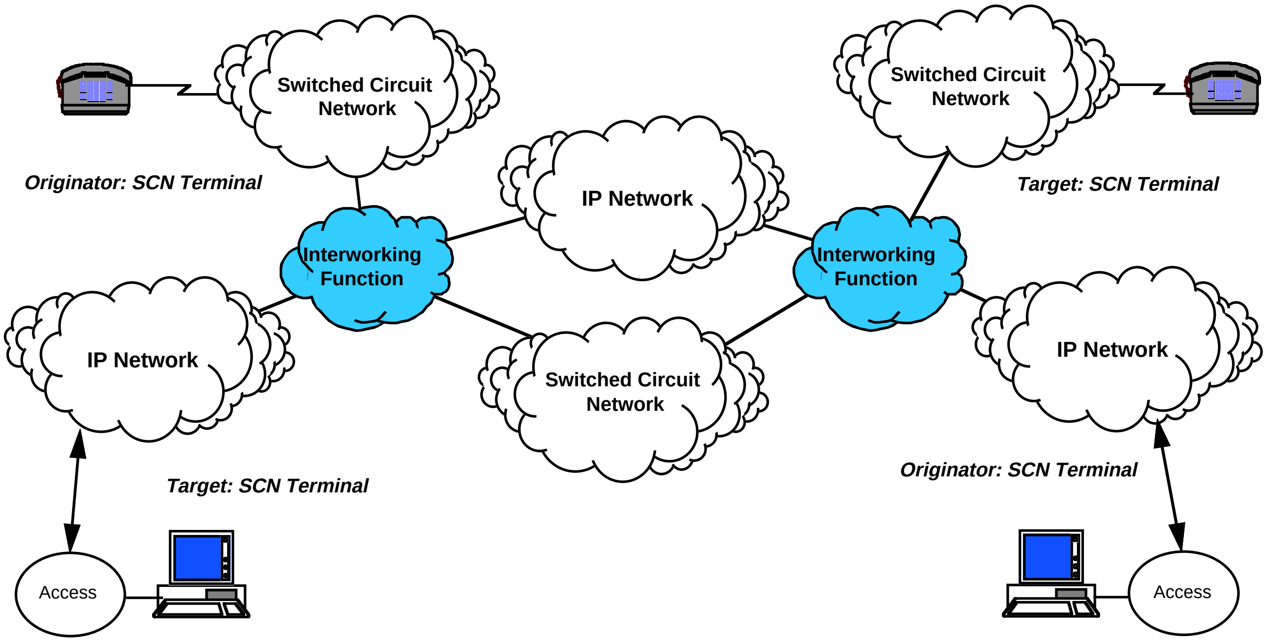
Call control procedures, information flows and protocols;

Naming, Numbering and Addressing;

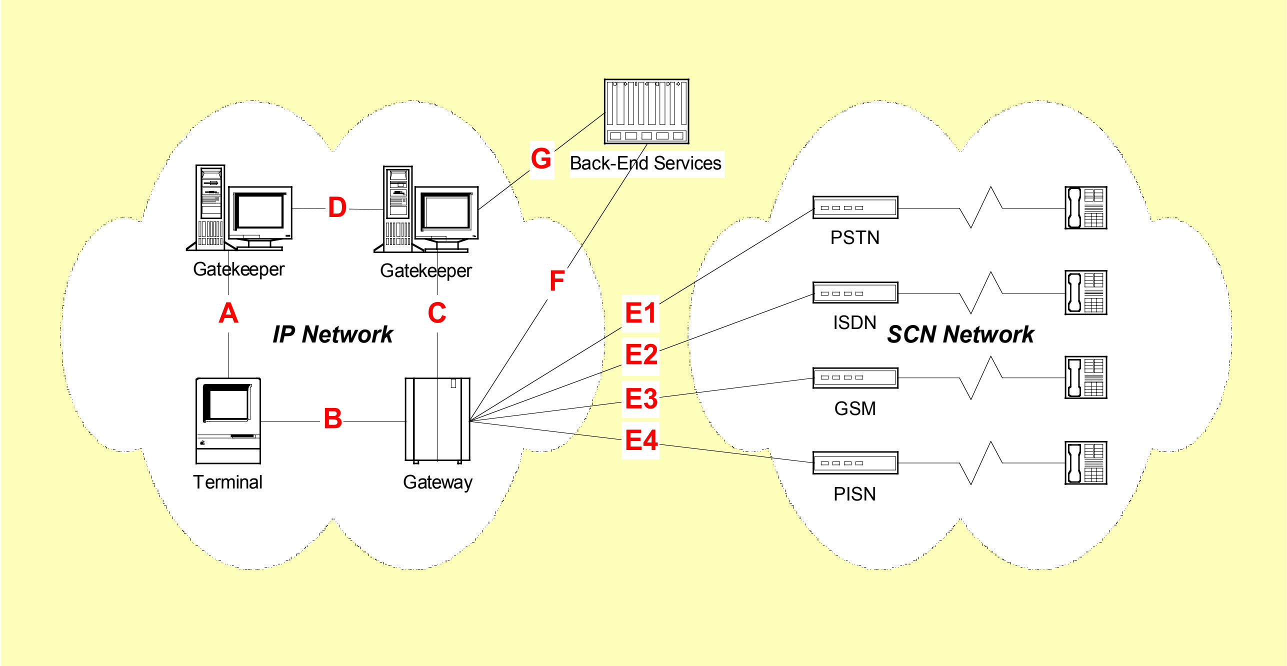
Quality of Service;

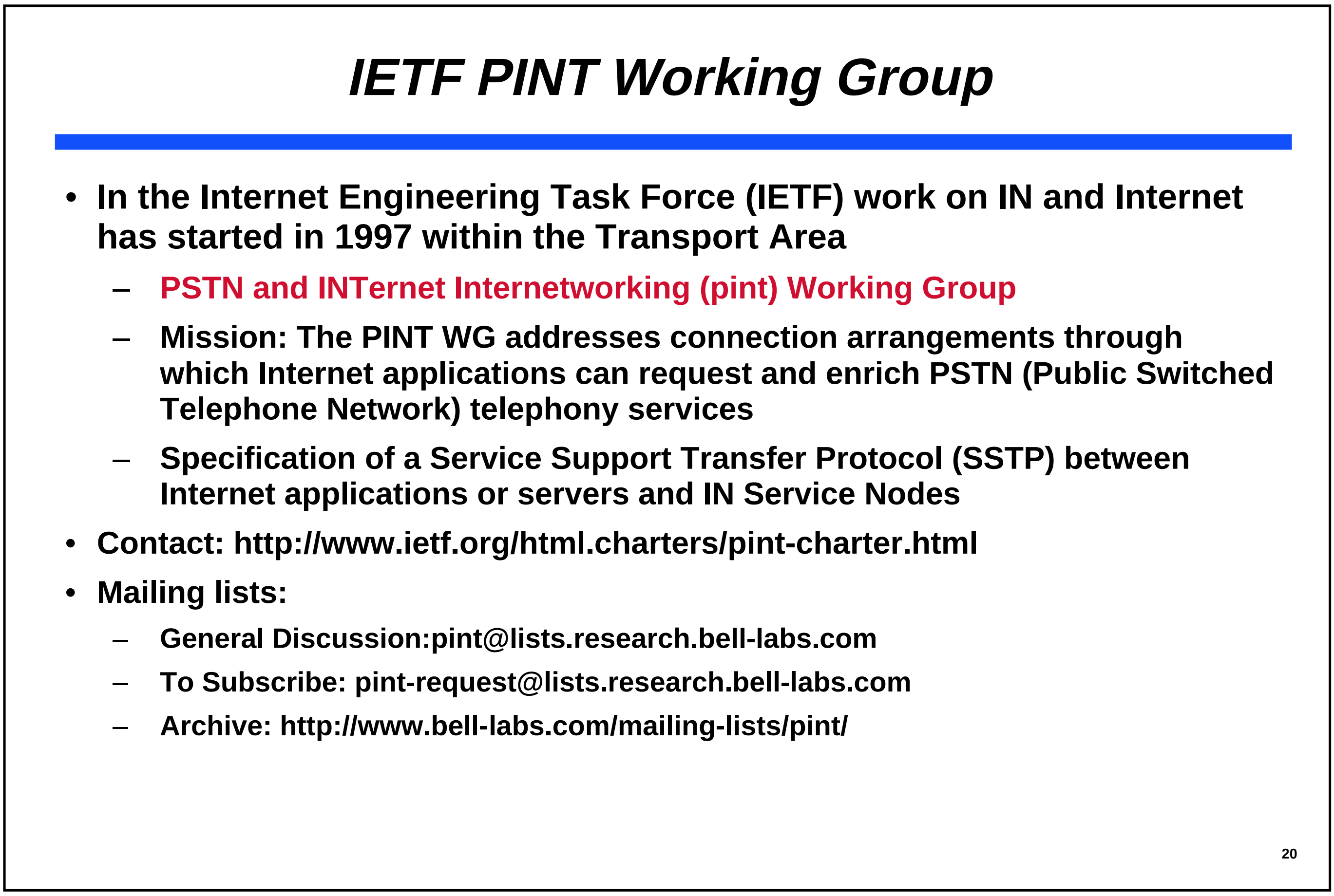
Verification and Demonstration Implementation.

TIPHON Interworking Scenarios



TIPHON Basic Call Reference Configuration





IETF PINT Working Group

- In the Internet Engineering Task Force (IETF) work on IN and Internet has started in 1997 within the Transport Area
  - PSTN and INTernet Internetworking (pint) Working Group
  - Mission: The PINT WG addresses connection arrangements through which Internet applications can request and enrich PSTN (Public Switched Telephone Network) telephony services
  - Specification of a Service Support Transfer Protocol (SSTP) between Internet applications or servers and IN Service Nodes
- Contact: http://www.ietf.org/html.charters/pint-charter.html
- Mailing lists:
  - General Discussion:pint@lists.research.bell-labs.com
  - To Subscribe: pint-request@lists.research.bell-labs.com
  - Archive: http://www.bell-labs.com/mailing-lists/pint/

The PSTN/Internet Interfaces (PINT) WG addresses connection arrangements through which Internet applications can request and enrich PSTN (Public Switched Telephone Network) telephony services. An example of such services is a Web-based Yellow Pages service with the ability to initiate PSTN calls between customers and suppliers.

This working group has six main objectives:

* Study architecture and protocols needed to support services in which a user of the Internet requests initiation of a telephone (i.e., PSTN-carried) call to a PSTN terminal (i.e., telephone, FAX machine). The protocols are not to support any form of third-party call control or, for that matter, any type of call control; their role is rather to securely carry call requests to the PSTN. Specific services to be considered initially are Click-to-Dial, Click-to-Fax, Click-to-Fax-Back, and Web access to voice content delivered over the PSTN.

* Produce an informational RFC that describes current practices for supporting the services in question.

* Based on the existing practice and agreed on improvements, develop a standards track RFC that specifies a Service Support Transfer Protocol (SSTP) between Internet applications or servers and PSTN Intelligent Network Service Nodes (or any other node that implement the Service Control Function). SSTP is an application-specific transport protocol operating over TCP.

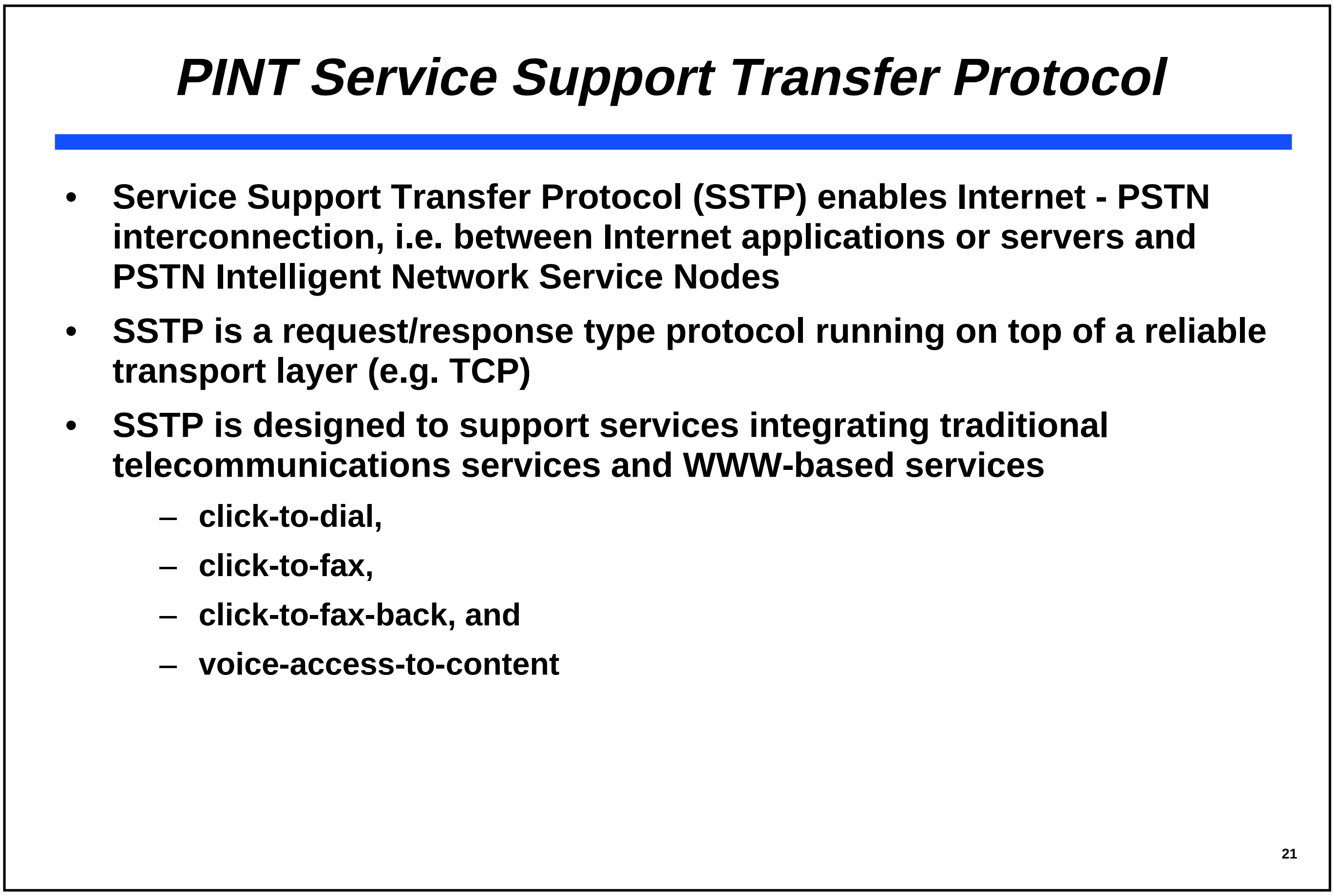
* Consider security issues relating to prividing functions of this type. In particular understand any threats posed by this technology and resolve them, and any other security issues in the proposed standard.

* Based on the existing practice and agreed on improvements, develop a standards track RFC for a relevant MIB (SSTP MIB) to support the service management protocol between Internet applications and the PSTN Service Management System. The SSTP MIB is to conform to SNMP standards.

* Consider extensions of the above architecture and protocols to support a wider range of PSTN Intelligent Network (IN) based services.

IETF PINT URL: “http://www.ietf.org/html.charters/pint-charter.html”

PINT documents are available through “http://www.bell-labs.com/mailing-lists/pint/”



PINT Service Support Transfer Protocol

- **Service Support Transfer Protocol (SSTP) enables Internet - PSTN interconnection, i.e. between Internet applications or servers and PSTN Intelligent Network Service Nodes**
- **SSTP is a request/response type protocol running on top of a reliable transport layer (e.g. TCP)**
- **SSTP is designed to support services integrating traditional telecommunications services and WWW-based services**
  - **click-to-dial,**
  - **click-to-fax,**
  - **click-to-fax-back, and**
  - **voice-access-to-content**

Service Support Transfer Protocol (SSTP) is running on top of a reliable transport layer (such as TCP). The SSTP is for interconnection between the Internet and the Public Switched Telephone Network (PSTN), specifically, involving Web Server in the former; and Service Node (SN) or Service Control Point (SCP), and Service Management System (SMS) in the latter. It is to support a combination of services provided otherwise disjointly by either of the network types. Service examples are those integrating the traditional telephony services on the PSTN and the World Wide Web-based services on the Internet.

Envisaged services to be supported by SSTP:

• Click-to-Dial: A web user can request a PSTN call by clicking a button during a Web session. A typical scenario is that a user, while browsing through a catalogue, clicks the button inviting a sale representative to call him or her.

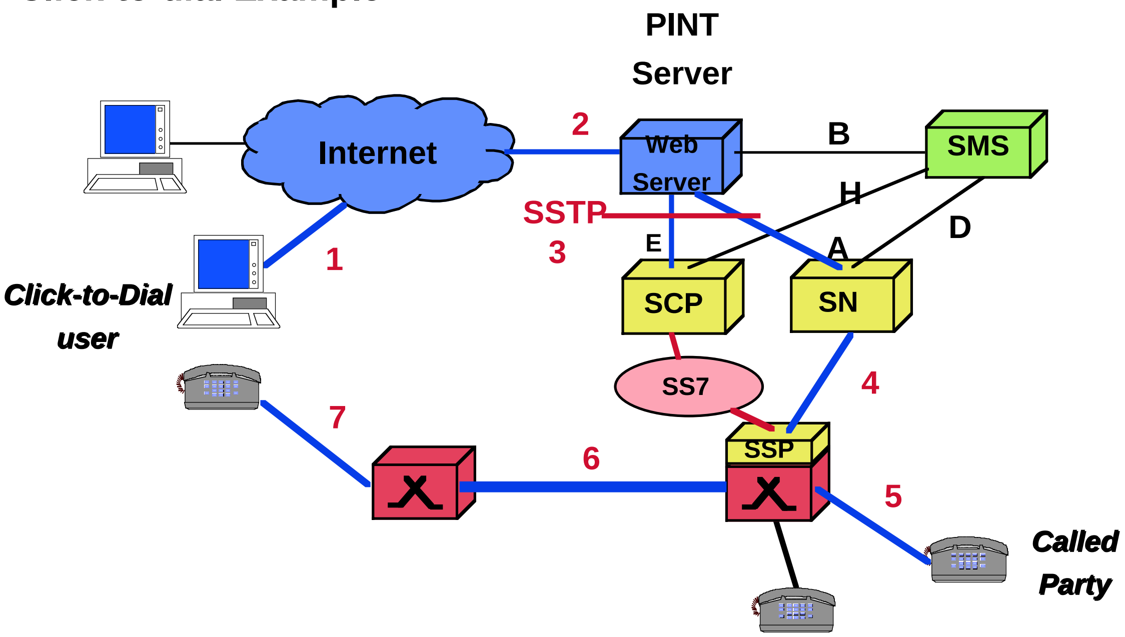
• Click-to-Fax: A web user can send a fax by clicking a button during aweb session.

• Click-to-Fax-Back: A web user can request (and subsequently receive) a fax by clicking a button during a web session.

• Voice-Access-to-Content: A web user can have access to the web content by telephone. The content is converted to speech and transmitted to the user on a telephone line.

PINT Architecture

Click-to-dial Example



A Click-to-Dial Service Scenario

A Web user, who has simultaneous access to the Web and telephone services (this can be achieved, for example, by having an ISDN connection), is browsing through a sales catalogue and deciding to speak to a sales representative.

When the Web user clicks a button inviting a telephone call from the sales office, the Web server sends a message to the SN over the A interface, thus crossing the Internet-to-PSTN boundary. By matching the information received from the Web server with the content provider profile that had been previously loaded and activated by the SMS over the D interface, the SN recognizes the signal.

At this point, the SN calls the Web user. The user answers the call, hears an announcement, e.g., "Please wait, while we are connecting youto the sale agent", and is waiting to be connected to the sale agent.Then the SN invokes service logic as indicated in the profile. The execution of this logic selects an appropriate sales agentto call based on the time of the day. It is 8 P.M. in New York where the Web user is located, and the New York sales office has closed. But the San Francisco office is still open, and so the SN selects an appropriate central office, establishes the connection (the interface C) to this central office, verifies that there is at least one sales agent line that is free and instructs the switch to call the agent. Finally, the SN bridges the two calls andestablishes a two-party call between the sales agent and the Web user.

IETF IPTEL Working Group

- **In the Internet Engineering Task Force (IETF) work on Internet Telephony is studied in the IP Telephony (iptel) Working Group**
- **Mission:**
  - **Develop supportive protocols and frameword documents for Internet Telephony. These include signaling and capabilities exchange, but also include a number of "peripheral" protocols for providing related services. Current work focuses on**
    - **Call Processing Syntax**
    - **Gateway Location Protocol (GLP)**
- **Contact: http://www.ietf.org/html.charters/iptel-charter.html**
- **Mailing lists:**
  - **General Discussion:iptel@lists.research.bell-labs.com**
  - **Archive: http://www.bell-labs.com/mailing-lists/iptel/**

Before Internet telephony can become a widely deployed service, a number of protocols must be deployed. These include signaling and capabilities exchange, but also include a number of "peripheral" protocols for providing related services.

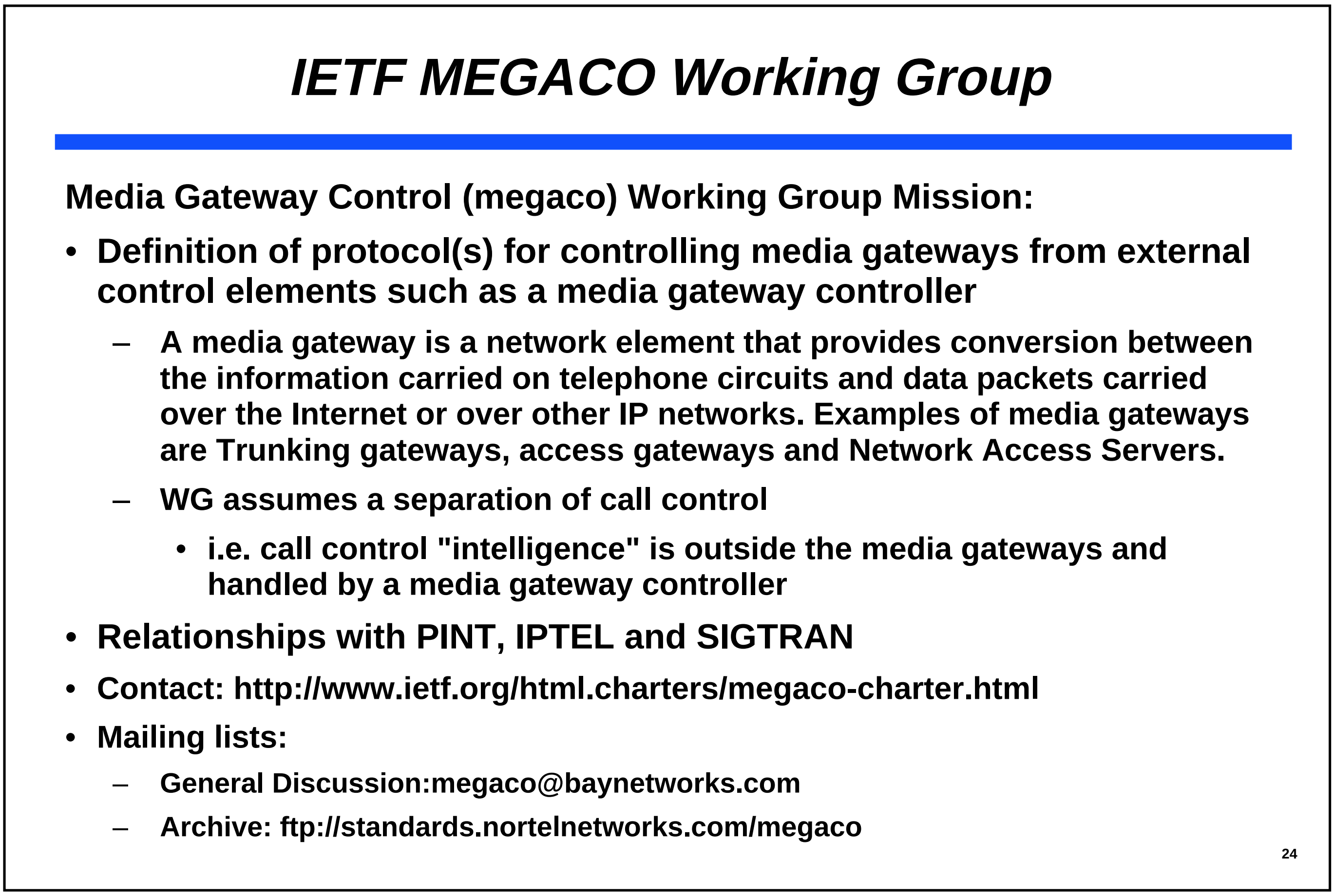
The primary purpose of the IETF IP Telephony (IPTEL) WG is to develop two such supportive protocols and a frameword document. They are:

1. Call Processing Syntax/Logic (CPL): When a call is setup between two endpoints, the signaling will generally pass through several servers (such as an H.323 gatekeeper) which are responsible for forwarding, redirecting, or proxying the signaling messages. For example, a user may make a call to j.doe@bigcompany.com. The signaling message to initiate the call will arrive at some server at bigcompany. This server can inform the caller that the callee is busy, forward the call initiation request to another server closer to the user, or drop the call completely (among other possibilities). It is very desirable to allow the callee to provide input to this process, guiding the server in its decision on how to act. This can enable a wide variety of advanced personal mobility and call agent services. Such preferences can be expressed in a call processing syntax, which can be authored by the user (or generated automatically by some tool), and then uploaded to the server. The group will develop this syntax, and specify means of securely transporting and extending it. The result will be a single standards track RFC. In addition, the group will write a service model document, which describes the services that are enabled by the call processing syntax, and discusses how the syntax can be used. This document will result in a single RFC.

3. As IP telephony gateways grow in terms of numbers and usage, managing their operation will become increasingly complex. One of the difficult tasks is that of gateway location, also known as gateway selection, path selection, gateway discovery, and gateway routing. The essence of the problem is that an ingress gateway or end user must select a gateway to terminate a call towards the PSTN. This gateway may lie in a remote administrative domain, and the selection of the gateway may be based on a number of criteria. To support discovery and location of gateways in remote administrative domains, a protocol is being developed, call the Gateway Location Protocol (GLP).

IETF PINT URL: “http://www.ietf.org/html.charters/iptel-charter.html”

PINT documents are available through “http://www.bell-labs.com/mailing-lists/iptel/”



IETF MEGACO Working Group

Media Gateway Control (megaco) Working Group Mission:

- **Definition of protocol(s) for controlling media gateways from external control elements such as a media gateway controller**
  - **A media gateway is a network element that provides conversion between the information carried on telephone circuits and data packets carried over the Internet or over other IP networks. Examples of media gateways are Trunking gateways, access gateways and Network Access Servers.**
  - **WG assumes a separation of call control**
    - **i.e. call control "intelligence" is outside the media gateways and handled by a media gateway controller**
- **Relationships with PINT, IPTEL and SIGTRAN**
- **Contact: http://www.ietf.org/html.charters/megaco-charter.html**
- **Mailing lists:**
  - **General Discussion:megaco@baynetworks.com**
  - **Archive: ftp://standards.nortelnetworks.com/megaco**

24

IETF Media Gateway Control (megaco)Working Group

The working group will develop an informational RFC detailing the architecture and requirements for controlling Media Gateways from external control elements such as a Media Gateway Controller. A media gateway is a network element that provides conversion between the information carried on telephone circuits and data packets carried over the Internet or over other IP networks. This work will be done in consultation with other IETF working groups looking at similar issues. The working group will also ensure that good information conduits exist with groups in other standards groups with expertise in the relevant PSTN technology. Other IETF working groups include PINT, IPTEL and SIGTRAN. In addition the working group will ensure that reasonable liaisons exist with similar activities in other standards bodies such as the ITU-T and ETSI.

This working group will also define standards track protocol(s) for controlling media gateways from external control elements such as a media gateway controller.

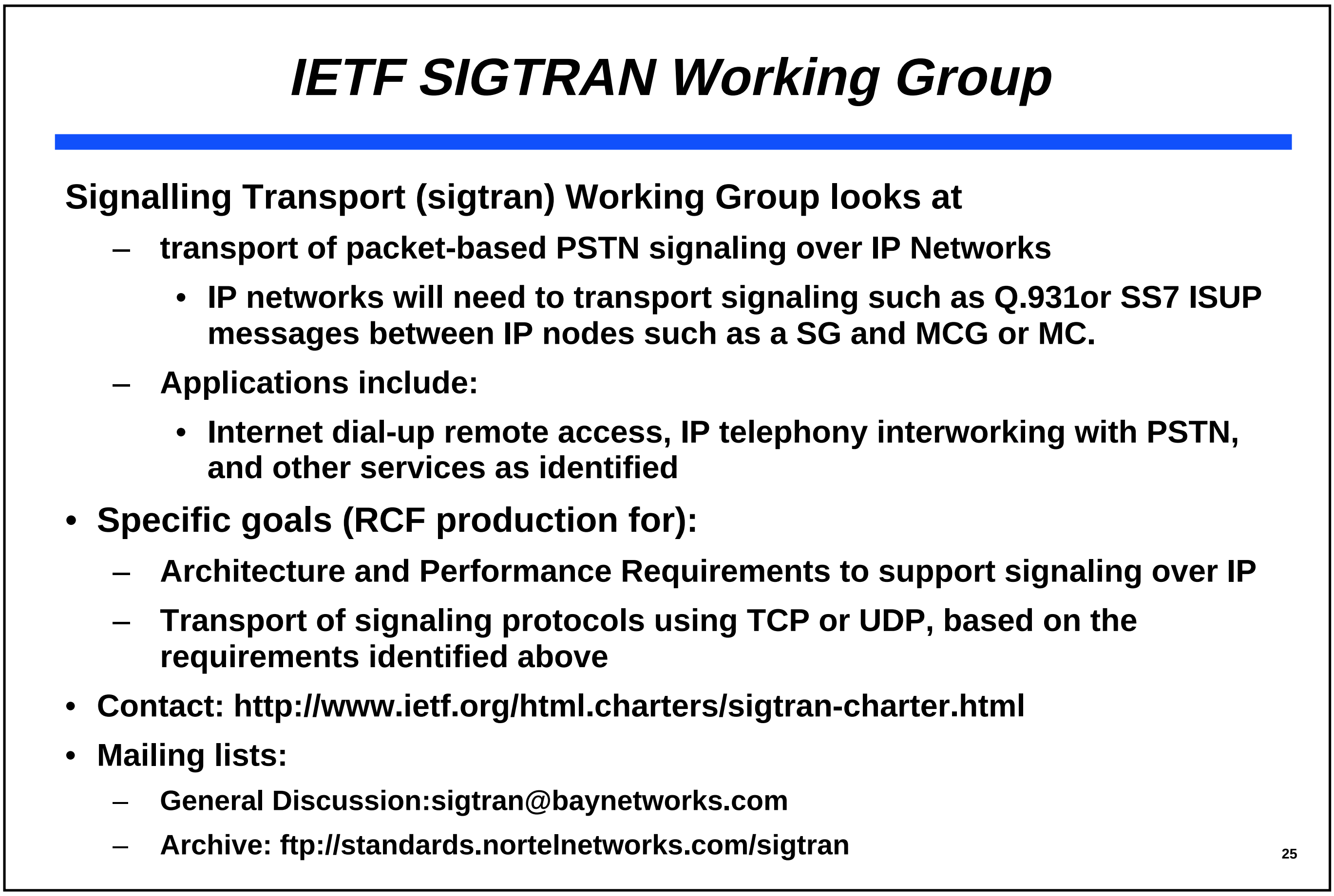
Examples of media gateways are:

* Trunking gateways that interface between the telephone network and a Voice over IP network. Such gateways typically manage a large number of digital virtual circuits.

* Access gateways that provide traditional analog or Primary Rate (PRI) line interfaces to a Voice over IP network.

* Network Access Servers that can attach a "modem" to a telephone circuit and provide data access to the Internet.

This working group assumes a separation of call control so the call control "intelligence" is outside the media gateways and handled by a media gateway controller. This group will NOT work on media gateway controller to media gateway controller protocols, nor on media gateway to media gateway protocols.



IETF SIGTRAN Working Group

Signalling Transport (sigtran) Working Group looks at

- **transport of packet-based PSTN signaling over IP Networks**
  - **IP networks will need to transport signaling such as Q.931or SS7 ISUP messages between IP nodes such as a SG and MCG or MC.**
- **Applications include:**
  - **Internet dial-up remote access, IP telephony interworking with PSTN, and other services as identified**

- **Specific goals (RCF production for):**
  - **Architecture and Performance Requirements to support signaling over IP**
  - **Transport of signaling protocols using TCP or UDP, based on the requirements identified above**
- **Contact: http://www.ietf.org/html.charters/sigtran-charter.html**
- **Mailing lists:**
  - **General Discussion:sigtran@baynetworks.com**
  - **Archive: ftp://standards.nortelnetworks.com/sigtran**

IETF SIGTRAN Working Group

The primary purpose of this working group will be to address the transport of packet-based PSTN signaling over IP Networks, taking into account functional and performance requirements of the PSTN signaling. For interworking with PSTN, IP networks will need to transport signaling such as Q.931 or SS7 ISUP messages between IP nodes such as a Signaling Gateway and Media Gateway Controller or Media Gateway.

Examples of such transport include:

- transport of signaling between a Signaling Gateway and Media Gateway or Media Gateway Controller

- transport of signaling ("backhaul") from a Media Gateway to a Media Gateway Controller

- transport of TCAP between a Signaling Gateway and other IP nodes

Applications include:

- Internet dial-up remote access, IP telephony interworking with PSTN, and other services as identified

Specific goals are:

1. Architecture and Performance Requirements: The working group will produce an informational RFC identifying functionality and performance requirements to support signaling over IP. Signaling messages have very stringent loss and delay requirements in the existing telephone networks that need to be supported.

2. Transport: The working group will produce a standards track proposal or proposals defining transport of signaling protocols using TCP or UDP, based on the requirements identified above. These proposals will identify the method of encapsulation of different signaling protocols. This will include differentiating between different protocols being carried, and what components are transported, translated or terminated at the SG. Security and resilience must be addressed.

This work will be done in conjunction with other IETF working groups looking at similar issues.

The group will make use of existing IETF QoS and security technology and will not address creation of new QoS or security functions for IP networks. Nor will the working group work on defining new call control or device control protocols.

IETF PIN (PSTN - IP Notification) Group

- **Motivation: increasing desire for events occurring on the PSTN domain to be propagated to the Internet domain**
- **PIN attempts to define an appropriate protocol**
  - **between PIN Gateway, the PIN Server and various PIN clients.**
  - **Entities on the Internet domain can receive the events generated by the PSTN domain and act appropriately.**
  - **Internet Draft "Need for PSTN Internet Notification (PIN) Services" is available from the official IETF directory.**
- **Relations to other IETF working groups**
  - **most notably IPTEL, PINT, and MEGACO**
- **PIN mailing list is located at ietf-pin@lists.research.bell-labs.com.**
- **Chair: Alec Brusilovsky <abrusilovsky@lucent.com>**

IETF PSTN - IP Notification (PIN) Working Group

With the proliferation and wide acceptance of the Internet, and more so with the convergence of the Internet and PSTN, there is an increasing desire for events occurring on the PSTN domain to be propagated to the Internet domain.

The protocol of the PINT working group is used to pass requests for service from an IP host to a telephone network, and to receive responses back from the telephone network. But some Interworking services require that requests for services go in the opposite direction: from a telephone network to an IP host.

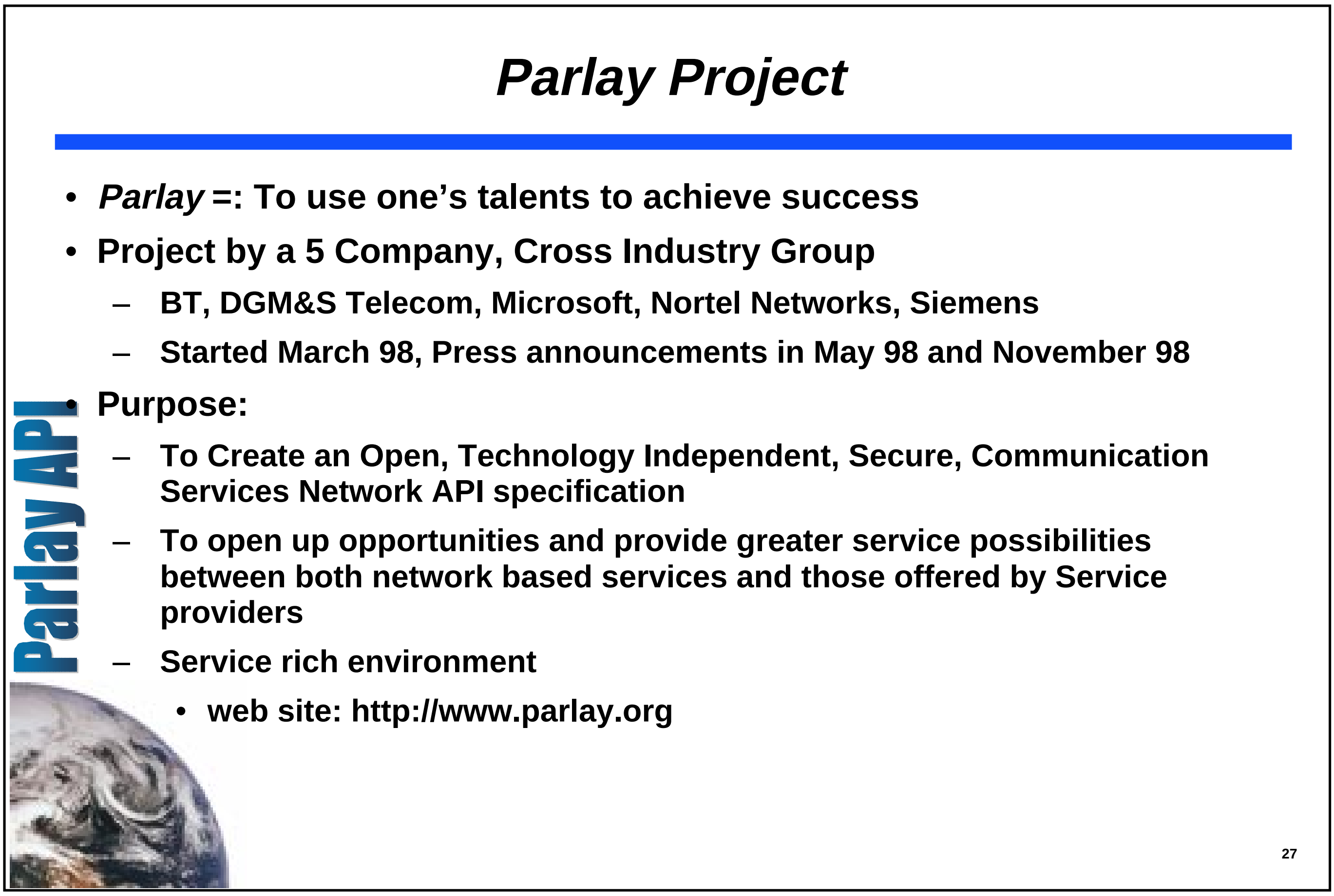
The PIN attempts to fill this void. Entities on the Internet domain can receive the events generated by the PSTN domain and act appropriately. The major entities that comprise of the PIN protocol are the PIN Gateway, the PIN Server and various PIN clients.

The Internet Draft "Need for PSTN Internet Notification (PIN) Services" is available from the official IETF directory and explains the need and gives some requirements for such requests for service and for notification.

The group will examine the sorts of Telephone Services that present requirements for such requests from a telephone network to a set of IP hosts. It is important that we agree on the scope of requests for service and notification that such services might generate. There are various working groups and protocols (most notably IPTEL, PINT, and MEGACO) which might have relevance for the sort of services the group want to enable.

The PIN mailing list is located at ietf-pin@lists.research.bell-labs.com.

Chair: Alec Brusilovsky <abrusilovsky@lucent.com>



Parlay Project

- *Parlay* =: To use one’s talents to achieve success
- Project by a 5 Company, Cross Industry Group
  - BT, DGM&S Telecom, Microsoft, Nortel Networks, Siemens
  - Started March 98, Press announcements in May 98 and November 98
- Purpose:
  - To Create an Open, Technology Independent, Secure, Communication Services Network API specification
  - To open up opportunities and provide greater service possibilities between both network based services and those offered by Service providers
  - Service rich environment
    - web site: http://www.parlay.org

BT, DGM&S Telecom, Microsoft, Nortel Networks and Siemens have formed a new working group, called the Parlay Group, to develop a specification for an open network interface that will widen the horizons of the intelligence in networks. This specification will enable service providers, ISVs and other developers in the IT and telecommunications industries to generate a new range of applications which access and use functionality resident in public and/or private communications networks while maintaining the integrity, performance, and security of those networks.

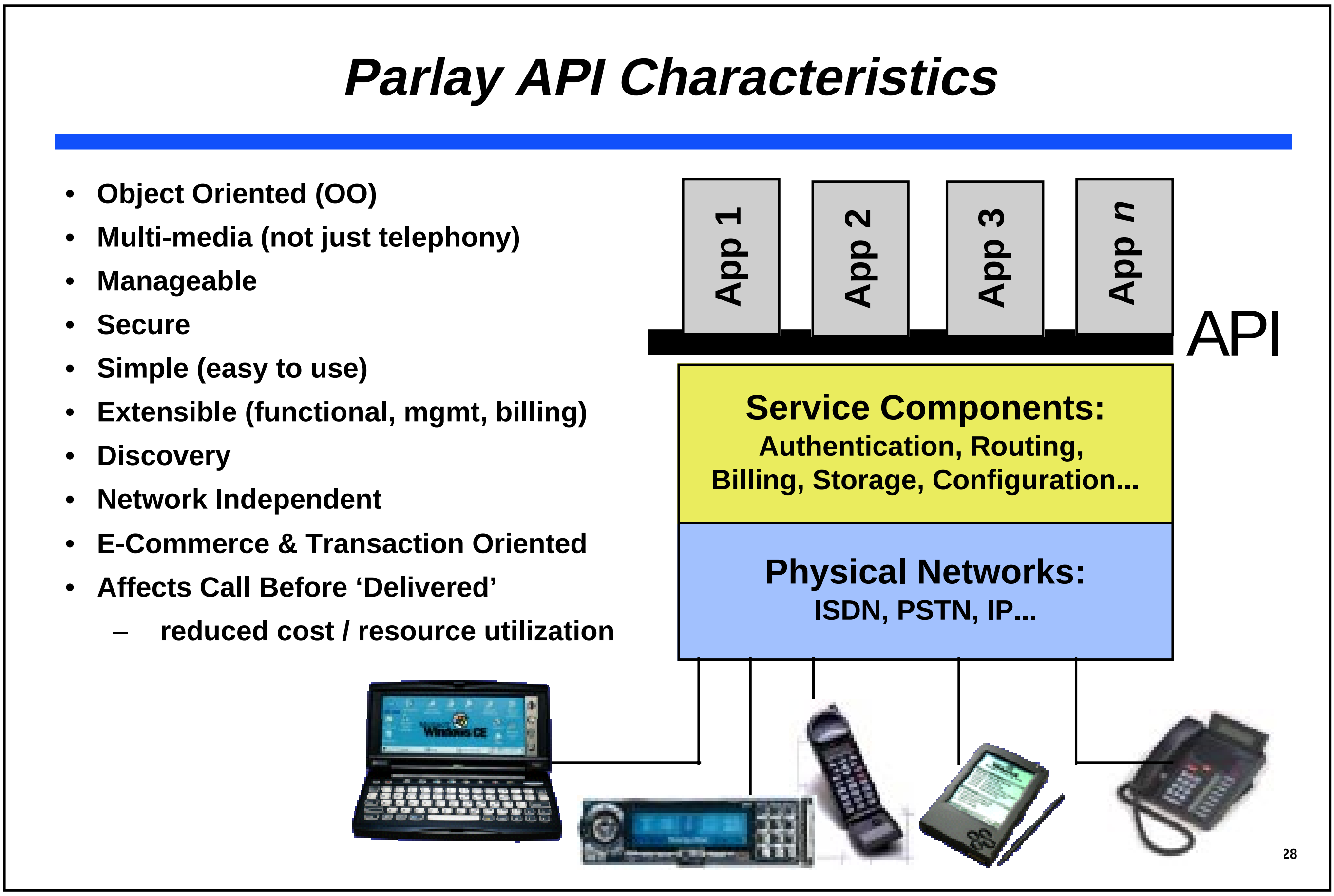
The Parlay Group will produce an open, technology-independent, and extensible API specification. This will be done quickly, in small increments, to promote open access to a wide range of network-based telecom and network control functions - many which were previously inaccessible outside of the core signalling networks.

As part of Parlay Group activities, the API specification will be publicly presented and the concept demonstrated in trial applications. Publication of the initial specification and a demonstration of its initial capabilities occurred 4 December 1998.

The API will allow secure access to core and advanced capabilities embedded in the networks of today’s telephone companies while being sufficiently adaptable to address similar capabilities in future communication technologies. The scope of the activity includes all kinds of communication capabilities such as telephone technologies (wireline, wireless, and IP telephony), as well as video and data communications.

The API will not directly open up the networks' signalling for public usage. Rather, network capabilities will be encapsulated and made visible using object technology in a secure, manageable, and billable manner. Client access to these capabilities will be realised using object access technologies such as COM or CORBA.

It is the Parlay Group's intent to bring a practical API specification to market as quickly as possible. The Parlay Group intends to develop a high-level definition of the overall Parlay API, and then identify subsets to be released and supported at frequent intervals. Each of these subsets will be ready for immediate, practical use. Additionally, the intent of the Parlay Group is to provide unique value in the API; members will maximise the use of existing technologies and member knowledge to produce the API specification in an incremental manner.



Parlay API Characteristics

- Object Oriented (OO)
- Multi-media (not just telephony)
- Manageable
- Secure
- Simple (easy to use)
- Extensible (functional, mgmt, billing)
- Discovery
- Network Independent
- E-Commerce & Transaction Oriented
- Affects Call Before 'Delivered'
  - reduced cost / resource utilization

App 1
App 2
App 3
App *n*
API

Service Components:
Authentication, Routing, Billing, Storage, Configuration...

Physical Networks:
ISDN, PSTN, IP...

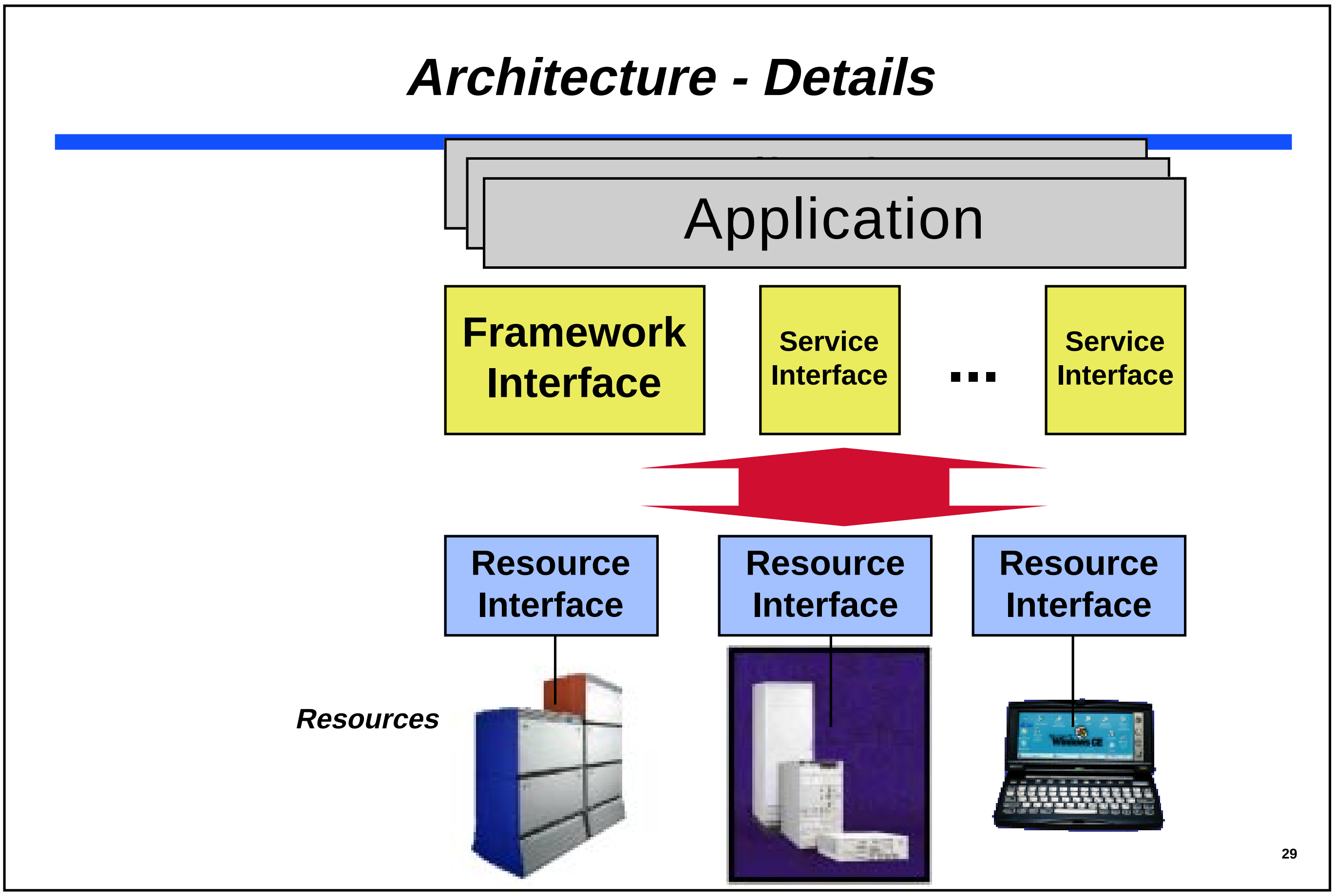
28

The API will allow secure access to core and advanced capabilities embedded in the networks of today's telephone companies while being sufficiently adaptable to address similar capabilities in future communication technologies. The scope of the activity includes all kinds of communication capabilities such as telephone technologies (wireline, wireless, and IP telephony), as well as video and data communications.

The API will not directly open up the networks' signalling for public usage. Rather, network capabilities will be encapsulated and made visible using object technology in a secure, manageable, and billable manner. Client access to these capabilities will be realised using object access technologies such as COM or CORBA.

The Parlay API will allow a wide range of new communication services to be built and deployed both within the network and outside of the network. This will yield opportunities for third parties to create a diverse set of new "value added" services and make these available to the mass market. Ultimately, this will allow customers to buy integrated communications/telephony applications focused on their own niche market at their neighbourhood computer store or download them via the Internet from their favourite Independent Software Vendor (ISV). It is possible that such services will be billed on a basis different from today's telephony services following, for example, the micro-billing concepts now being introduced for the Internet. This and other secondary effects offer potentially massive revenue enhancement opportunities in the convergence of network-based telecommunication, CTI, and IT technologies.

The Parlay Group plans to produce a non-proprietary open interface for use by anyone, including the world's carriers, equipment and technology suppliers, third party service providers, and ISVs.



Architecture - Details

Application

Framework Interface

Service Interface

...

Service Interface

Resource Interface

Resource Interface

Resource Interface

Resources

29

The API is object-oriented and consists of two categories of interface:

- Service Interfaces. These offer applications access to a range of network capabilities.
- Framework Interfaces. These provide 'surround' capabilities necessary for the Service Interfaces to be open, secure, resilient and manageable.

The services are assumed to be co-located with the framework. They are also considered trusted, such that once an application has been authorised to use particular services by the framework, then no further authorisation is required when those services are used. In some scenarios, however, additional authentication may be required for the application to use certain service features.

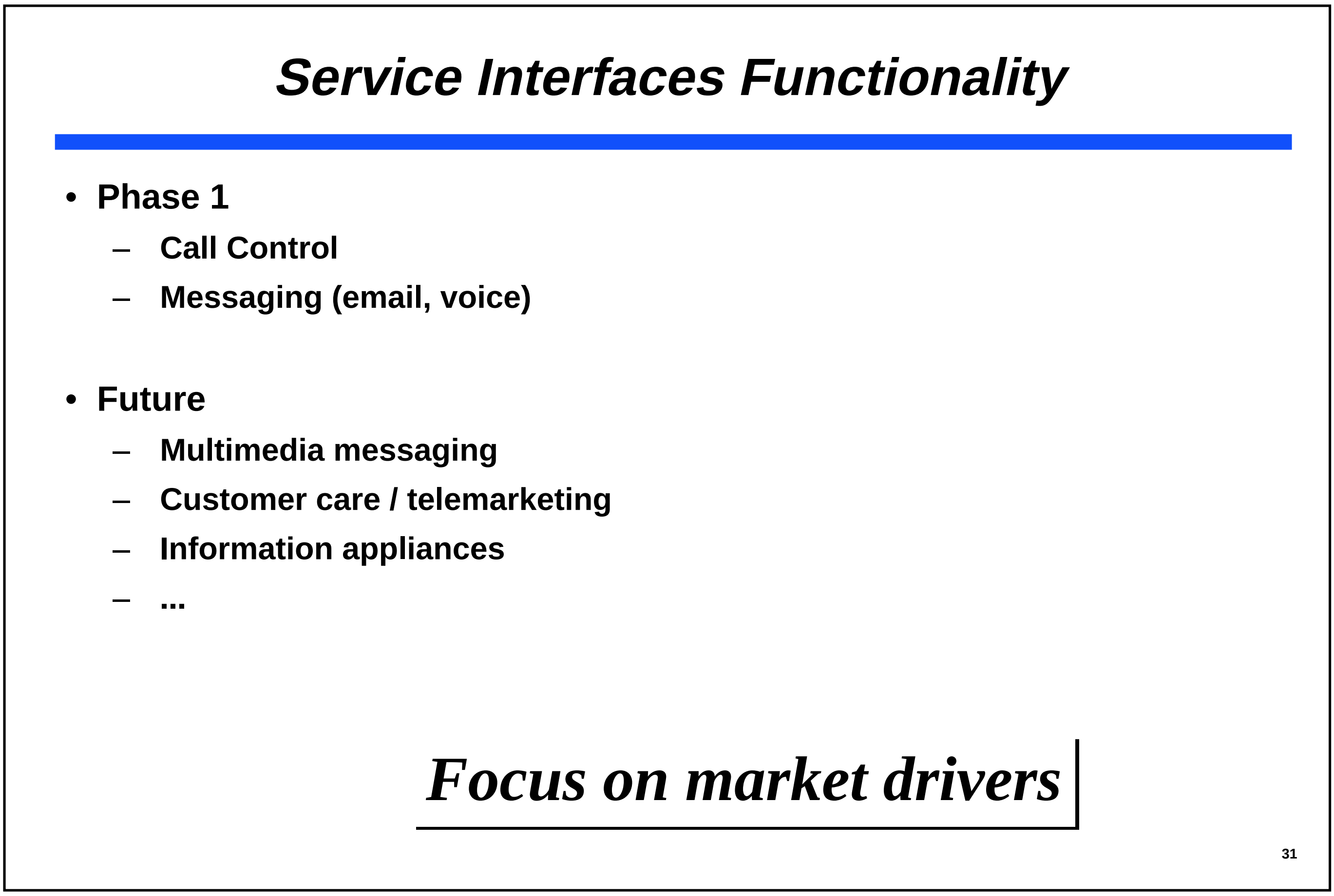
In order to realise the Service and Framework interfaces, it is recognised that categories of resource interfaces and service/framework interfaces are required to facilitate integration of network equipment and Parlay services. The definition of the resource interfaces is not in the scope of the Parlay group at this time.

The architecture of the API is described more fully in the Parlay organisation White Paper. The white paper provides more information on resource interfaces and the relationship between resource, service and framework interfaces.

Framework Interface Functionality

- **Phase 1**
  - **Authentication**
  - **Event Notification**
  - **Integrity Management**
  - **OA&M**
  - **Discovery**

- **Future**
  - **Accounting / Billing Management**
  - **Logging / Auditing**
  - **Directory Access**
  - **Configuration / Provisioning Management**
  - **Performance Management**



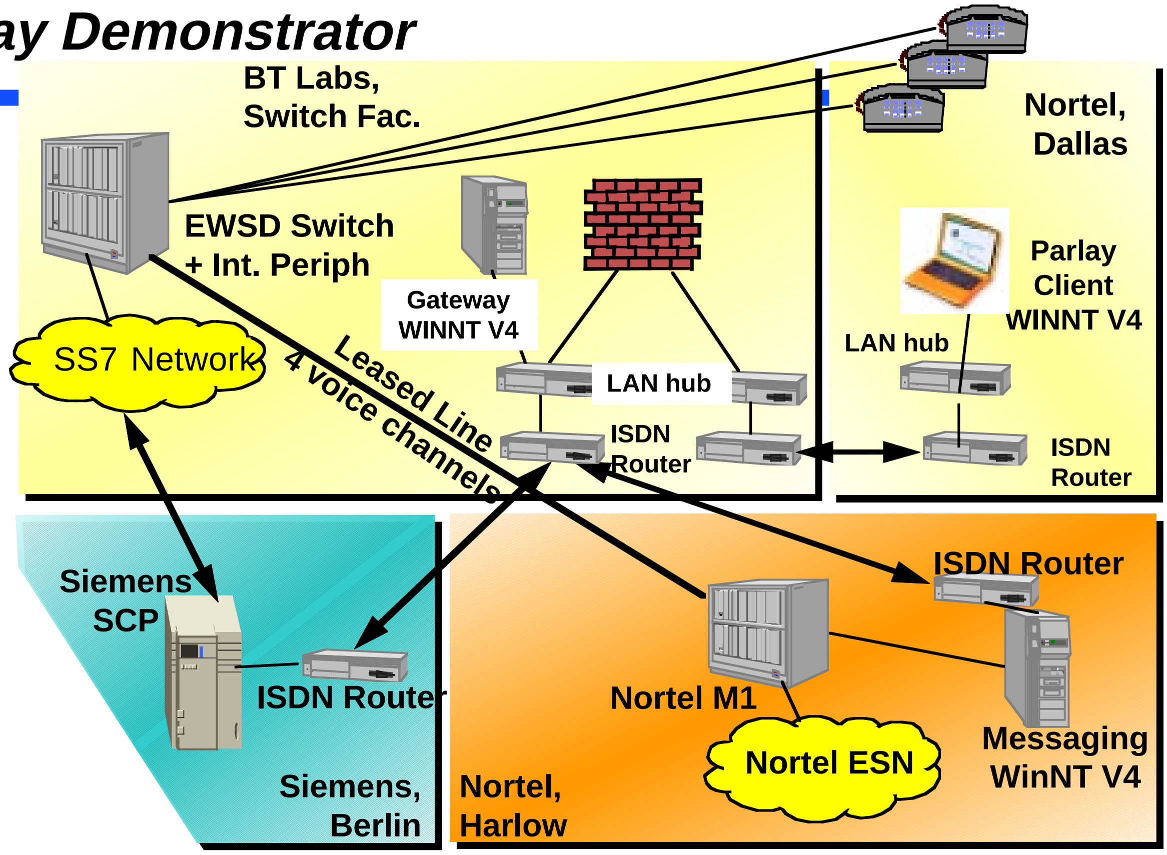
Service Interfaces Functionality

- Phase 1
  - Call Control
  - Messaging (email, voice)

- Future
  - Multimedia messaging
  - Customer care / telemarketing
  - Information appliances
  - ...

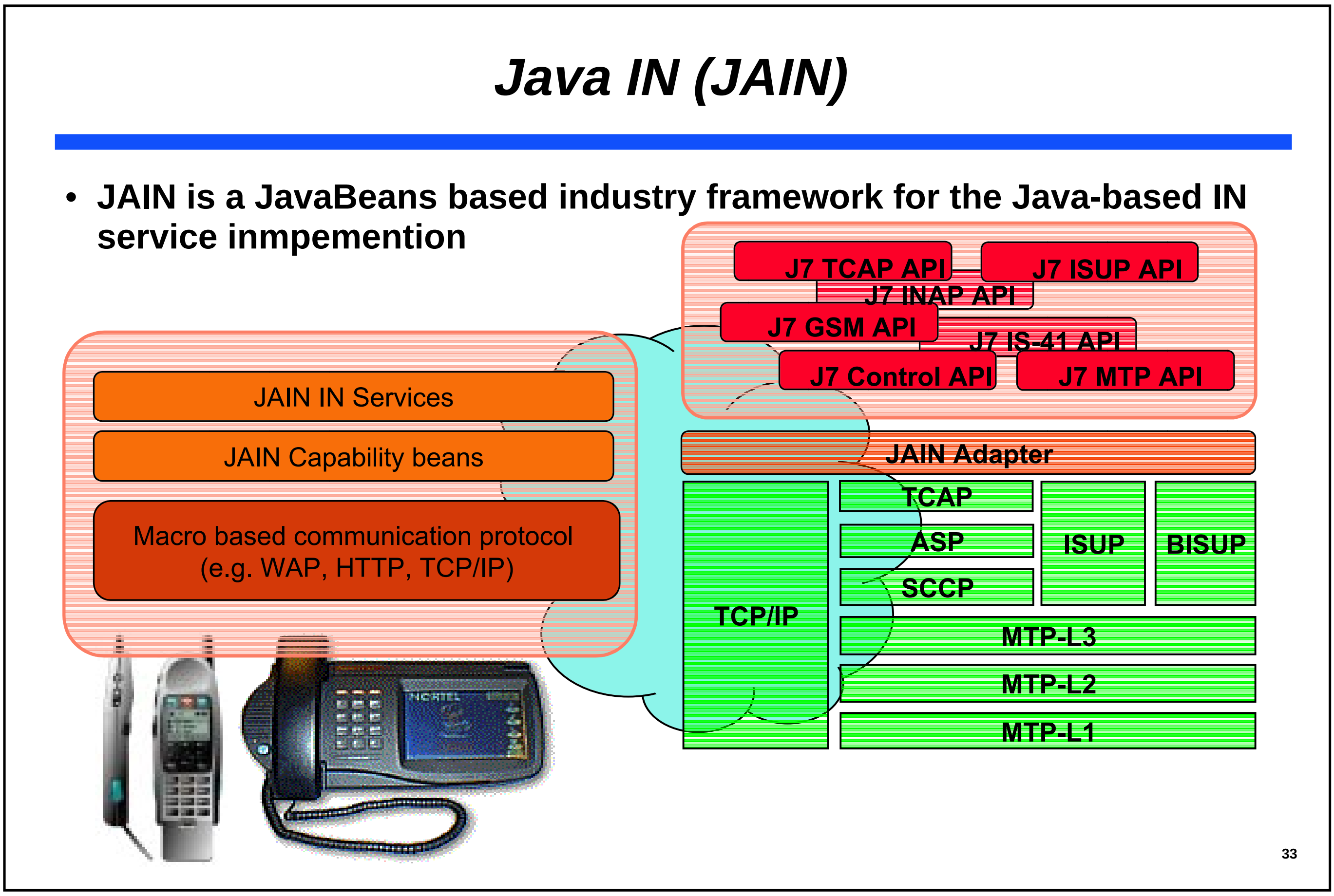
Focus on market drivers

Parlay Demonstrator



The demonstration (a fictional service called Guaranteed Call Delivery) illustrates how a customer premises service application can be built using the capabilities of today's IN infrastructure to create capabilities and advantages in a manner not previously possible.

The Guaranteed Call Delivery application is a Parlay Group API client program. It runs on a user’s computer system outside of the protected and secured telecom network environment. The database used by the Guaranteed Call Delivery application is also located at the user’s site - and remains private to the user. The Guaranteed Call Delivery program will use the Parlay API to invoke the services of the host telecom network. The program complements these functions with appropriate business logic and access to the user's database to build a completely new network application.



Java IN (JAIN)

- JAIN is a JavaBeans based industry framework for the Java-based IN service inmpemention

JAIN IN Services

JAIN Capability beans

Macro based communication protocol (e.g. WAP, HTTP, TCP/IP)

J7 TCAP API

J7 ISUP API

J7 INAP API

J7 GSM API

J7 IS-41 API

J7 Control API

J7 MTP API

JAIN Adapter

TCAP

ASP

SCCP

ISUP

BISUP

TCP/IP

MTP-L3

MTP-L2

MTP-L1

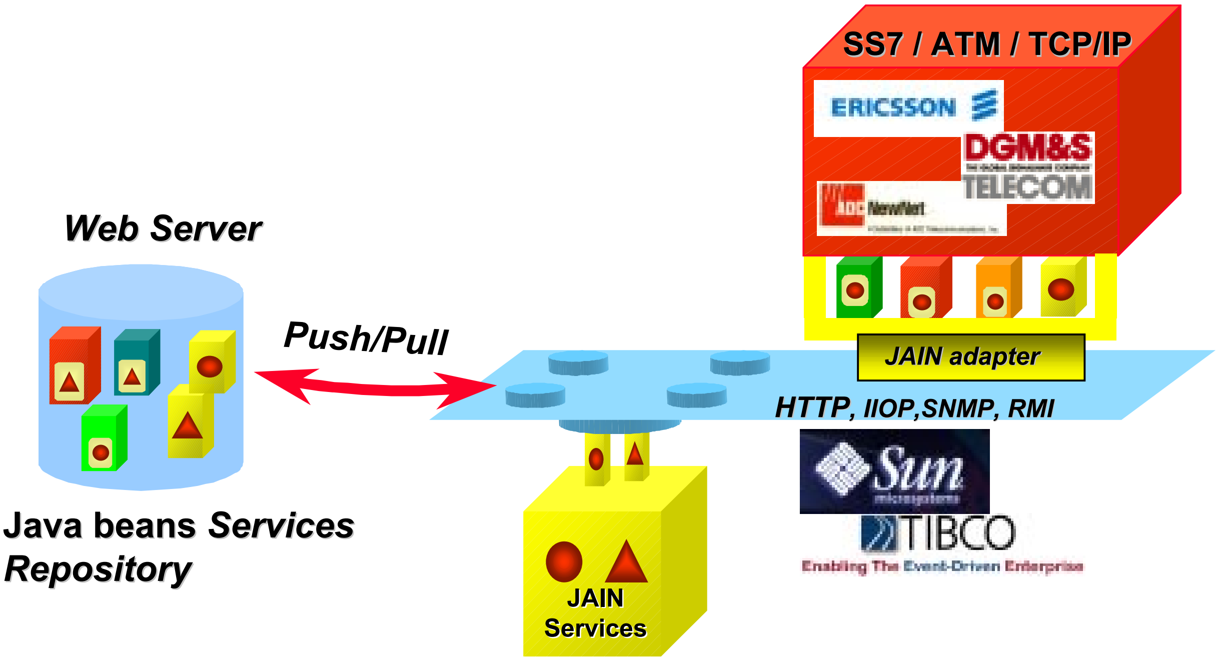
Java IN (JAIN) is an activity initiated by Sun in order to allow the use of Sun’s Java technology for providing IN services.

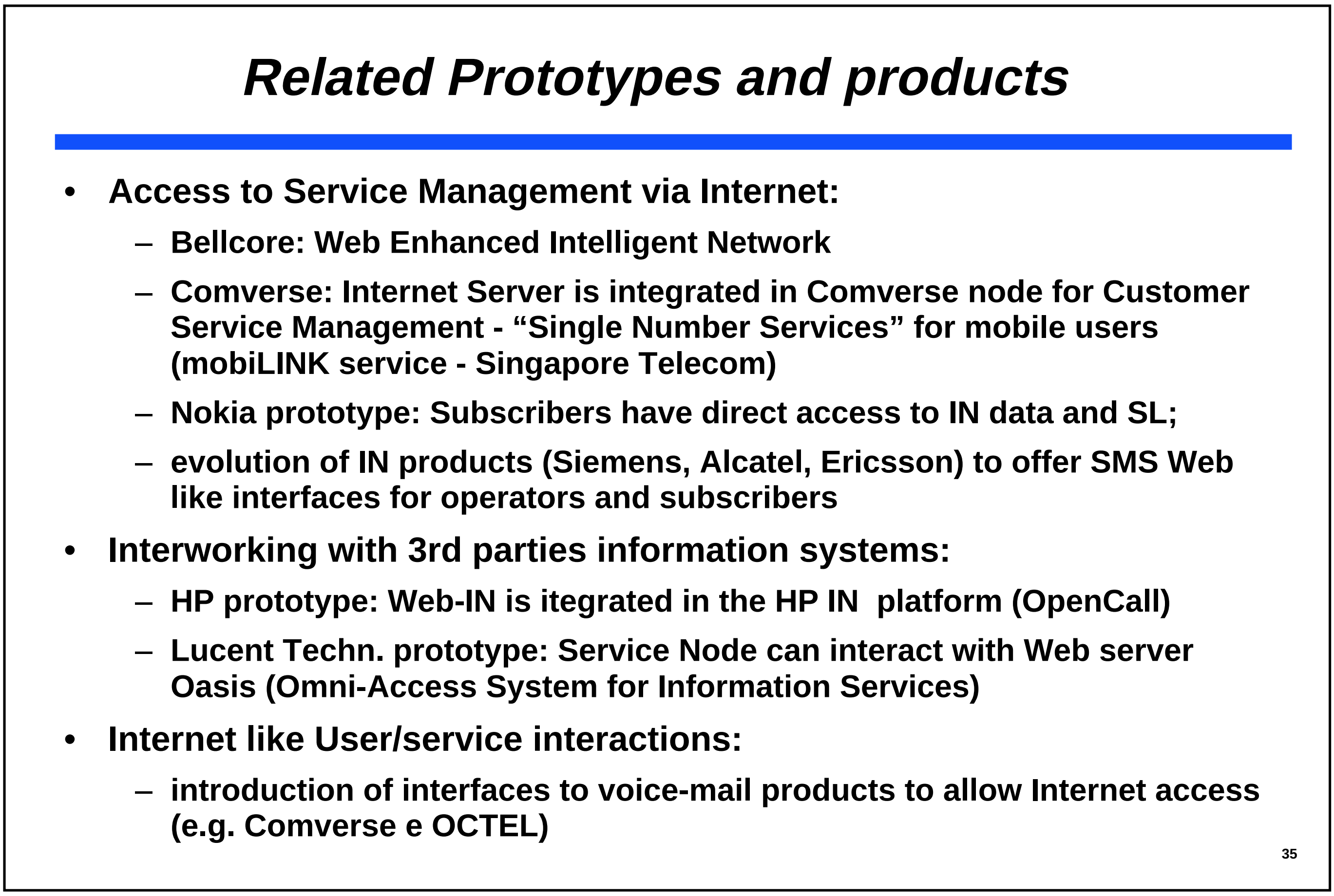
Unfortunately there is not much information available about JAIN, just an undocumented set of slides and some press releases.

So stay tuned for further information and look periodically at the SUN web server.

See “http://www.sun.com/smi/Press/sunflash/9806/sunflash.980609.9.html” for more details.

JAIN Architecture



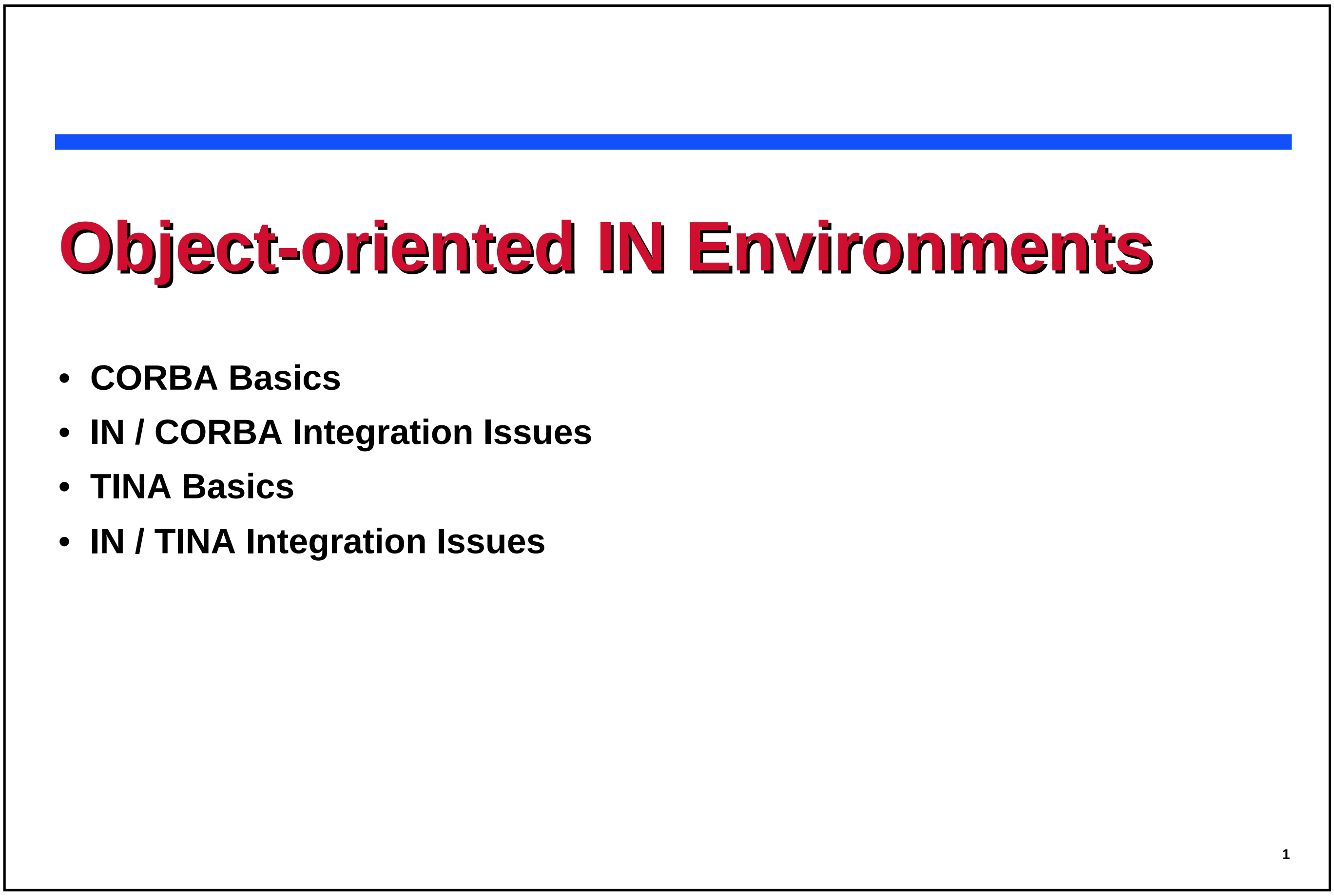


Related Prototypes and products

- Access to Service Management via Internet:
  - Bellcore: Web Enhanced Intelligent Network
  - Comverse: Internet Server is integrated in Comverse node for Customer Service Management - “Single Number Services” for mobile users (mobiLINK service - Singapore Telecom)
  - Nokia prototype: Subscribers have direct access to IN data and SL;
  - evolution of IN products (Siemens, Alcatel, Ericsson) to offer SMS Web like interfaces for operators and subscribers
- Interworking with 3rd parties information systems:
  - HP prototype: Web-IN is itegrated in the HP IN platform (OpenCall)
  - Lucent Techn. prototype: Service Node can interact with Web server Oasis (Omni-Access System for Information Services)
- Internet like User/service interactions:
  - introduction of interfaces to voice-mail products to allow Internet access (e.g. Comverse e OCTEL)

Summary

- **IN / Internetintegration is a topic in IN CS-3 standardisation**
- **Key issues to be addressed are:**
  - **SMS interface toward the internet for advanced service management**
  - **SCP interface toward the internet for 3rd party control (to PINT server)**
  - **SCP interface to H.323 Gatekeeper**
  - **SRF is becomng a key component for Unified Messaging**



Object-oriented IN Environments

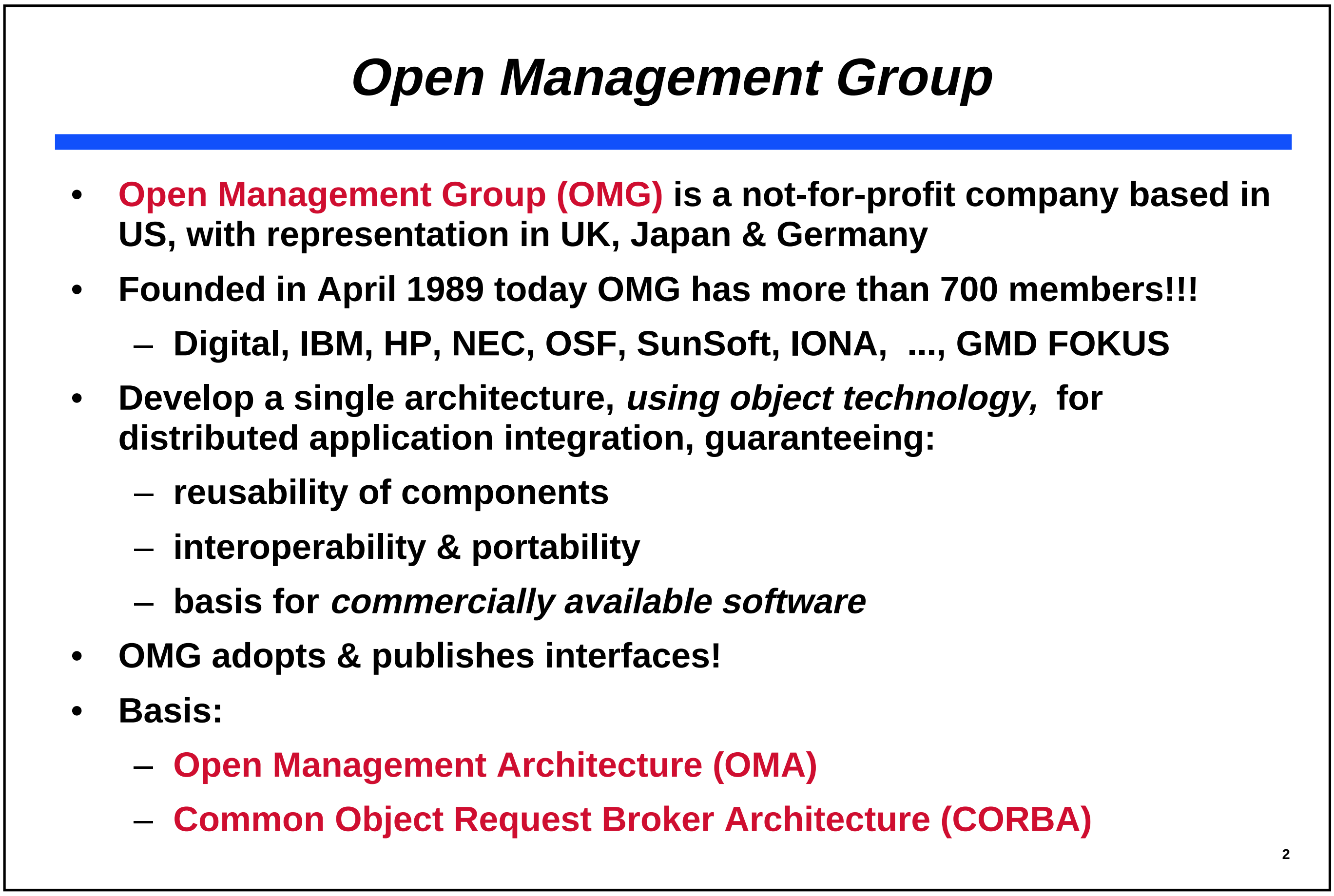
- CORBA Basics
- IN / CORBA Integration Issues
- TINA Basics
- IN / TINA Integration Issues

1

This part investigates the impact of new object-oriented middleware technologies and new open service archiectures on IN.

In this context the Open Management Group’s (OMG’s) Common Object Request Broker Architecture (CORBA) is becoming the defacto standard for distributed object computing middleware gaining importance in telecommunications. We provide an overview of CORBA and outline the current IN/CORBA integration activities undertaken by the OMG.

Furthermore, we provide an overview of the Telecommunications Information Networking Architecture (TINA), gaining importance as universal open service platform for the near future, and address potential migration steps from IN-based toward TINA-based platforms.



Open Management Group

- **Open Management Group (OMG) is a not-for-profit company based in US, with representation in UK, Japan & Germany**
- **Founded in April 1989 today OMG has more than 700 members!!!**
  - **Digital, IBM, HP, NEC, OSF, SunSoft, IONA, ..., GMD FOKUS**
- **Develop a single architecture, *using object technology,* for distributed application integration, guaranteeing:**
  - **reusability of components**
  - **interoperability & portability**
  - **basis for *commercially available software***
- **OMG adopts & publishes interfaces!**
- **Basis:**
  - **Open Management Architecture (OMA)**
  - **Common Object Request Broker Architecture (CORBA)**

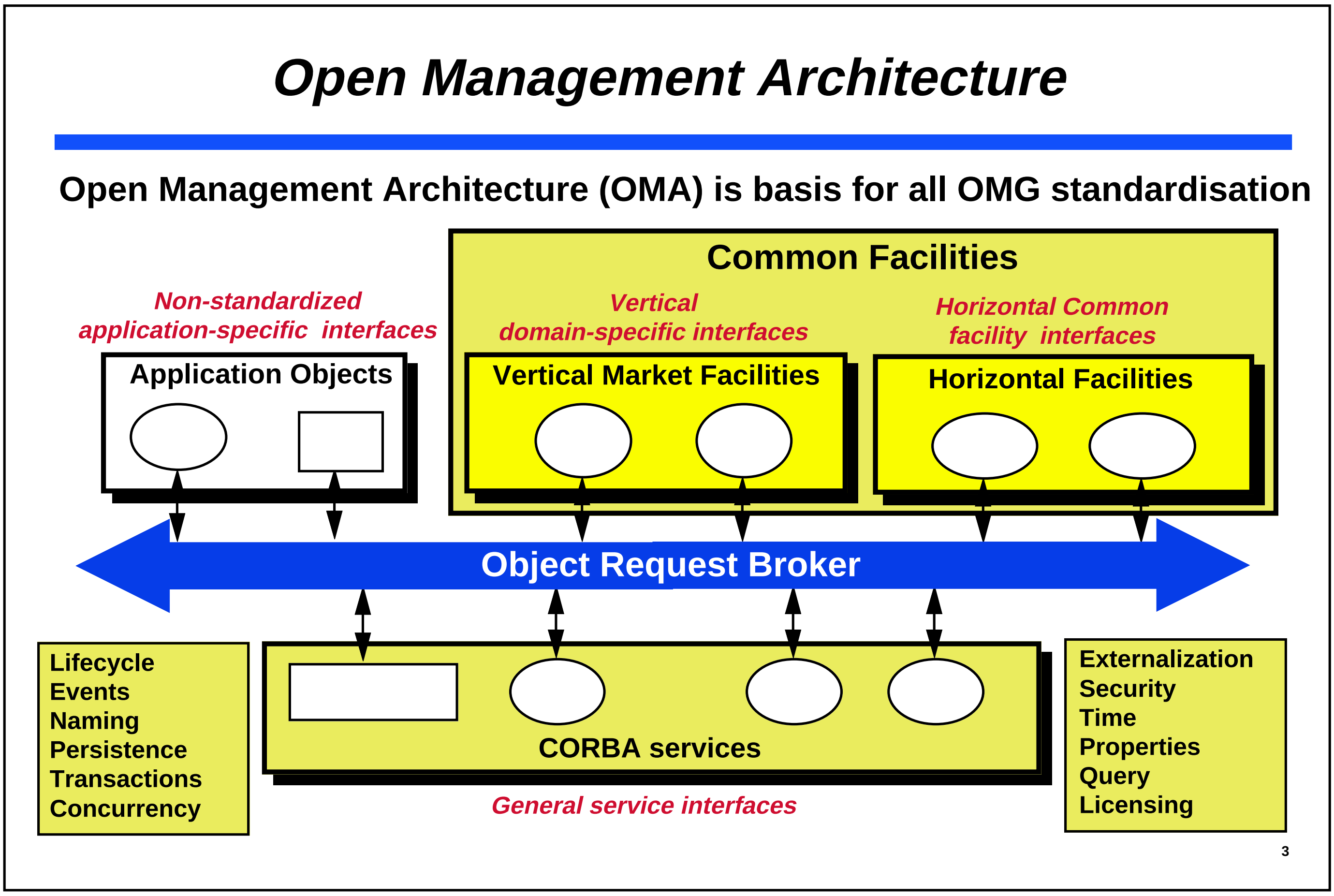
2

The Open Management Group (OMG) was founded in May 1989 by eight companies: 3Com Corporation, American Airlines, Canon, Inc., Data General, Hewlett-Packard, Philips Telecommunications N.V., Sun Microsystems and Unisys Corporation. In October 1989, OMG began independent operations as a non-profit corporation. OMG is developing the "Architecture for a Connected World" through its world-wide standard specifications: CORBA/IIOP, Object Services, Internet Facilities and Domain Interface specifications. OMG is headquartered in Framingham, Massachusetts, USA, with international marketing partners in the UK, Germany, Japan, India and Australia.

The Object Management Group has continued to build upon its success and its relevance during the last year as demonstrated by its ever increasing membership (now close to 760 companies) and with on average 5 new companies joining each week. It has extended its work area from looking at generic middle-ware issues, to attempting to address some specific requirements of vertical industries such as Manufacturing, Accountancy and Finance, Electronic Commerce, and or course Telecommunications. The OMG brings together companies and technical experts from all aspects of industry from independent software and hardware vendors to large systems vendors, to global service providers.

For general information on OMG see „http://www.omg.org“.

For CORBA beginners we recommend “http://www.omg.org/news/begin.htm”.



Open Management Architecture

Open Management Architecture (OMA) is basis for all OMG standardisation

Common Facilities

Non-standardized application-specific interfaces

Vertical domain-specific interfaces

Horizontal Common facility interfaces

Application Objects

Vertical Market Facilities

Horizontal Facilities

Object Request Broker

Lifecycle
Events
Naming
Persistence
Transactions
Concurrency

CORBA services

General service interfaces

Externalization
Security
Time
Properties
Query
Licensing

3

The Open Management Architecture (OMA) comprises:

1. Object Request Broker (known as CORBA) providing a runtime system for the execution of client and server objects and their communication. The ORB relays the invocation from the client to the object implementation on the server side and the result back to the client. It lets objects make requests transparently to other locally or remotely located objects.

2. CORBA Services providing necessary basic capabilities. CORBA Object Services are collections of system-level services, specified in Common Object Services Specification (COSS) of the OMG. They augment and complement the functionality of the ORB and can be used to create a component, name it, and introduce it into the environment.

3. Common Facilities representing standardized building blocks, which include specifications for higher level services and vertical market speciality areas. They are separated into Horizontal Common Facilities and Vertical Market Facilities.

a) Horizontal Common Facilities including higher level services shared by many or most systems, regardless of application content area. Four major sets are identified: User Interface, Information Management, Systems Management and Task Management.

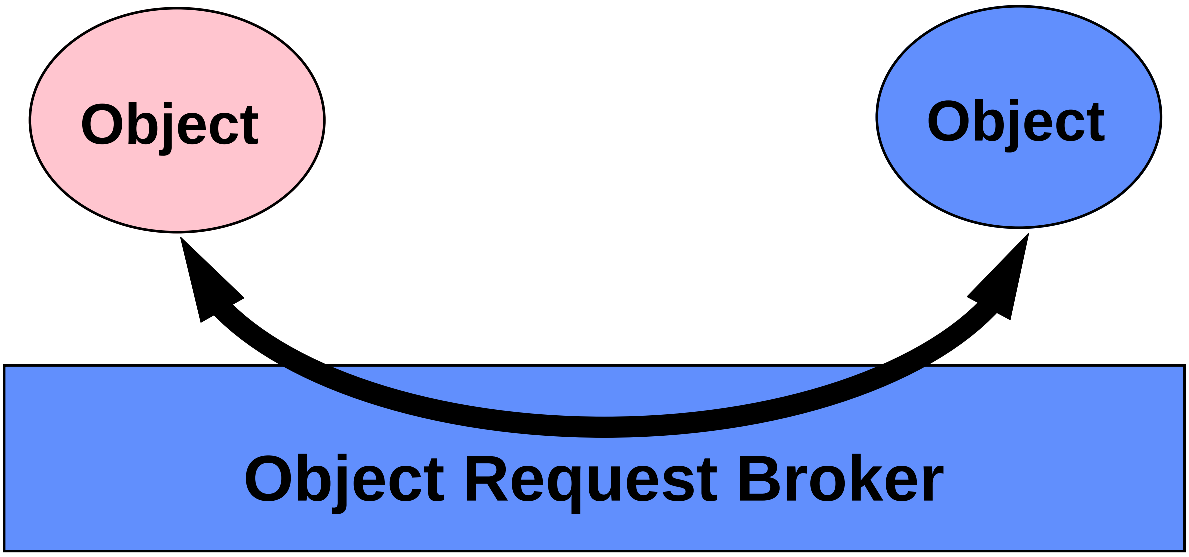
b) Vertical Market Facilities representing standardized application areas. They support the domain-specific tasks associated with vertical market segments, representing standards for interoperability in particular speciality areas, e.g. Computer Integrated Manufacturing (CIM). Each speciality area should represent the needs of an important computing market. The number of Vertical Market Facilities is not limited.

4. Application Objects created and developed by developers. Application Objects are components specific to end-user applications representing an enterprise model. An application is typically built from a number of cooperating business components that together serve a specific purpose. These application objects are built on top of services provided by the ORB, the Common Facilities and the Common Object Services.

Note that all standardization within OMG populates the OMA!

An introduction to OMA can be found under “http://www.omg.org/about/omaov.htm”

CORBA Overview

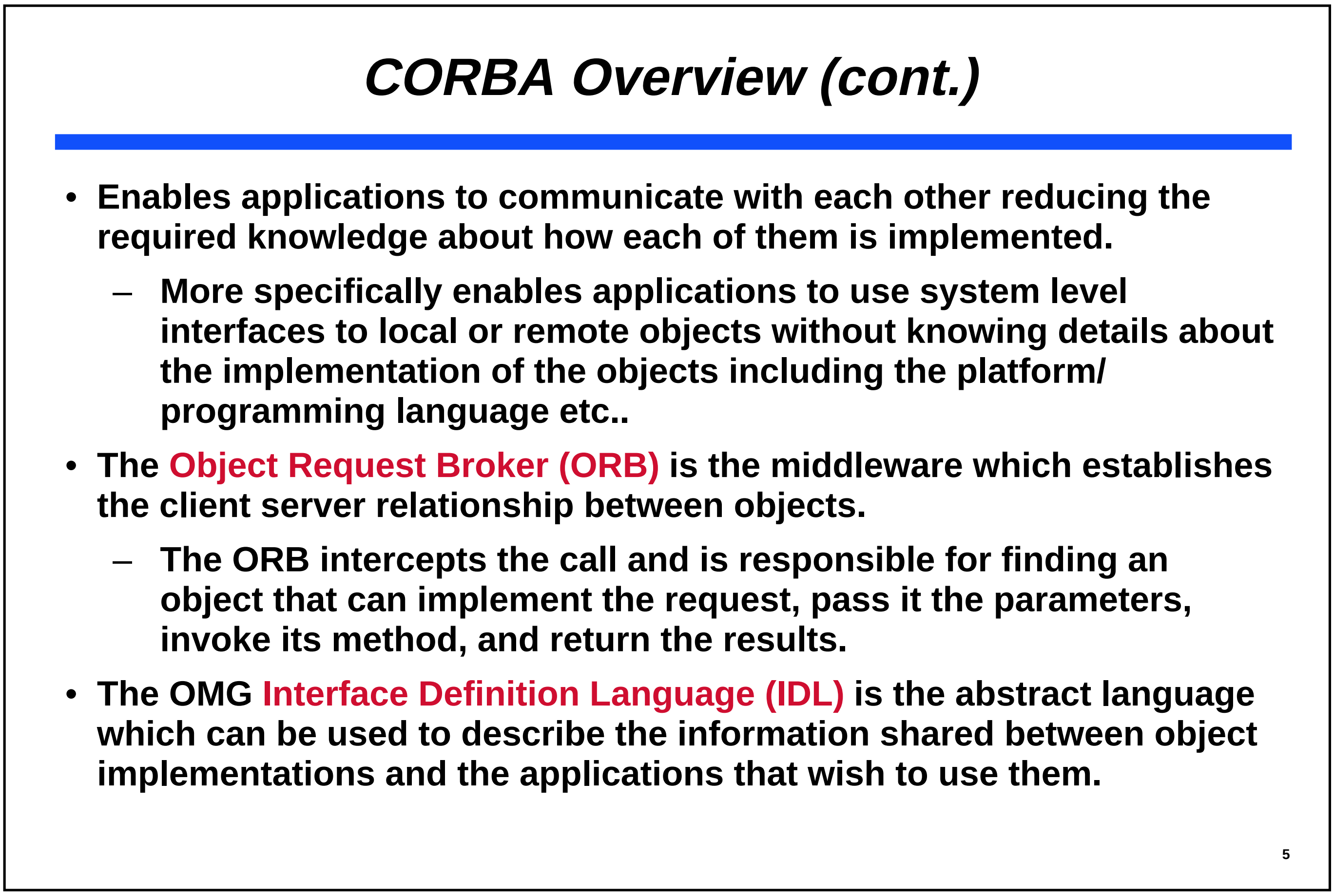


Object

Object

Object Request Broker

The Object Request Broker (ORB) is the middleware that establishes the client-server relationships between objects. Using an ORB, a client can transparently invoke a method on a server object, which can be on the same machine or across a network. The ORB intercepts the call and is responsible for finding an object that can implement the request, pass it the parameters, invoke its method, and return the results. The client does not have to be aware of where the object is located, its programming language, its operating system,or any other system aspects that are not part of an object's interface. In so doing, the ORB provides interoperability between applications on different machines in heterogeneous distributed environments and seamlessly interconnects multiple object systems.



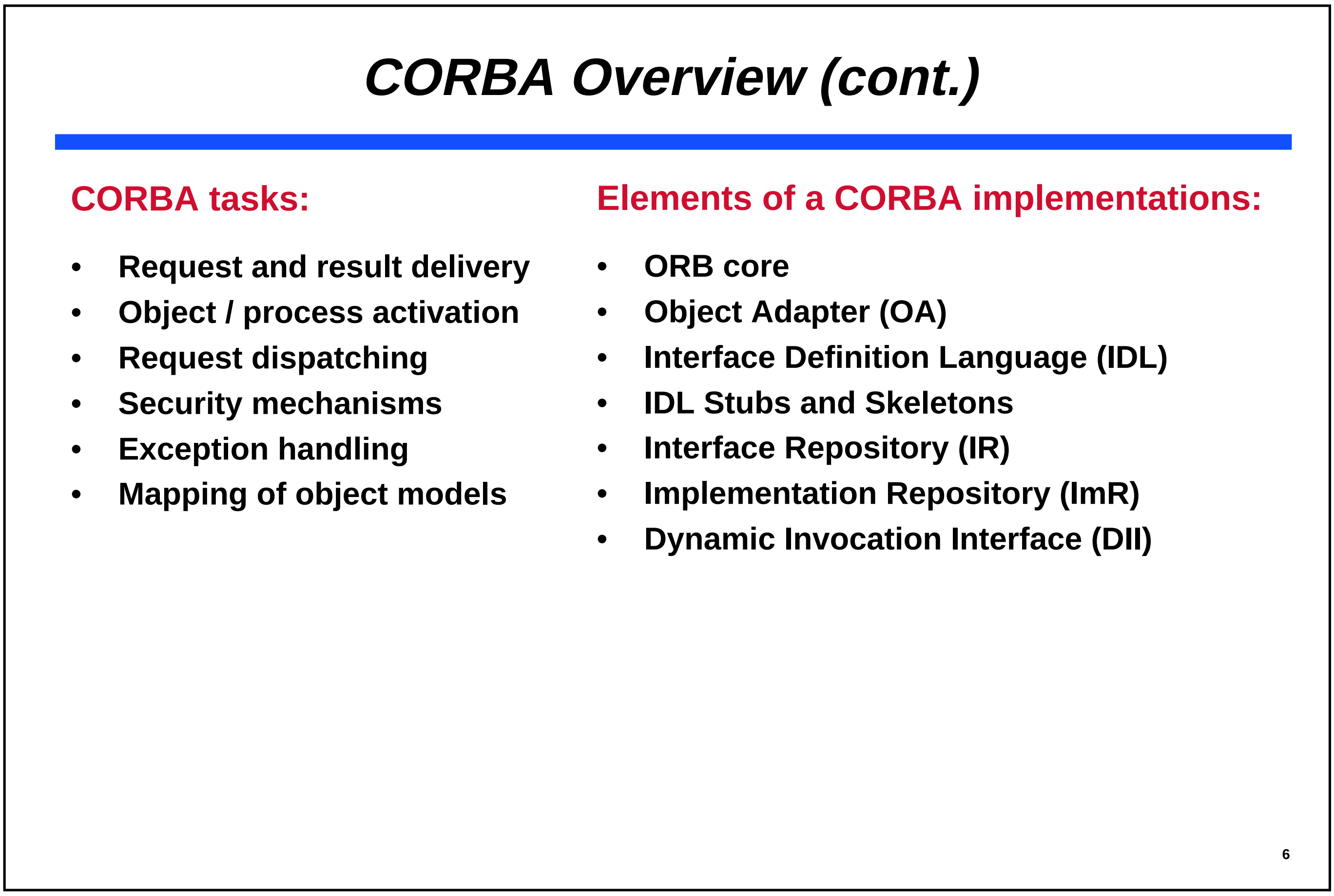
CORBA Overview (cont.)

- Enables applications to communicate with each other reducing the required knowledge about how each of them is implemented.
  - More specifically enables applications to use system level interfaces to local or remote objects without knowing details about the implementation of the objects including the platform/ programming language etc..
- The Object Request Broker (ORB) is the middleware which establishes the client server relationship between objects.
  - The ORB intercepts the call and is responsible for finding an object that can implement the request, pass it the parameters, invoke its method, and return the results.
- The OMG Interface Definition Language (IDL) is the abstract language which can be used to describe the information shared between object implementations and the applications that wish to use them.

The Common Object Request Broker Architecture (CORBA), is the Object Management Group's answer to the need for interoperability among the rapidly proliferating number of hardware and software products available today. Simply stated, CORBA allows applications to communicate with one another no matter where they are located or who has designed them. CORBA 1.1 was introduced in 1991 by Object Management Group (OMG) and defined the Interface Definition Language (IDL) and the Application Programming Interfaces (API) that enable client/server object interaction within a specific implementation of an Object Request Broker (ORB). CORBA 2.0, adopted in December of 1994, defines true interoperability by specifying how ORBs from different vendors can interoperate.

In fielding typical client/server applications, developers use their own design or a recognized standard to define the protocol to be used between the devices. Protocol definition depends on the implementation language, network transport and a dozen other factors. ORBs simplify this process. With an ORB, the protocol is defined through the application interfaces via a single implementation language-independent specification, theInterface Definition Language (IDL). And ORBs provide flexibility. They let programmers choose the most appropriate operating system, execution environment and even programming language to use for each component of a system under construction. More importantly, they allow the integration of existing components. In an ORB-based solution,developers simply model the legacy component using the same IDL they use for creating new objects, then write "wrapper" code that translates between the standardized bus and the legacy interfaces.

A tutorial on CORBA can be found under “http://www.omg.org/about/wicorba.htm”



CORBA Overview (cont.)

CORBA tasks:

- Request and result delivery
- Object / process activation
- Request dispatching
- Security mechanisms
- Exception handling
- Mapping of object models

Elements of a CORBA implementations:

- ORB core
- Object Adapter (OA)
- Interface Definition Language (IDL)
- IDL Stubs and Skeletons
- Interface Repository (IR)
- Implementation Repository (ImR)
- Dynamic Invocation Interface (DII)

6

ORB Core
The ORB Core is that part of the ORB that provides the basic representation of objects and communication of requests. The components above the ORB Core provide interfaces that can mask the differences between ORB Cores.

Object Adapter (OA)
An object adapter is the interface between the ORB core and the object implementation, the primary way that an object implementation accesses services provided by the ORB. Three kinds are identified: the Basic Object Adapter, which can be used for most ORB objects with conventional implementations. The Library Object Adapter, primary used for objects that have library implementations. Finally the Object-Oriented Database Adapter, which uses the connection to an object-oriented database to provide access to the objects stored in it.

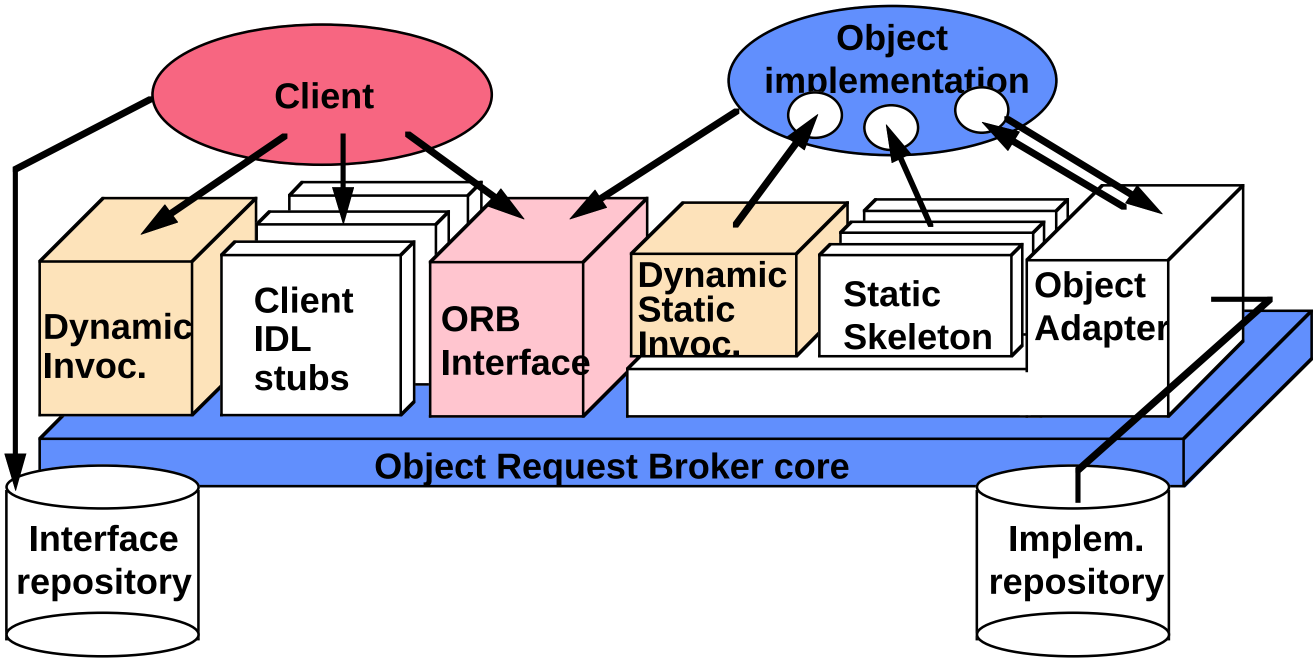
The OMG Interface Definition Language (IDL)
The Interface Definition Language (IDL) defines the types of objects by specifying their interfaces, which consist of a set of named operations and their parameters. By this, IDL describes the interface between the client and the object implementation (often called server).

IDL Stubs and Skeletons
They are statically generated from the IDL interface definition. Based on an OMG defined language mappings from/to IDL, at compile-time the Stubs are generated in the programming language of the Client (f. ex. Java), while the Skeletons are generated into the programming language the object implementation (server) is implemented (f. ex. C++).

Dynamic Invocation Interface (DII)
The DII allows for the client's side the dynamic creation and invocation of request to objects. No information about the type of object is needed at compile-time.

Dynamic Skeleton Interface (DSI)
The DSI is the dynamic interface for the server. It delivers requests from an ORB to an object implementation that at compile-time has no knowledge of the type of object it is implementing.

CORBA Overview (cont.)



7

Client Object

Represents the Client object, which requests a server object.

Server Object

Represents one object, which can be requested by client objects. Server objects are part of the Object Implementation (server).

IDL Interface Definition

The Interface Definition contains information about supported operations, the types of their parameters, exceptions which might be raised and context information the interface definition may use.

Interface Repository

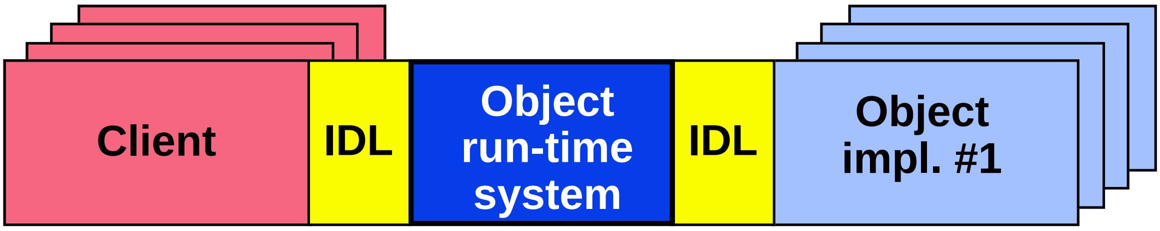
The Interface Repository provides persistent storage of interfaces, specified in IDL.
It manages and provides the access to a collection of interface definitions.

Implementation Repository

The Implementation Repository contains information that allows the ORB to locate and activate object implementations. Usually control of policies related to the activation and execution of object implementations as well as the installation of implementations is done via operations of the Implementation Repository. Further the IR is the common place to store information like debugging information, administrative control, resource allocation and security related to ORB objects.

Interface Definition Language

- Separates the Interface from the Implementation
  - multiple-inheritance, strongly typed, public interface specification language
  - independent of any particular language/compiler
  - mappings will be provided for many languages/compilers (C++, Smalltalk, Java, ...)
  - *not* a programming language!
- Enables Interoperability



8

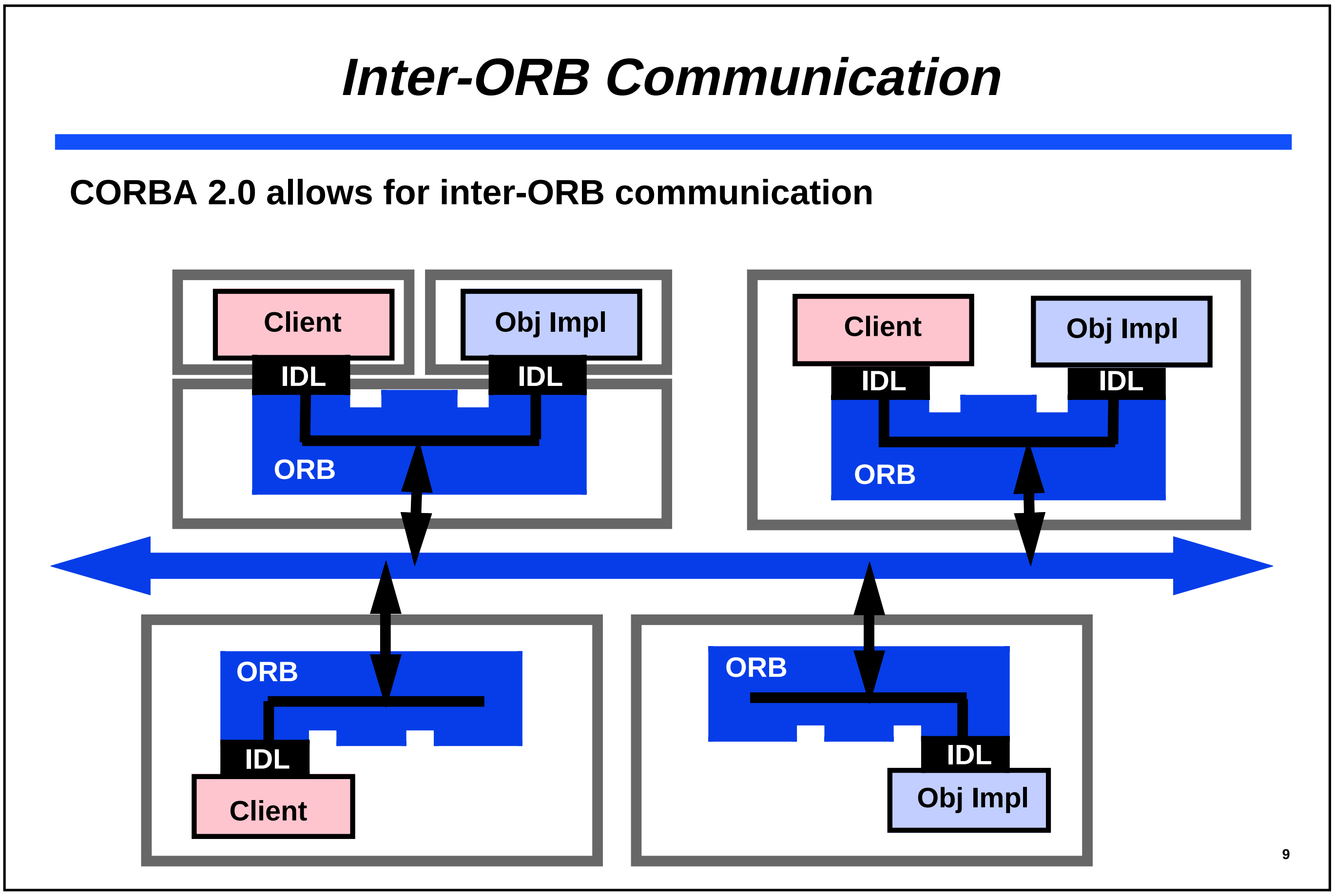
Communication of Client and Object Implementation

A request of a client is made via a Dynamic Invocation Interface or a Stub. The ORB is responsible for locating the object implementation and for transporting the data from the client to the server independent of location, implementation and host. The result of the request is returned to the client in the same way through the ORB. The server uses therefore the Dynamic Skeleton Interface or the static Skeleton (at compile-time generated from IDL interface definition).

Only one IDL interface definition is needed to generate the Client Java-Stub as well as the Skeleton for the Object Implementation(s).

Clients can use this Java-Stub to request the Object Implementation(s).

Note, the Java mapping is standardized by OMG since September 1997. Older versions of Java-ORBs may not be conform to it; this reduces the portability of the produced Java source code including the generated Java Stubs and Skeletons.



Inter-ORB Communication

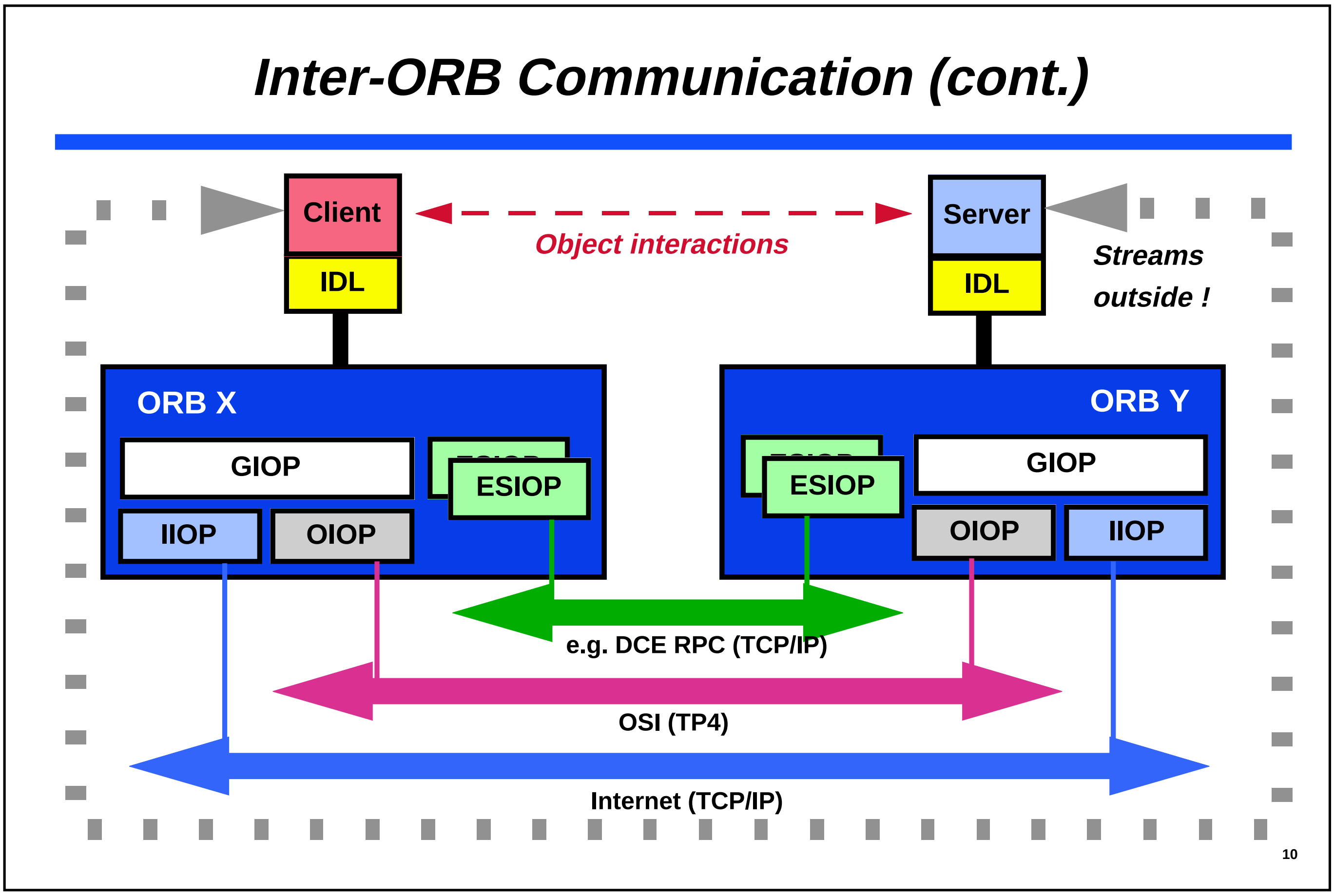
CORBA 2.0 allows for inter-ORB communication

Possibilities of Inter-ORB-Communication

1. Client and Object Implementation are implemented by two different programming languages. If the ORB supports the IDL-mapping into both languages they can use the same ORB to communicate. For example: a Java-Client can communicate with a C++ Object Implementation. The ORB supports the IDL/C++ and the IDL/JAVA mapping.

2. Client and Object implementation are implemented in the same programming language. To communicate, they can use the same ORB, which supports the IDL-Mapping of this language.

3. Client and Object Implementation can communicate over different ORBs, using one of the OMG specified Inter-ORB protocol. Client and Object Implementation can be implemented in the same or in different programming languages.



Inter-ORB Communication (cont.)

Client

IDL

Object interactions

Server

IDL

Streams outside !

ORB X

GIOP

ESIOP

IIOP

OIOP

ORB Y

ESIOP

GIOP

OIOP

IIOP

e.g. DCE RPC (TCP/IP)

OSI (TP4)

Internet (TCP/IP)

10

The Inter-ORB Communication Architecture

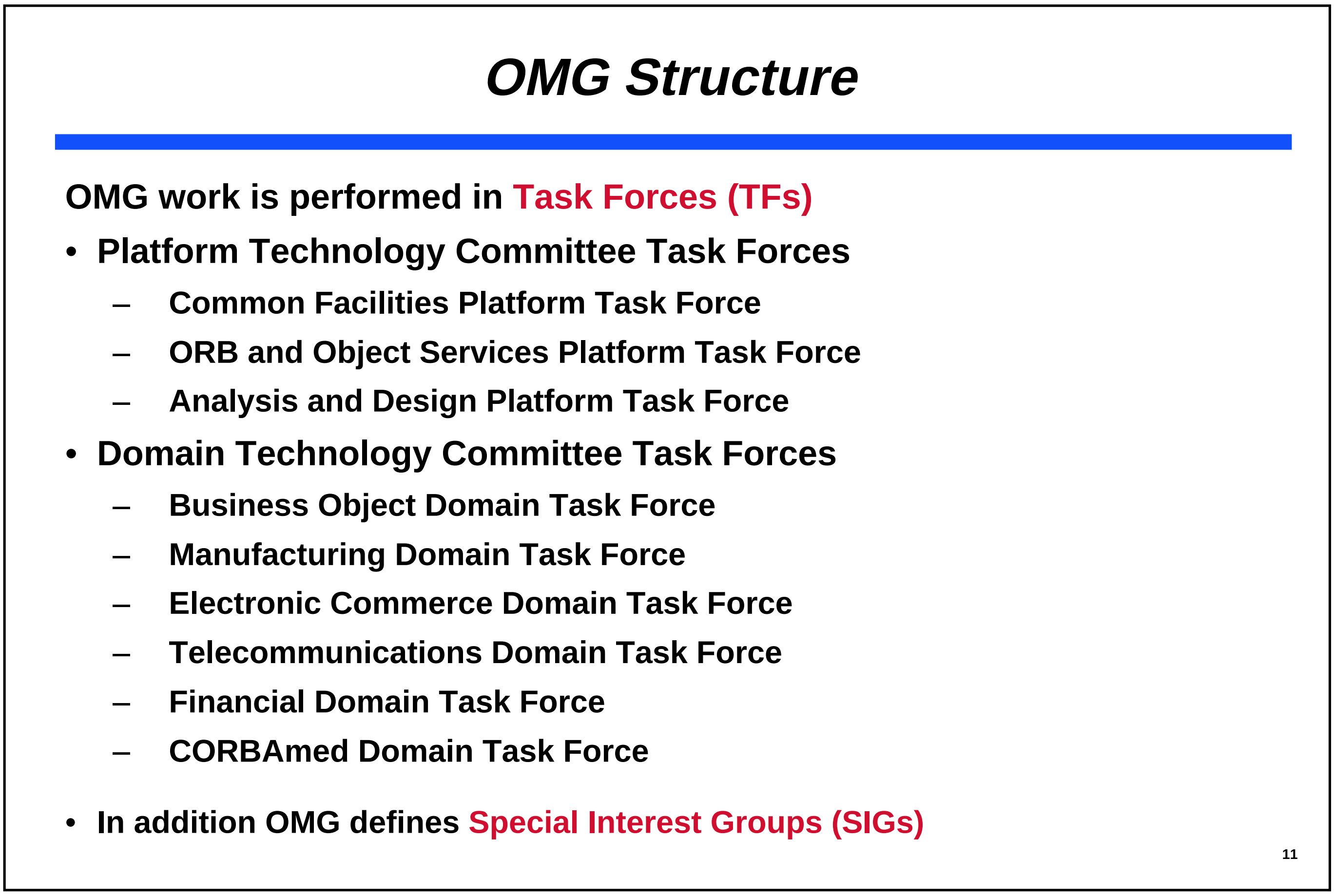
To communicate between different ORB implementations, OMG defines special protocols which have to be used. These protocols are the Internet Inter-ORB Protocol (IIOP) and the Environment-Specific Inter-ORB Protocol (ESIOP). To pass object references between different ORB, a special object reference has to be used, the Interoperable Object Reference. CORBA 2.0 defines the interoperability between ORBs of different vendors by specifying a mandatory Internet Inter-ORB Protocol (IIOP). The IIOP is basically TCP/IP with some CORBA-defined message exchanges that serve as a common backbone protocol. To be considered CORBA compliant an ORB must either implement IIOP natively or provide a `half-bridge' to it.

IOR: GIOP also defines a format for Interoperable Object References (IORs). To pass an object reference between ORBs, an IOR must be created. An IOR describes how an object is to be contacted using a special ORB mechanism. An IOR includes self-describing data that identifies the ORB domain to which a reference is associated and the protocols it supports.

GIOP: For the communication between ORBs, a specially built Protocol is defined, the General Inter-ORB Protocol (GIOP). It specifies a standard transfer syntax (low-data representation) and a set of message formats for the Inter-ORB communication. GIOP is designed to work directly over any connection-oriented transport protocol and defines seven message formats that cover all the ORB request and reply semantics. Therefore no format negotiations are needed. Versions of the GIOP running on different transports would not be directly interoperable, but their commonality would allow efficient and easy bridging between such networking domains.

IIOP: The Internet Inter-ORB Protocol (IIOP) specifies how GIOP messages are exchanged over a TCP/IP network. The IIOP makes it possible to use the Internet itself as a backbone ORB through which other ORBs can bridge. To be CORBA 2.0 compliant, an ORB must support GIOP over TCP/IP or connect to it via a half-bridge. The IIOP can also be used as protocol between half-bridges.

Note that all these protocols are used for transfering non-stream information, i.e. the transfer of audio or video information between client and server has to be performed outside the ORB(s), where only the “signalling information” is tranfered through the ORB(s).



OMG Structure

OMG work is performed in Task Forces (TFs)

- **Platform Technology Committee Task Forces**
  - **Common Facilities Platform Task Force**
  - **ORB and Object Services Platform Task Force**
  - **Analysis and Design Platform Task Force**
- **Domain Technology Committee Task Forces**
  - **Business Object Domain Task Force**
  - **Manufacturing Domain Task Force**
  - **Electronic Commerce Domain Task Force**
  - **Telecommunications Domain Task Force**
  - **Financial Domain Task Force**
  - **CORBAmed Domain Task Force**
- **In addition OMG defines Special Interest Groups (SIGs)**

The basic structure of OMG are Task Forces. The Task Forces are chartered by the OMG Technical Committee (TC) for the purpose of solving some particular problem or problems in a particular area for recommendation to the TC. One of the specific purposes of the Task Forces is to generate Requests for Information (RFIs) or Requests for Proposals (RFPs) and to evaluate the responses. Task Forces generally meet when the Technical Committee meets every six to eight weeks.

Currently there are six Domain Task Forces and three Platform Task Forces in the OMG:

I. Platform Technology Committee Task Forces

• Common Facilities Platform Task Force

• ORB and Object Services Platform Task Force

• Analysis and Design Platform Task Force

II. Domain Technology Committee Task Forces

• Business Object Domain Task Force

• Manufacturing Domain Task Force

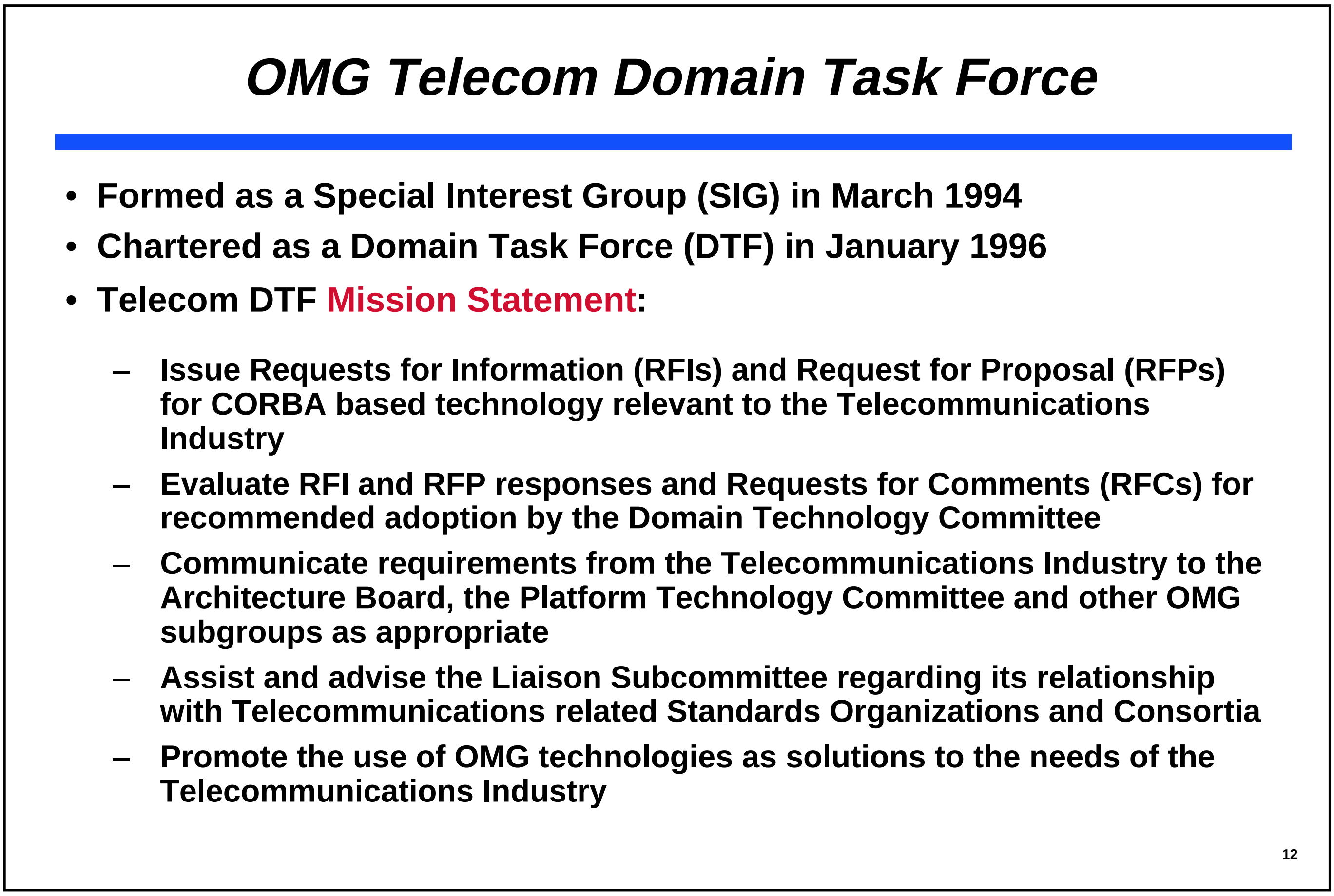
• Electronic Commerce Domain Task Force

• Telecommunications Domain Task Force

• Financial Domain Task Force

• CORBAmed Domain Task Force

In addition OMG defines socalled Special Interest Groups (SIGs). These have a significant role in the OMG architecture. Chartered by the one of the three OMG Technology Committees, the SIGs consist of a group of active OMG members who share a common interest in the application of object technology to a specific vertical market or technology area. SIG membership is open to all OMG members.

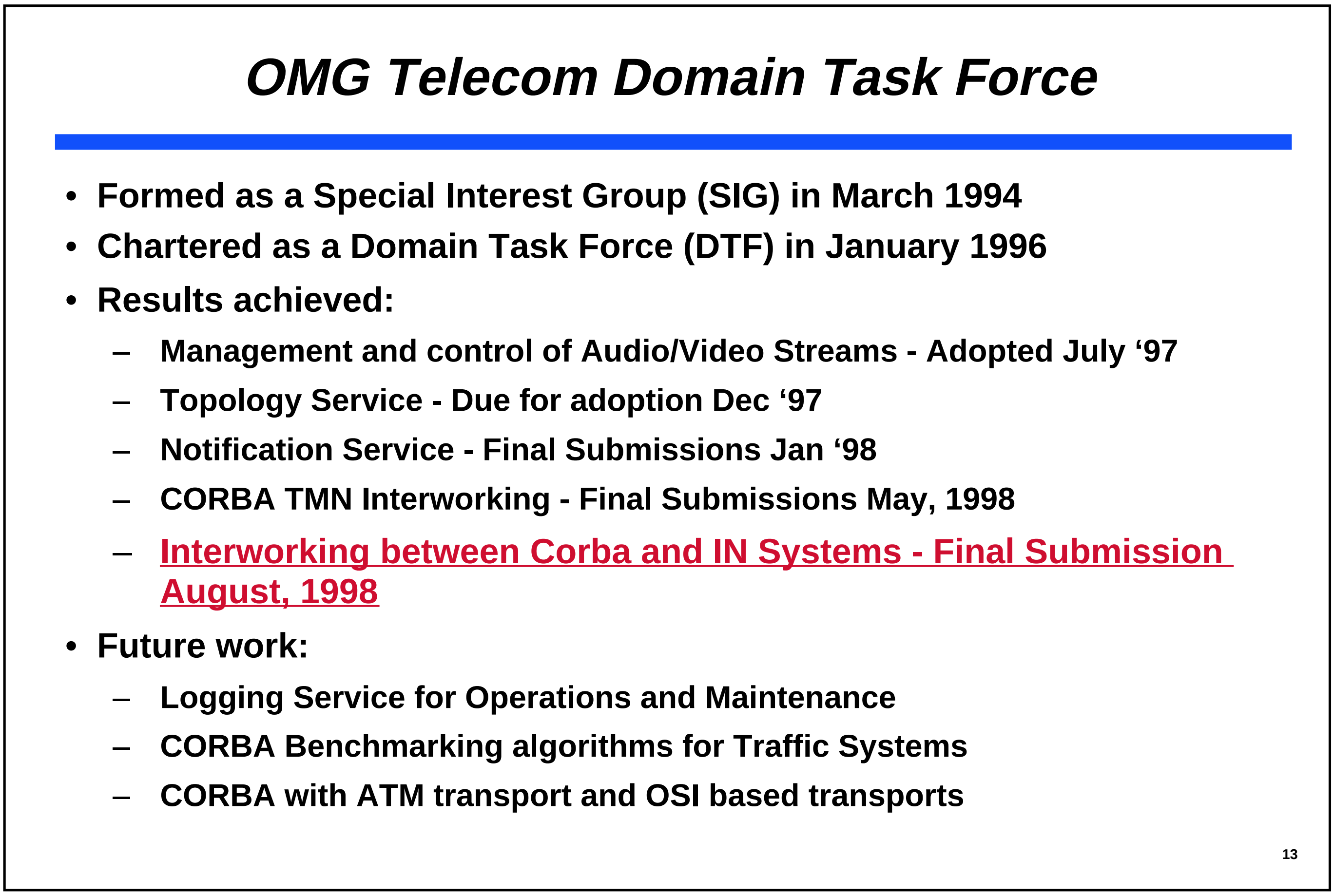


OMG Telecom Domain Task Force

- **Formed as a Special Interest Group (SIG) in March 1994**
- **Chartered as a Domain Task Force (DTF) in January 1996**
- **Telecom DTF Mission Statement:**
  - **Issue Requests for Information (RFIs) and Request for Proposal (RFPs) for CORBA based technology relevant to the Telecommunications Industry**
  - **Evaluate RFI and RFP responses and Requests for Comments (RFCs) for recommended adoption by the Domain Technology Committee**
  - **Communicate requirements from the Telecommunications Industry to the Architecture Board, the Platform Technology Committee and other OMG subgroups as appropriate**
  - **Assist and advise the Liaison Subcommittee regarding its relationship with Telecommunications related Standards Organizations and Consortia**
  - **Promote the use of OMG technologies as solutions to the needs of the Telecommunications Industry**

A dedicated task force is focusing on telecommunications, the Telecommunications Domain Task Force (TDTF). It is in existence for one and a half years and is arguably the most active and successful domain task force of the OMG.

The OMGs Telecoms Task Force is a forum for telecommunications operators and vendors to actively participate in the standardisation effort. Operators and vendors should raise issues which are important for CORBA based technology to make a real difference in the engineering of telecommunications solutions and in facilitating inter-operability between existing systems.



OMG Telecom Domain Task Force

- **Formed as a Special Interest Group (SIG) in March 1994**
- **Chartered as a Domain Task Force (DTF) in January 1996**
- **Results achieved:**
  - **Management and control of Audio/Video Streams - Adopted July ‘97**
  - **Topology Service - Due for adoption Dec ‘97**
  - **Notification Service - Final Submissions Jan ‘98**
  - **CORBA TMN Interworking - Final Submissions May, 1998**
  - **Interworking between Corba and IN Systems - Final Submission August, 1998**
- **Future work:**
  - **Logging Service for Operations and Maintenance**
  - **CORBA Benchmarking algorithms for Traffic Systems**
  - **CORBA with ATM transport and OSI based transports**

A dedicated OMG task force is focusing on telecommunications, the Telecommunications Domain Task Force (TDTF). It is in existence for one and a half years and is arguably the most active and successful domain task force of the OMG.

The OMGs Telecoms Task Force is a forum for telecommunications operators and vendors to actively participate in the standardisation effort. Operators and vendors should raise issues which are important for CORBA based technology to make a real difference in the engineering of telecommunications solutions and in facilitating inter-operability between existing systems.

One standard, "the control and management of audio visual streams", has been already finalised. Two more are close to completion, i.e. "Notification Service" and "Topology Service". An RFP on "CORBA/TMN Interworking" has been issued in September 1997 and another RFP on "CORBA/IN Interworking" has been issued in December 1997.

More information on these RFPs in progress can be obtained from „http://www.omg.org/library/schedule/“.

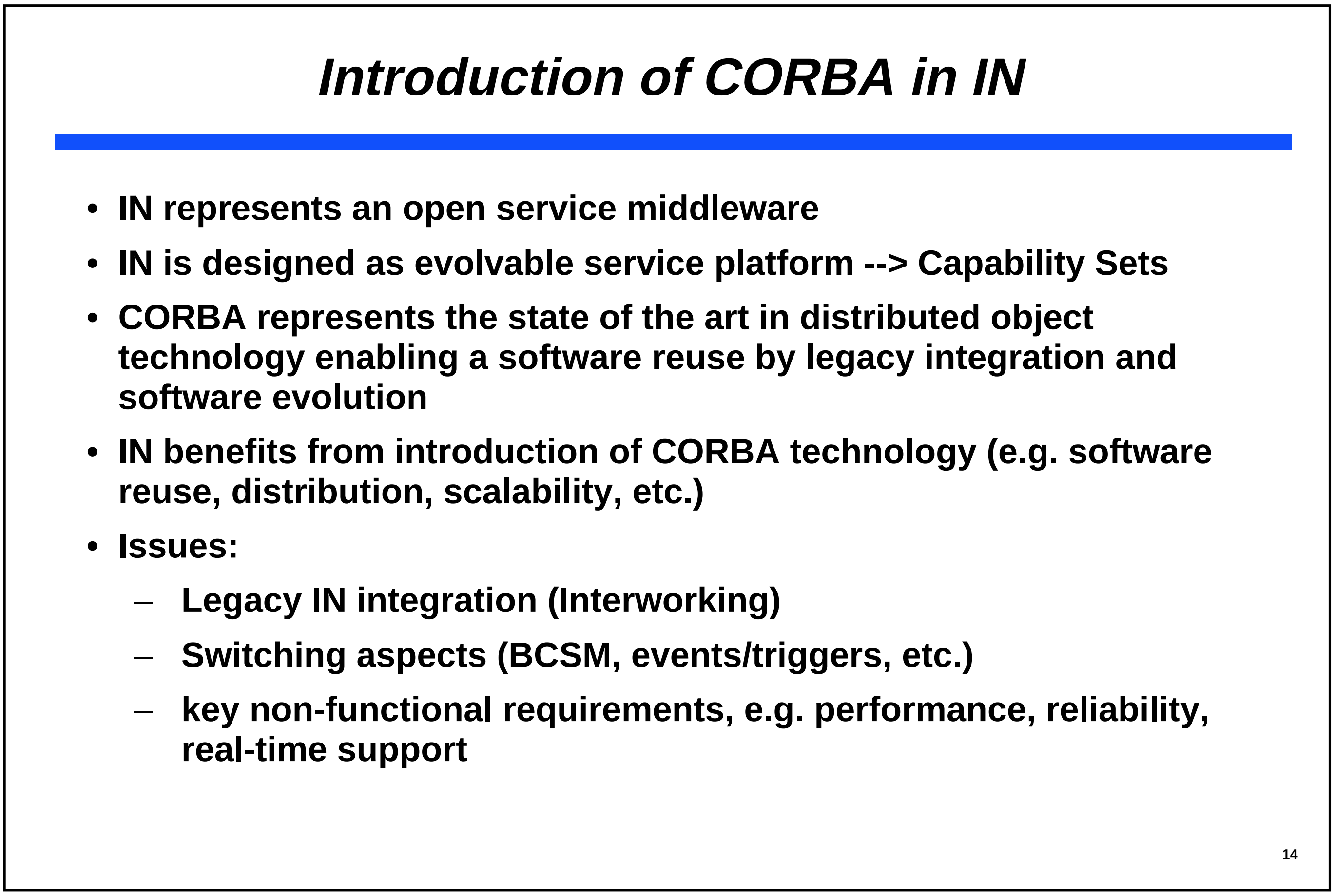
In the future it is the task forces’ members stated intention to address issues such as:

• a logging service for Operations and Maintenance systems,

• standardising benchmarks relevant for attaining concrete performance measurements for CORBA implementations in telecommunications contexts,

• standardising an approach to using ATM as a transport for CORBA systems, and

• standardising an approach to using OSI stack configurations and a transport for CORBA systems.



Introduction of CORBA in IN

- **IN represents an open service middleware**
- **IN is designed as evolvable service platform --> Capability Sets**
- **CORBA represents the state of the art in distributed object technology enabling a software reuse by legacy integration and software evolution**
- **IN benefits from introduction of CORBA technology (e.g. software reuse, distribution, scalability, etc.)**
- **Issues:**
  - **Legacy IN integration (Interworking)**
  - **Switching aspects (BCSM, events/triggers, etc.)**
  - **key non-functional requirements, e.g. performance, reliability, real-time support**

In the last decade the Intelligent Network (IN) has coined the face of telecommunication service provision. The IN aims for service and network independence and thus represents a flexible service platform. Evolution of the IN platform should be possible by the definition of IN Capability Sets, which enable the incorporation of new service needs and new technologies.

Today object-oriented middleware, such as CORBA, is gaining increasing acceeptance, since it promises on the one hand the necessary openness required in the telecommunications market, as well as the possibility to interwork with/integrate legacy technologies.

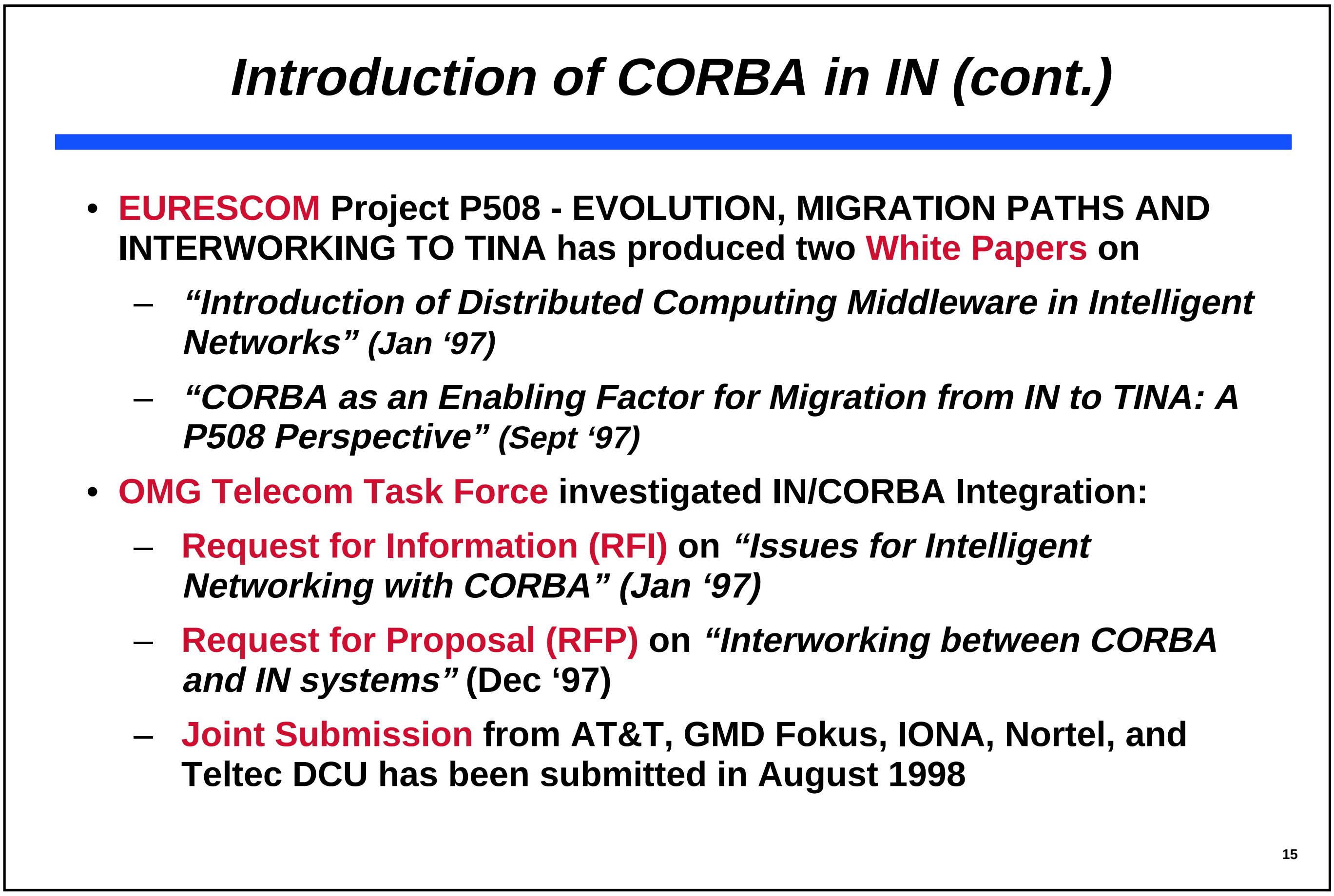
In the following we look at the evolution of IN in face of emerging CORBA-based middleware. The idea is to adopt the OMG CORBA standard, enhancing it to make it suitable for telecom system, in particular for the Intelligent Network (IN).

Distributed object-oriented computing has to be introduced first in the network intelligence. In order for this to be possible, there are three main factors to be taken into account:

• the IN legacy systems, including the Signalling System No. 7 (SS7) network;

• the IN event/trigger computing paradigm, based on the Basic Call State Model (BCSM);

• and the key non-functional requirements, as high performance and reliability, real-time support.



Introduction of CORBA in IN (cont.)

- **EURESCOM Project P508 - EVOLUTION, MIGRATION PATHS AND INTERWORKING TO TINA has produced two White Papers on**
  - ***“Introduction of Distributed Computing Middleware in Intelligent Networks” (Jan ‘97)***
  - ***“CORBA as an Enabling Factor for Migration from IN to TINA: A P508 Perspective” (Sept ‘97)***
- **OMG Telecom Task Force investigated IN/CORBA Integration:**
  - **Request for Information (RFI) on *“Issues for Intelligent Networking with CORBA” (Jan ‘97)***
  - **Request for Proposal (RFP) on *“Interworking between CORBA and IN systems”* (Dec ‘97)**
  - **Joint Submission from AT&T, GMD Fokus, IONA, Nortel, and Teltec DCU has been submitted in August 1998**

15

In December 1996 the Object Management Group (OMG) Telecommunications Domain Task Force (TDTF) issued a White Paper on "Intelligent Networking with CORBA" (OMG DTC Document: telecom/96-12-02.pdf). In January 1997 the OMG TDTF has issued an RFI on „Issues for Intelligent Networking with CORBA“ (OMG DTC Document: telecom/96-12-02.pdf) in order to setup a corresponding RFP. The answers provided the basis for the development of a corresponding Request for Response (RFP) on “Interworking between CORBA and IN Systems” (OMG DTC Document: telecom/97-12-06.pdf) which has been issued in December 1997.

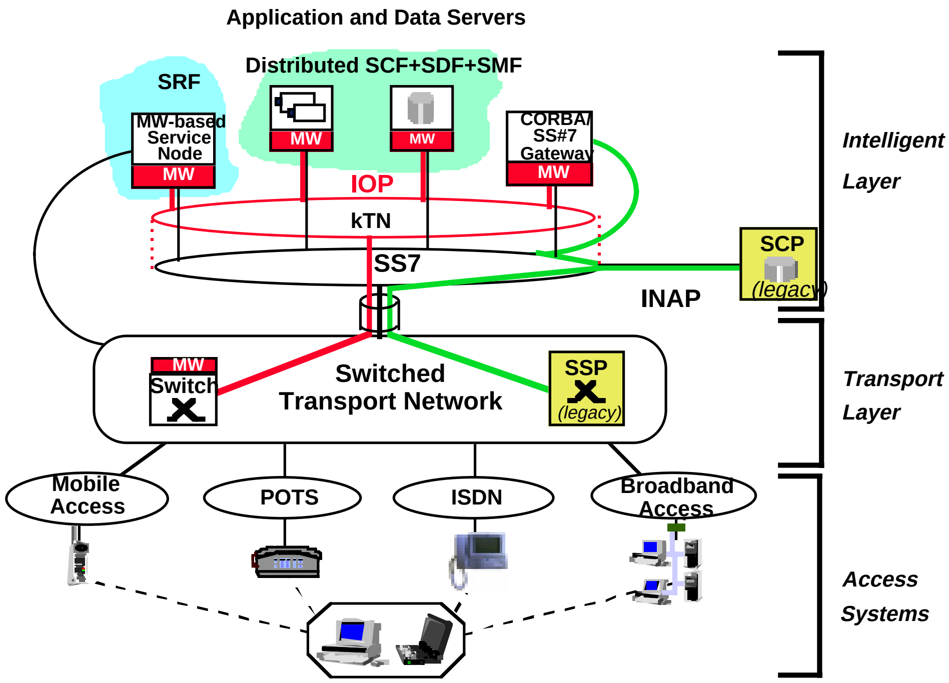
Two initial RFP responses have been produced by IONA/Nortel and AT&T/GMD Fokus/Teltec DCU in May 1998 (OMG DTC Documents: telecom/98-05-06.pdf and telecom/98-05-03.pdf). However, a revised joint submission by AT&T, GMD Fokus, IONA, Nortel, Teltec DCU has been submitted in August 1998 (OMG DTC Document: telecom/98-08-11.pdf) represents the latest state of the art. All these documents are available on the public OMG FTP server under the following URL: "ftp://ftp.omg.org/pub/docs/telecom/".

More information concerning the RFP responses can be found under “http://www.omg.org/library/schedule/ CORBA_Intelligent_Networks_RFP.htm”.

In addition, the EURESCOM Project 508 has contributed by the production of two White Papers; "White Paper on CORBA as an Enabling Factor for Migration from IN to TINA: A P508 Perspective" (OMG DTC Document: telecom/97-01-01.pdf) in January 1997 and "Introduction of Distributed Computing Middleware in Intelligent Networks" (OMG DTC Document: telecom/97-09-01.pdf) in September 1997.

These two documents are also available from the OMG FTP server.

IN and CORBA - Overall Vision



INAP	Intelligent Network Application Protocol
IOP	Interoperability Protocol
IP	Intelligent Peripheral
ISDN	Integrated Service Digital Network
kTN	Kernel Transport Network
MW	Middleware
POTS	Plain Old Telephone System
SCF	Service Control Function
SCP	Service Control Point
SDF	Service Data Function
SMF	Service Management Function
SN	Service Node
SRF	Special Resource Function
SS7	Signalling System N. 7
SSP	Service Switching Point

16

The introduction of a middleware software layer in application and data servers, and eventually also in switching systems, enables to have a component-based, distributed intelligence replacing the “traditional” monolithic Intelligent Network functional entities.

The switched transport network ensures transfer of user information across connections, on which calls are established. It is accessed via a variety of access systems, each roughly corresponding to a class of customer premises equipment:

- mobile systems connecting cellular phones;
- Plain Old Telephone System (POTS);
- Integrated Service Digital Network (ISDN) connecting digital phones or similar devices;
- broadband networks, typically connecting Local Area Networks (LANs).

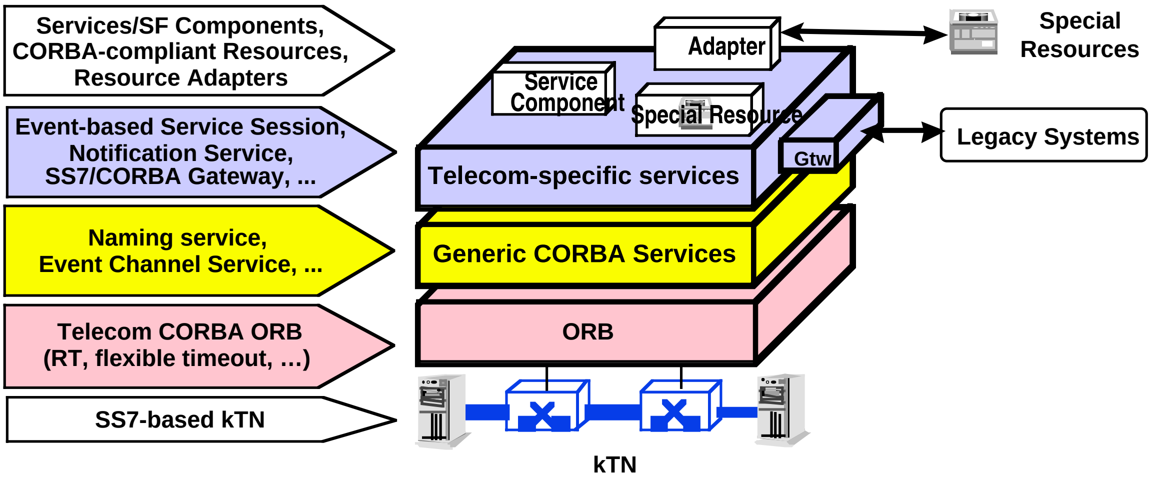
Personal computers can be networked on the public network using any of these access systems.

The basic call service in the switched network is typically offered by circuit-related signalling protocols, such as ITU-T Q.931 for N-ISDN.

The middleware platform enables to realise IN functions in a distributed way, that is by a set of application and data servers interacting via the platform corresponding to IN functional entities. This platform is based on CORBA, but requires enhancements to the state-of-the-art CORBA specifications and products. CORBA servers contain CORBA objects, acting as reusable service components. The approach is similar to that of the Telecommunications Information Networking Architecture (TINA).

Communications at the middleware platform level is based on an Interoperability Protocol (IOP), relying either on CORBA. The application-level signalling network ensuring communication among platform nodes is termed Kernel Transport Network (kTN). It should rely on the existing SS7 signalling network, which fulfillls important requirements for telecom applications (such as high reliability). Interoperability with legacy IN elements and services is ensured by a CORBA/SS7 gateway.

Introduction of CORBA in IN



The architecture of the middleware layer is based on an enhanced CORBA Object Request Broker (ORB), targeted to telecom applications: it has to include real-time features and a flexible timeout mechanism. The kTN is a high-capacity and very reliable packet-switched network, that transports signalling messages. It needs to be based on Signalling System No. 7.

On top of the ORB, some generic CORBA services (i.e., not specifically targeted to telecom applications) are needed. At least, the naming service and the generic CORBA event service (termed also Event Channel service) are needed, but other services may be also required.

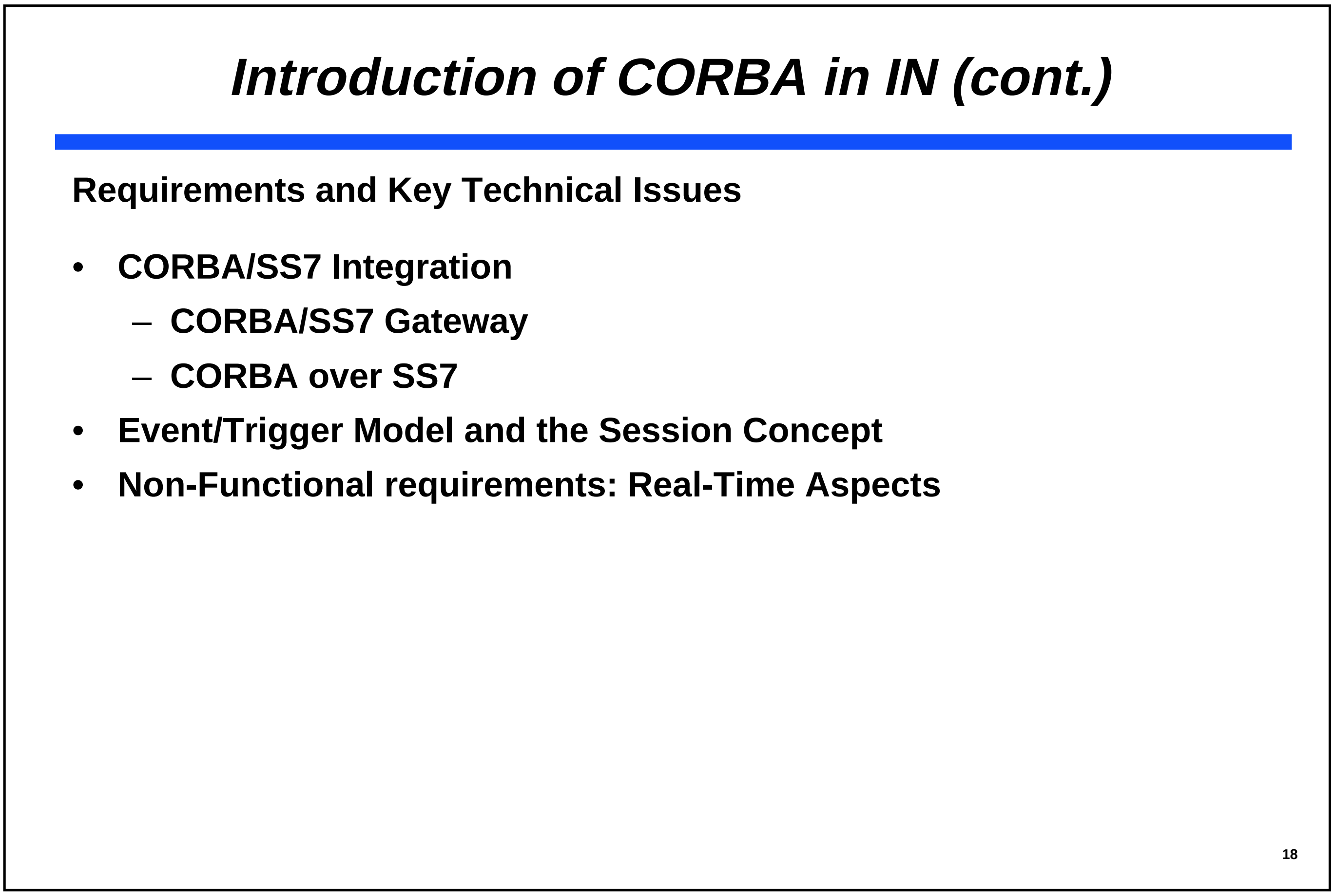
Generic CORBA services, however, are not enough to fulfill telecommunications application requirements. Telecom-specific services are needed, for instance to handle different kind of events and to establish a context (service session) for the provision of telecom services. For interoperability with legacy systems, a gateway between CORBA and the SS7 protocol stack must be foreseen; this gateway would enable interoperation between CORBA servers and legacy IN. The usage of special purpose real-time event services, besides the Event Channel service, could also be investigated; in alternative, event handling could be also embedded in the telecom ORB.

On top of the middleware layer, depicted as white boxes in figure, three types of entities are deployed:

• Service/Service Feature components: the "service logic", implementing a wide range of services and service features by means of reusable components;

• CORBA-compliant special resources: resources (such as bridges, multi-media servers, and so on) that have been designed in such a way that they can be directly "plugged in" on the middleware layer;

• Adapters for special resources (also bridges, multi-media servers and such) that interact with the exterior with a different paradigm than the distributed computing middleware – for example with proprietary protocols.



Introduction of CORBA in IN (cont.)

Requirements and Key Technical Issues

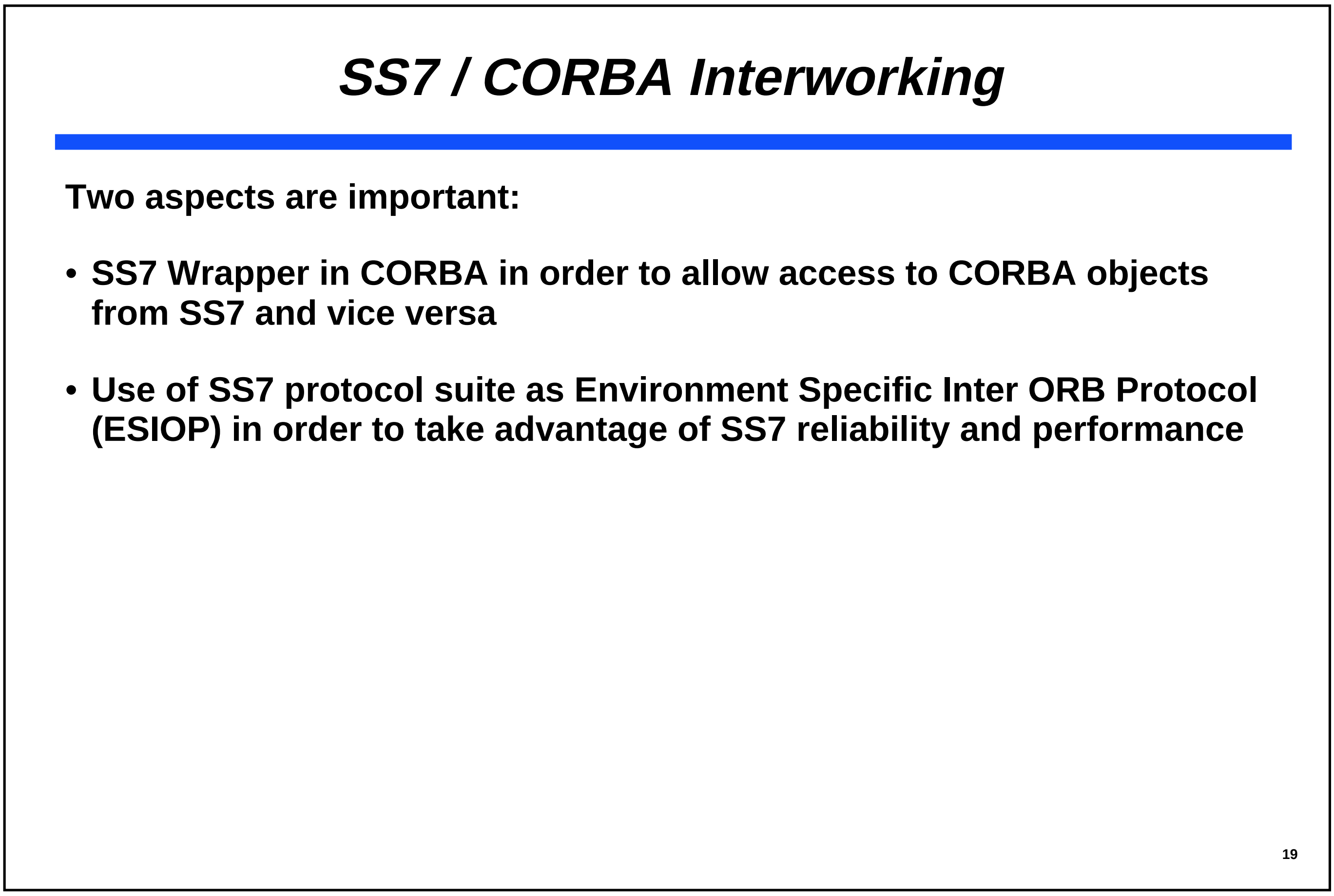
- **CORBA/SS7 Integration**
  - **CORBA/SS7 Gateway**
  - **CORBA over SS7**
- **Event/Trigger Model and the Session Concept**
- **Non-Functional requirements: Real-Time Aspects**

In order for this to be possible, three are three main factors to take into account: the IN legacy systems, including the Signalling System N. 7 (SS7); the IN event/trigger computing paradigm, based on the Basic Call State Model (BCSM); the key non-functional requirements, as high performance and reliability, real-time support.

Taking into account the SS7 leads to the issue of defining which profile of the protocol stack must be used, and of building a bridge (gateway) between the IN and the CORBA world based on SS7; it also leads to the idea of using the SS7 to develop a kernel Transport Network (kTN) to convey ORB messages over a geographical CORBA network.

Taking into account the IN computing paradigm leads to define an implementation of the IN service logic in terms of components interacting using CORBA-based event services.

Taking into account non-functional requirements leads to define requirements on a “telecom-specific” ORB.



SS7 / CORBA Interworking

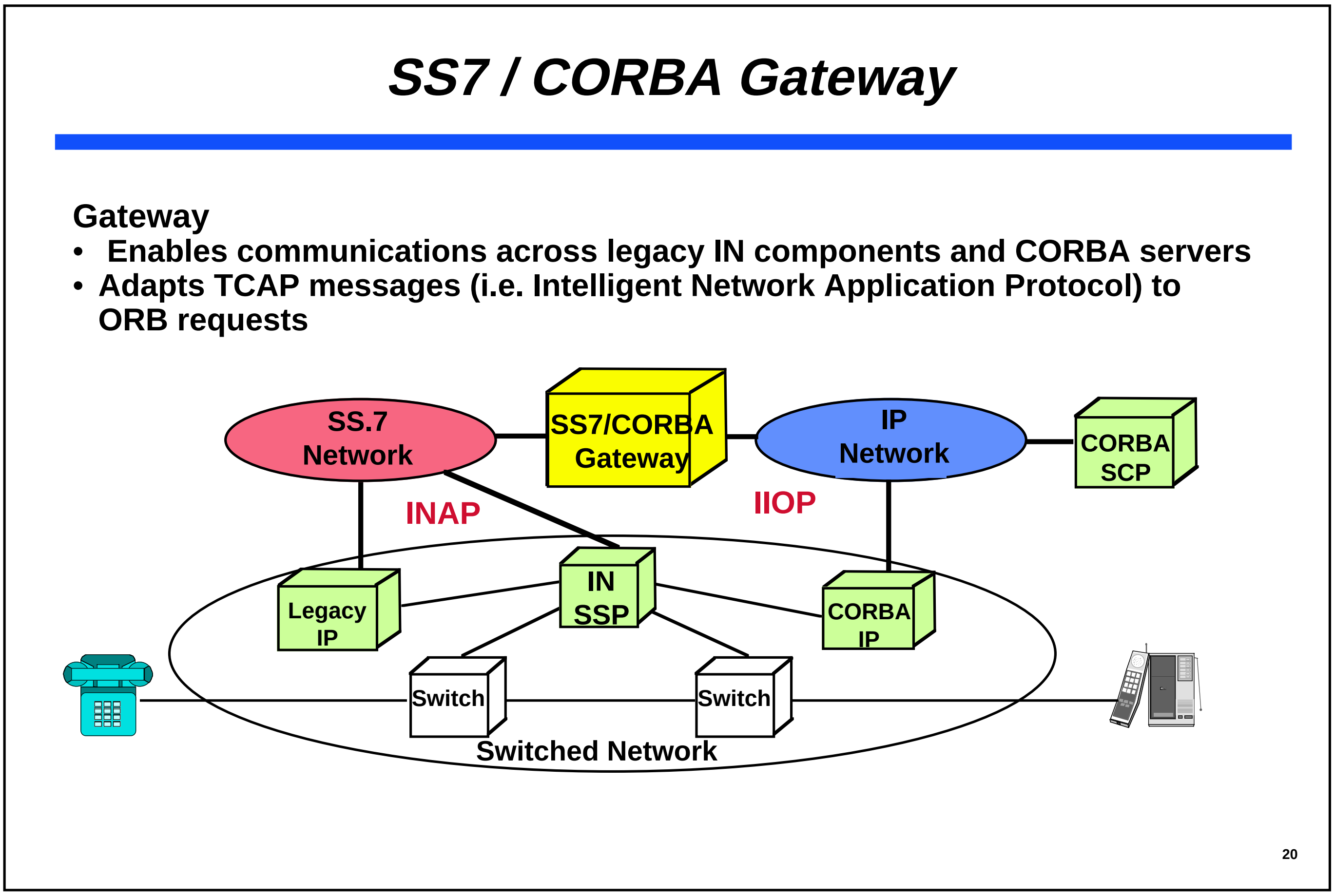
Two aspects are important:

- **SS7 Wrapper in CORBA in order to allow access to CORBA objects from SS7 and vice versa**
- **Use of SS7 protocol suite as Environment Specific Inter ORB Protocol (ESIOP) in order to take advantage of SS7 reliability and performance**

The migration from IN to TINA is based on the introduction of DPE technology in IN platform entities. Since CORBA represents the choice for implementing a TINA DPE, an adaptation / integration of SS7 and CORBA mechanisms is necessary. In this context basically two major issues have to be considered:

1. An SS7 gateway within CORBA has to be provided in order to enable the interworking of SS7 and CORBA/TINA. Such gateway enables the access to CORBA objects from SS7 entities and vice versa.

2. The use of SS7 network as TINA KTN or more generally the usage of the SS7 protocol suite as inter-ORB protocol allows to take advantage of SS7 reliability and performance in case CORBA/TINA extends to the switch.



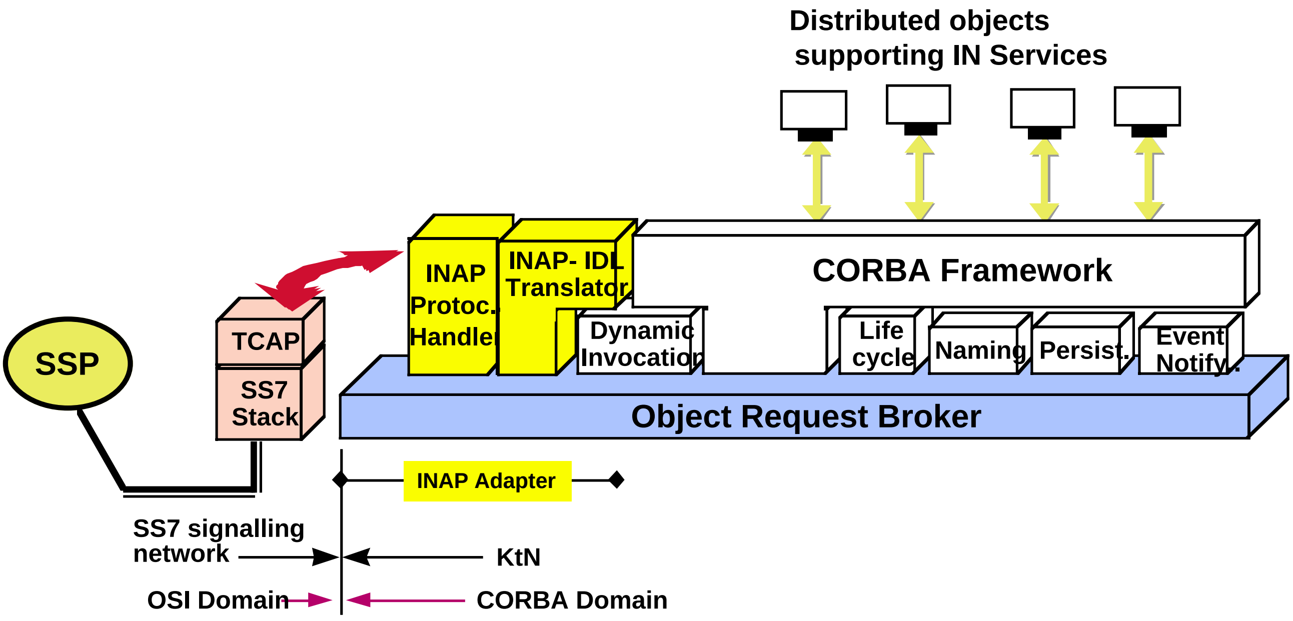
SS7 / CORBA Gateway

Gateway

- **Enables communications across legacy IN components and CORBA servers**
- **Adapts TCAP messages (i.e. Intelligent Network Application Protocol) to ORB requests**

In order to enable TINA/CORBA objects to control a IN SSPs it is essential to define a dedicated application-level gateway object (i.e. an INAP wrapper), that provides a CORBA/IDL interface (API) to the objects on the CORBA side and an SS7 interface towards the signalling network on the other side. This means that an IN SSP communicates with other network entities using SS7/INAP, while on the CORBA/ TINA side invocations of IDL interfaces are used for communication. The gateway, located in between, is in charge of transparently adapting both types of communication.

SS7 / CORBA Gateway (cont.)



Two options exist for the realisation of such a gateway:

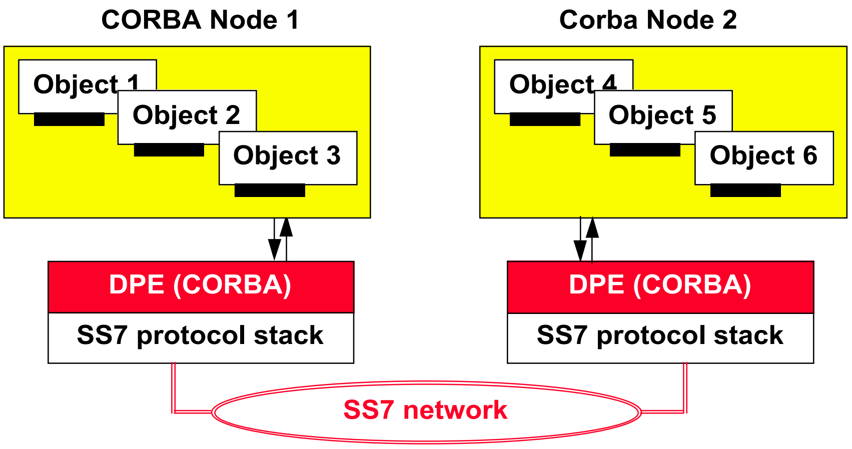
1. Definition of an INAP/CORBA gateway: INAP specifications can first be converted into IDL-based interfaces. This involves converting the IN operations and their returns into CORBA operation signatures. This would be a IN-specific mapping, but it must be noted that there are various “dialects” of the ITU-T standardized INAP.

2. Definition of an TCAP/CORBA gateway: A generic application-level gateway could be defined for all TCAP / ROSE Users (of which INAP is one example) by providing translation algorithms for converting between ROS constructs defined in Abstract Syntax Notation One (ASN.1) and the corresponding CORBA constructs using IDL.

The second option is probably the better one. Internally the gateway could look like the following. INAP operations received from the SSP are forwarded from an SS7 driver (offering a TCAP interface) to an INAP Protocol Handler. The INAP Protocol Handler forwards the INAP operation to an INAP-IDL Translator. Besides the INAP operation also the ID of the TCAP transactions must be forwarded. This is necessary in order to dispatch all INAP operations belonging to one transaction to the same object in the CORBA/TINA environment. The functionality of the INAP-IDL Translator is very simple. It only takes the INAP operation with the included Information Elements and invokes an IDL interface of the Dispatcher, taking the name of the INAP operation and its Information Elements as parameters for the IDL interface invocation. The IDL interface at the Dispatcher is general and used for all kinds of INAP operations. These procedures are used analogously for sending messages into the opposite direction.

SS7 network as kTN

- **Issue: Provision of connectivity between CORBA nodes in telecom networks**
- **strong requirements in telecom world (real-time, reliability etc.)**
- **protection of operators investment**
- ***Reuse of SS7 for Inter-ORB communications***



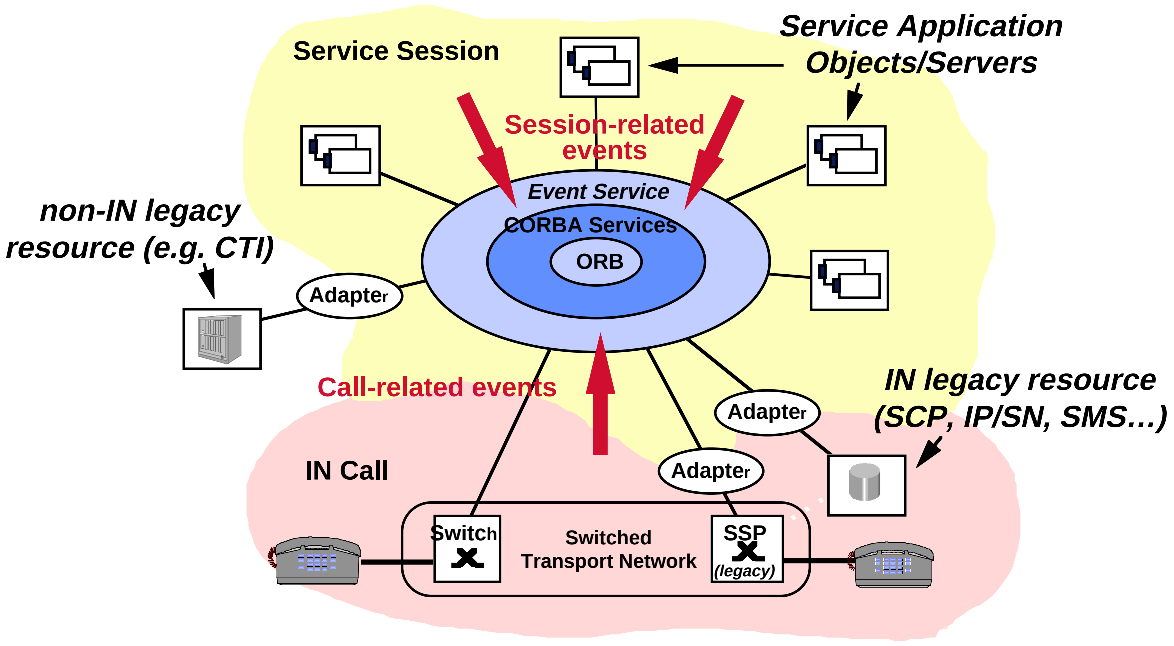
We assume that all IN platform entities are modelled by means of distributed CORBA/TINA objects located at DPE nodes. The SS7 signalling network acts as a KTN providing the communication infrastructure for a DPE. The main benefit to be obtained from this strategy is that any investments incurred by network operators during the deployment of their existing SS7 networks are preserved. On the other hand the inter-ORB communication will profit from the reliability and performance of the SS7 network!

A possibility is to build an implementation of Generic Inter-ORB Protocol (GIOP) on top of SS7 or if necessary build a new Environment-specific Inter-ORB Protocol (ESIOP) on top of SS7 layers. It is structurally similar to the engineering commonly used to bridge nodes within a single ORB. It means in particular adding a new inter-process communication schema on top of new transport protocol stack. This strategy needs of course the participation of ORB vendors.

At present, OMG has defined a general purpose inter-ORB protocol called GIOP, that can be used over transport services.

Using GIOP as the basis for ORB interoperability in SS7 networks is not as easy as for TCP/IP or OSI networks due to the following reasons. SS7 does not currently specify a transport service for non-circuit related applications such as IN services. This type of applications are defined over TCAP which is conceptually supported by an Intermediate Service Part (ISP) covering layers 4 through 6 of the ISO OSI reference model. However, existing SS7 standards do not specify ISP and thus TCAP is directly positioned above the SS7 network service (i.e. SCCP). Although SCCP specifies connection-oriented protocol classes, most SS7 networks are connectionless. Furthermore, the connectionless services of SCCP limit the size of network service data units to 2K octets.

Event-Based Service Session


Service Session
Service Application Objects/Servers
Session-related events
Event Service
CORBA Services
ORB
non-IN legacy resource (e.g. CTI)
Adapter
Call-related events
IN legacy resource (SCP, IP/SN, SMS…)
Adapter
Adapter
IN Call
Switch
Switched Transport Network
SSP
(legacy)

The introduction of a distributed processing platform in the intelligent layer enables the service logic to be structured in service components, corresponding to objects on a CORBA platform. This component-based approach enables dynamic service composition, i.e. the flexible assembling of pre-existing components. The IN is based on the event/trigger paradigm, by which the service logic is triggered by events expressed as detection points in the BCSM. This paradigm is well suited to support dynamic service composition. Therefore it is necessary to define a context for the execution of advanced services composed dynamically. The concept of service session, taken from TINA, can be used to this purpose.

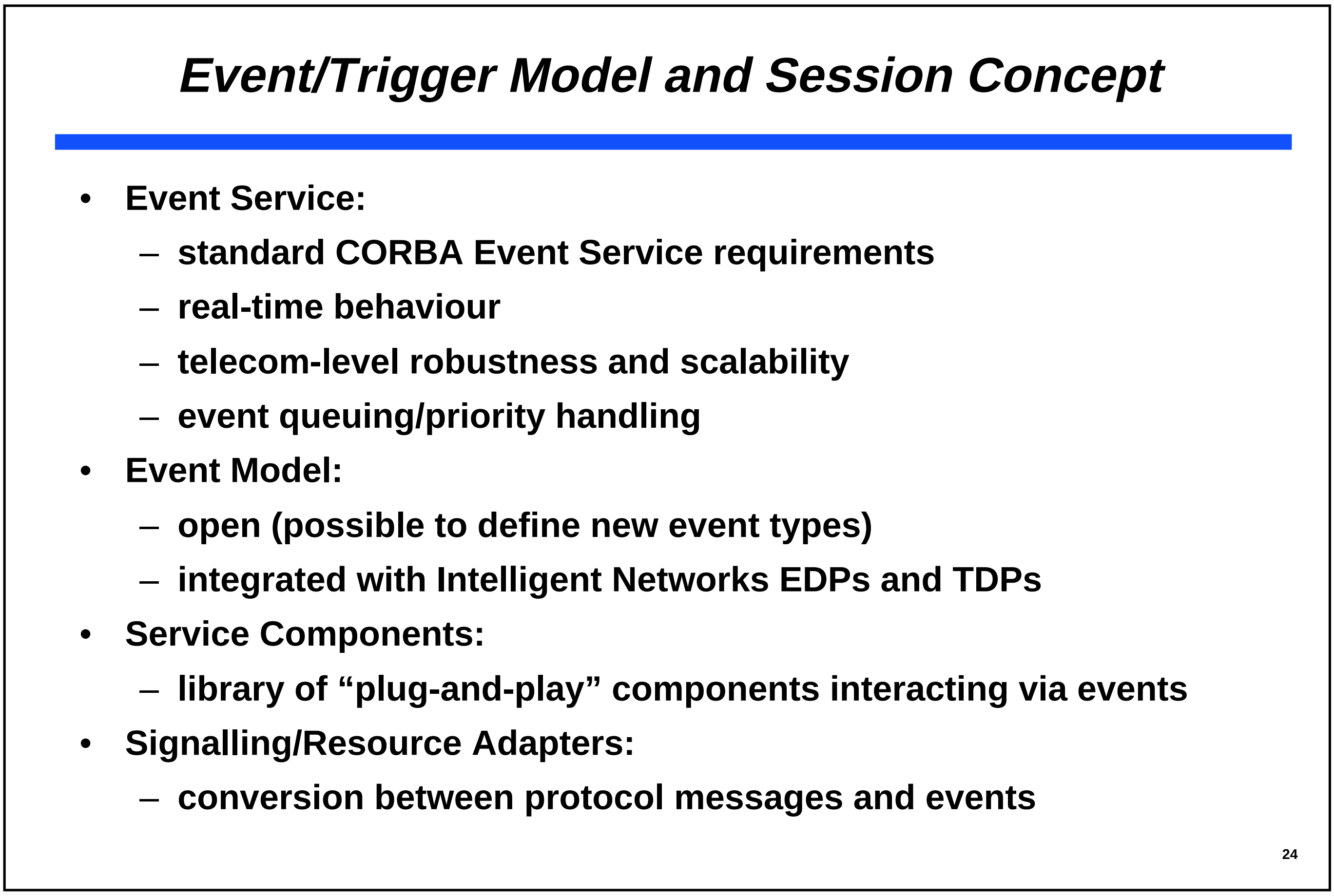
Consequently, an IN call, on which value-added services are provided, corresponds to a service session. Within the session, services are provided by means of components running as objects in CORBA servers, interacting among themselves with an event-based paradigm.

Events can be classified in two categories:

• Call-related events, originated by the SSF (i.e., in the SSP); these events are related to the IN Basic Call State Model (BCSM), which is expressed by a state machine governing the interaction between call control and service control; the IN trigger detection points (TDPs) and event detection points (EDPs) in the BCSM can be easily mapped to this concept.

• Session-related events, originated by service components in the service session; these events can be seen as "value-added" events produced by service components: components process call related events, and produce these more specific events, which in turn can be "consumed" by other components in a flexible event / trigger activation mechanism dynamically binding all components.

This approach, based on the service session concept, enables the integration of a wide range of systems with the IN infrastructure. In fact, adapters can be implemented that convert signalling protocol messages (such as INAP) into events, as shown in figure. This facilitates the adoption of this approach already on present-day IN systems. Non-IN devices, such as CTI resources, can also be plugged in the event platform and thus be made available to the service session. In this way the same service logic can be used to provide a given service on multiple network infrastructures (such as, for example, IN and CTI).



Event/Trigger Model and Session Concept

- **Event Service:**
  - **standard CORBA Event Service requirements**
  - **real-time behaviour**
  - **telecom-level robustness and scalability**
  - **event queuing/priority handling**
- **Event Model:**
  - **open (possible to define new event types)**
  - **integrated with Intelligent Networks EDPs and TDPs**
- **Service Components:**
  - **library of “plug-and-play” components interacting via events**
- **Signalling/Resource Adapters:**
  - **conversion between protocol messages and events**

24

Another important idea needs to be introduced to describe modular services: the service feature concept, defined by IN standards. A service feature is a constituent part of a service as perceived by the service user: PIN authentication, split charging, are service features. In a component-based approach, a service feature does not necessarily correspond to one component, but rather to a set of components (ultimately mapping to CORBA objects) distributed over multiple nodes, possibly spanning multiple administrative domains.

Service features can be viewed as entities emitting and receiving events; these entities are deployed in a plug-and-play fashion on an event bus (the event service on CORBA). These service features are implemented by a set of CORBA objects.

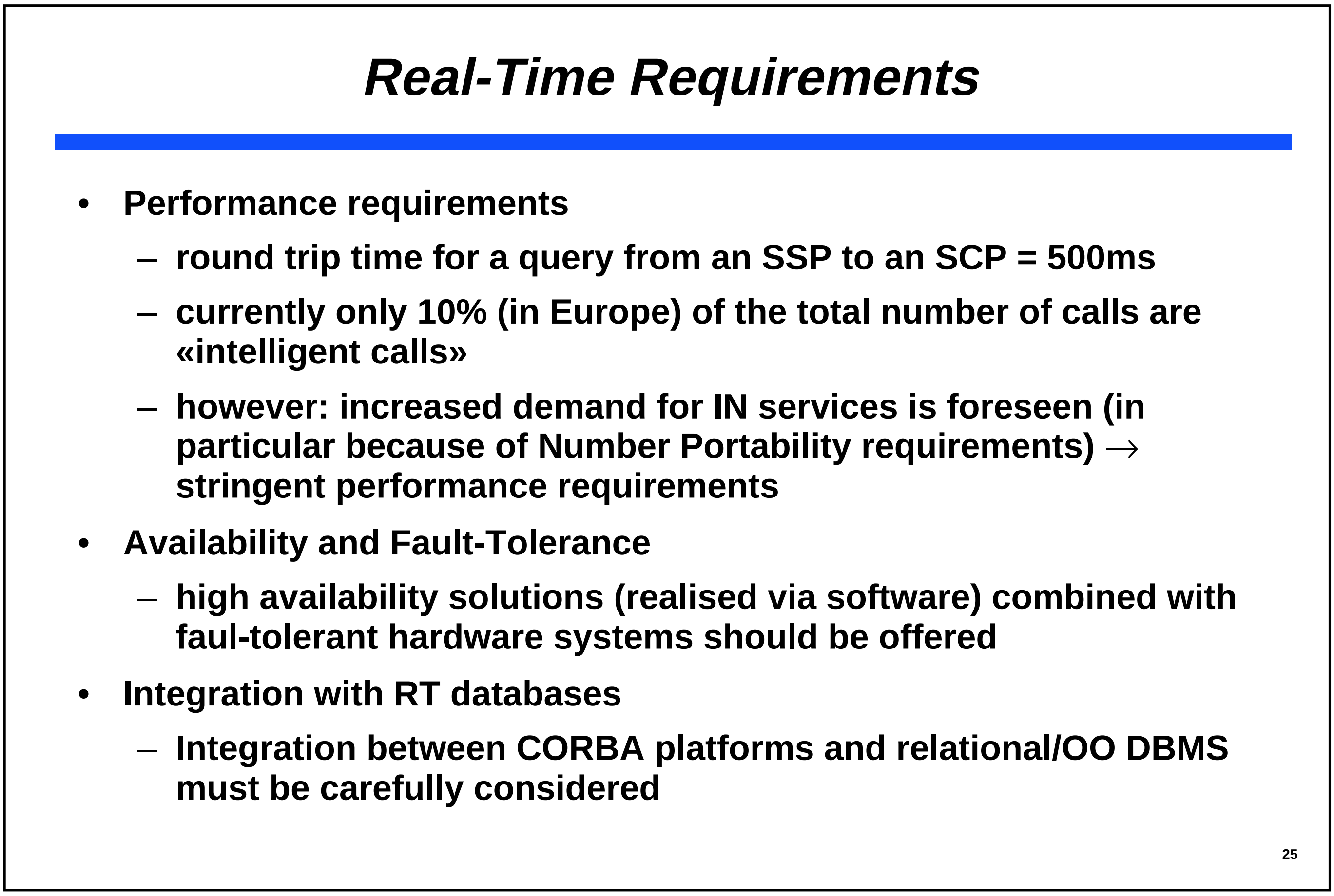
In order to achieve this, the following needs to be specified and implemented:

1. An event service targeted to the IN session; this event service has to be superposed to CORBA to constitute an upper middleware layer, as shown in figure.

2. An event model, i.e. an open set of pre-defined events; the set of event types constituting the event model is the shared knowledge among all plug-and-play components. Component designers have the knowledge of all possible events, and choose which defined events a given component reacts to; they can also define new events specific to services and/or network resources. These events need to be registered to the event service.

3. Service/Service Feature components; these are the software packages that implement service features; they are defined by service designers in accordance with the CORBA platform, the service session concept, the event service and the event model. Hence, they are “plug-and-play” in the IN context.

4. Signalling/resource adapters; they implement adaptation functions from different protocols to the service session.



Real-Time Requirements

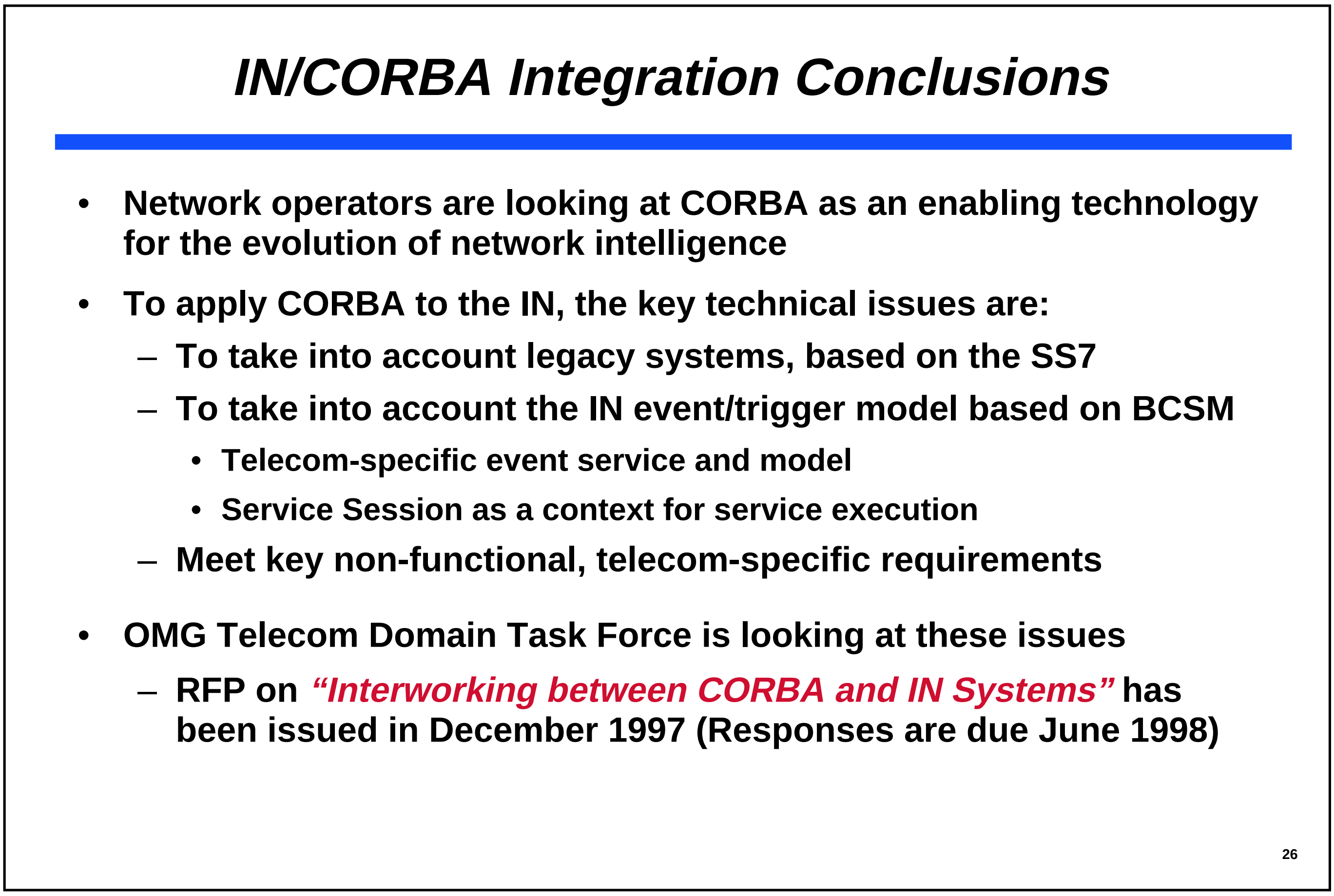
- **Performance requirements**
  - **round trip time for a query from an SSP to an SCP = 500ms**
  - **currently only 10% (in Europe) of the total number of calls are «intelligent calls»**
  - **however: increased demand for IN services is foreseen (in particular because of Number Portability requirements) → stringent performance requirements**
- **Availability and Fault-Tolerance**
  - **high availability solutions (realised via software) combined with faul-tolerant hardware systems should be offered**
- **Integration with RT databases**
  - **Integration between CORBA platforms and relational/OO DBMS must be carefully considered**

All the scenarios presented are based on the assumption that the platform is able to support real-time performance. This is a fundamental requirement in order to allow the use of distributed processing within IN environments. The round trip time of a query from an SSP to the Intelligent Layer has to be within 500 ms.

In many (European) countries the usage of IN services is below 10% of the total number of calls. In this respect the current configuration of SCP/SMS systems is appropriate for supporting this traffic. However, in the future it is foreseen a strong increase in the demand and usage of IN-intensive services. Many telecom actors consider, in the long run, the IN infrastructure as a means to cope with Number Portability. There could be the need to change the classical centralised IN architecture in order to introduce distributed processing mechanisms on a geographical scale. In order to achieve good performance figures, the processing time needed by the CORBA platform has to be negligible with respect to the total processing time required by an IN call. The foreseen increased demand for IN services and the fast response time requirement pose very stringent requirements on a CORBA based DPE.

The ideal solution for distributed IN systems is the use of general-purpose computing nodes. This could foster a reduction in prices of IN systems. However, it is requested that High Availability or Fault Tolerance capabilities are provided by distributed IN systems in order to ensure they have the same robustness as current IN systems. High availability solutions (realised via software) should be offered by CORBA platform. In addition, CORBA platforms should be developed and be compatible with fault-tolerant systems (hardware solutions). CORBA platform should be profiled in order to be applicable to different hardware and software configurations.

Another important issue is related to the capability to decouple data management from the service logic execution. Data are to be accessed in real-time for service execution purposes, and, at the same time, data should be stored for management purposes. Integration between CORBA platforms and relational and OO databases must be carefully considered in order to guarantee real-time access to data and data integrity.



IN/CORBA Integration Conclusions

- **Network operators are looking at CORBA as an enabling technology for the evolution of network intelligence**
- **To apply CORBA to the IN, the key technical issues are:**
  - **To take into account legacy systems, based on the SS7**
  - **To take into account the IN event/trigger model based on BCSM**
    - **Telecom-specific event service and model**
    - **Service Session as a context for service execution**
  - **Meet key non-functional, telecom-specific requirements**
- **OMG Telecom Domain Task Force is looking at these issues**
  - **RFP on *"Interworking between CORBA and IN Systems"* has been issued in December 1997 (Responses are due June 1998)**

Useful References for further studies

• T.Mowbray, R.Zahavi: The essential CORBA: System Integration using Distributed Objects; John Wiley; 1995

• R.Orfali, D.Harkey: The Essential Distributed Objects Survival Guide; John Wiley; 1995

• H. Zuidweg et.al.: "A Distributed CORBA-Based IN Architecture", pp. 260-265, Int. Conference on IN (ICIN), Bordeaux, France, November 1996

• S. Mazumdar, N. Mitra: „ROS-to-CORBA Mappings: A First Step towards Intelligent Networking using CORBA", in: "Technology for Cooperative Competition - IS&N'97", Springer Publishers, May 1997

• OMG White Paper on "Intelligent Networking with CORBA", OMG DTC Document: "telecom/96-12-04.xxx", December 1996

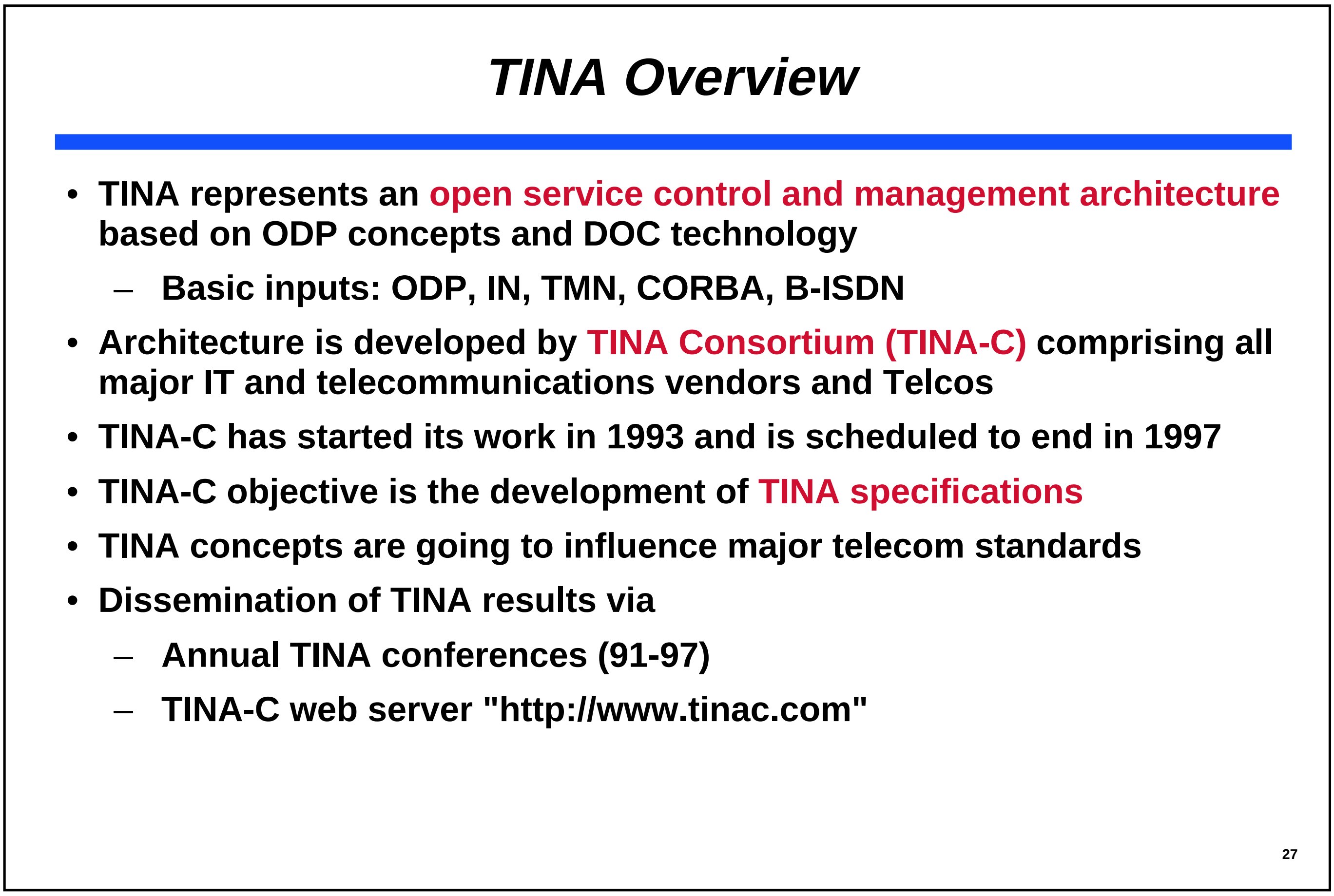
• OMG "Request for Information (RFI) on issues concerning intelligent networking with CORBA", OMG DTC Document:" telecom/96-12-02.xxx , November 1996

• OMG Request For Proposal (RFP) on "Interworking between CORBA and Intelligent Networks Systems", OMG DTC Document: telecom/97-12-06.xxx, December 1997

• EURESCOM Project P508 White Paper: "CORBA as an Enabling Factor for Migration from IN to TINA: A P508 Perspective", OMG DTC Document: "telecom/97-01-01.xxx", January 1997

• EURESCOM Project P508 White Paper: "Introduction of Distributed Computing Middleware in Intelligent Networks", OMG DTC Document: "telecom/97-09-01.xxx", October 1997

All OMG and EURESCOM documents are available on the public OMG FTP server under the following URL: "ftp://ftp.omg.org/pub/docs/telecom/".



TINA Overview

- **TINA represents an open service control and management architecture based on ODP concepts and DOC technology**
  - **Basic inputs: ODP, IN, TMN, CORBA, B-ISDN**
- **Architecture is developed by TINA Consortium (TINA-C) comprising all major IT and telecommunications vendors and Telcos**
- **TINA-C has started its work in 1993 and is scheduled to end in 1997**
- **TINA-C objective is the development of TINA specifications**
- **TINA concepts are going to influence major telecom standards**
- **Dissemination of TINA results via**
  - **Annual TINA conferences (91-97)**
  - **TINA-C web server "http://www.tinac.com"**

27

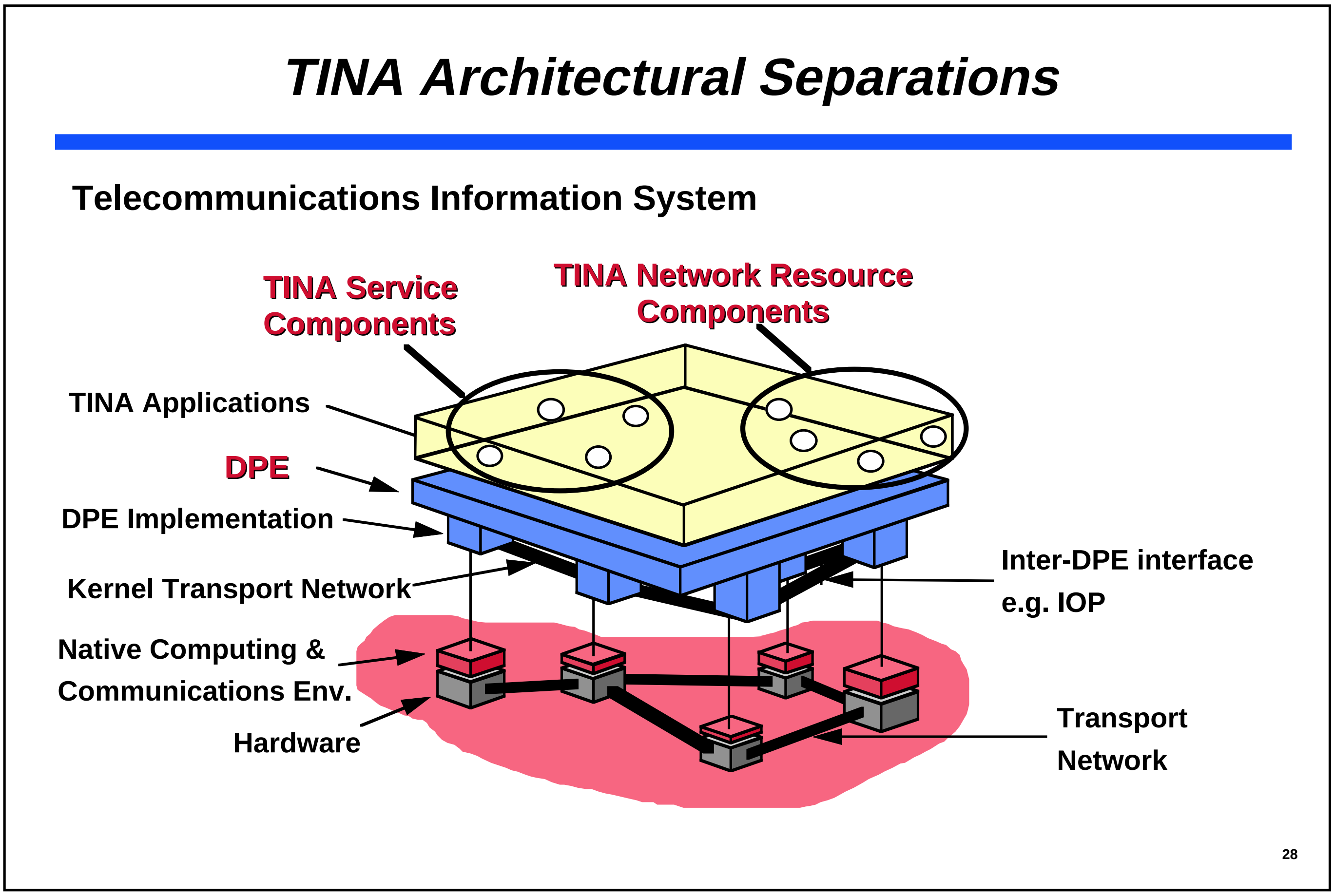
TINA is an open software architecture for distributed telecommunications, information and management services on top of any type of network (such as PSTN, ISDN, B-ISDN, etc.), although a broadband environment represents the primary target. The intent is that all software applications (telecommunications, operations and management) be easier to develop and maintain within an environment, where global aspects increasingly have to be taken into account. This architecture is being defined by the TINA consortium (TINA-C), which was founded in 1992 and comprises all major network operators, telecom manufacturers and computer manufacturers worldwide.

The focus within work of TINA-C is the development and validation of architecture specifications, which should be based as far as possible on an integration of available concepts, standards and products in the fields of telecommunications and information technology. These specifications will be developed by the TINA-C "core team", a permanent set of about 40 researchers working together in a single location.

Although TINA started as a closed research effort in the beginning, i.e. TINA draft specifications have been marked confidential to its members, TINA has been opened in 1996.

Besides annual TINA conferences detailed information on TINA-C and the specifications can be obtained from the TINA web server.

The TINA architecture comprises basically a TINA-C Distributed Processing Environment (DPE) which enables TINA applications to be transparently distributed on top of a heterogeneous hardware/software environment; a set of generic applications and service components to control and manage resources, providing the platform for flexible service realization; and a set of rules for the description, specification, deployment, usage and management of services.



TINA Architectural Separations

Telecommunications Information System

TINA Service Components

TINA Network Resource Components

TINA Applications

DPE

DPE Implementation

Kernel Transport Network

Native Computing & Communications Env.

Hardware

Inter-DPE interface e.g. IOP

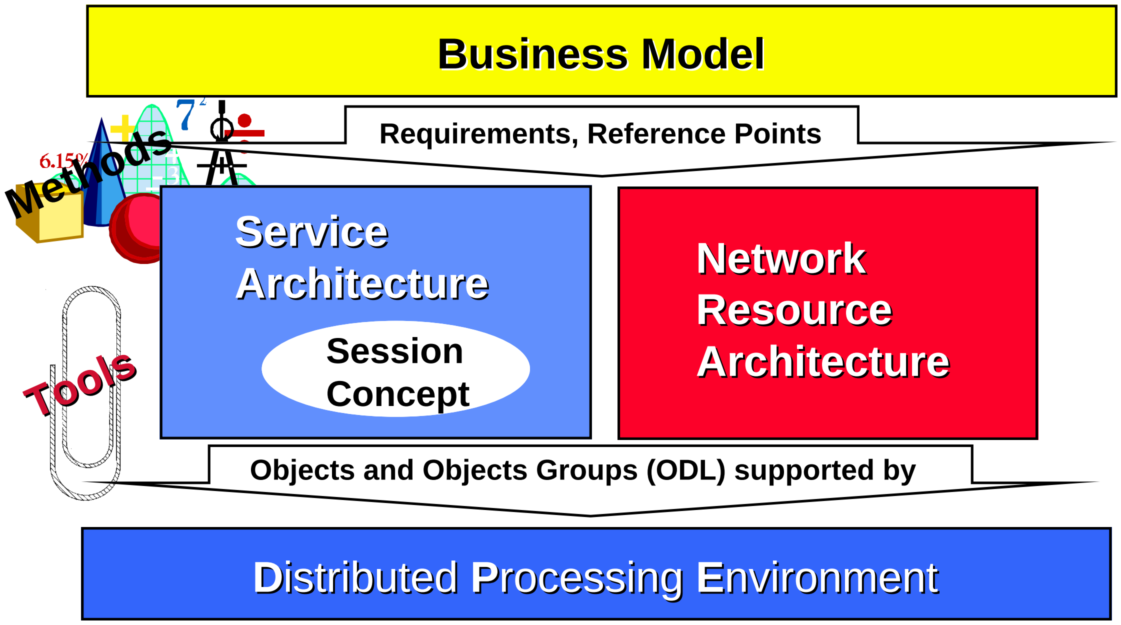
Transport Network

28

A basic architectural separation divides the TINA Architecture in three major parts:

- The DPE to get ubiquitous support for computational components, support location transparency. It was isolated because it provides common mechanism supporting all TINA applications
  - The DPE is made up of Computing supporting DPE implementations on nodes interconnected by a kernel Transport Network (kTN) for communication (e.g. IP)
  - The DPE can be split into multiple domains providing inter-operation between the domain through the IOP (CORBA)
- The Network Resource Architecture to provide modeling of Network resources in a technology transparent way. This will promote re-use and will make it possible to provide rapid changing services on a slow changing network environment. The Provisioning of Transport Connections to transport user data is provided though the Transport Network.
- The Service Architecture to provide the mechanisms and re-usable components to rapidly develop and deploy (using the DPE) services.

What is TINA



Business Model

Requirements, Reference Points

Methods

Service Architecture

Session Concept

Network Resource Architecture

Tools

Objects and Objects Groups (ODL) supported by

Distributed Processing Environment

The TINA Business model provides a framework in which stakeholders (e.g.. Telco's) can set the context for the provisioning of TINA services. It includes the concept of stakeholders, business roles, contracts and reference points. It also provides the requirements for the TINA system to provide the TINA service(s). Basic business roles identified so far: Customer, Retailer, 3Party Service Provider, Connectivity Provider, Broker.

Service Architecture:

Describes the information and computational objects supporting generic functions to provide TINA services. It provides the concept of Session (Service, Access and Communication) plus how these are combined in TINA services for the Customers. It also deals with Service Management.

Network Resource Architecture:

Provides an abstraction of the network resources. Its Informational and Computational objects support the functions to perform and manage connections based on Third party control paradigm and provides the FCAPS management separation.

Distributed Processing Environment:

Provides the description of the platform services on which TINA computational objects are run. It hides the location of the objects from its clients.

Methods:

The ODP viewpoints are applied (Business, Informational, Computational, engineering and Physical). TINA does not address the Physical viewpoint (protocol stacks, OS descriptions) to remain technology independent.

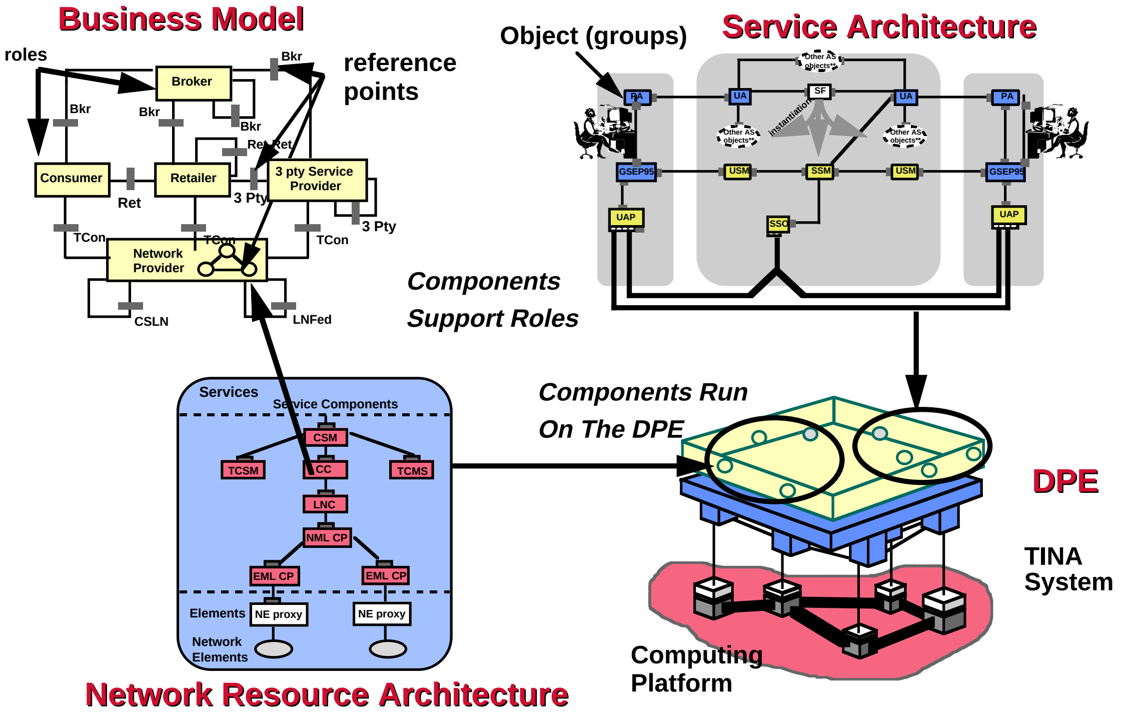
The tools currently used are:

StP and ObjectPartner (Verilog) to manage OMT object and relation descriptions for the Informational and Computational viewpoints.

Platytools (based on ORBIX) to compile ODL including an IDL compiler to make computational interface specifications executable on the DPE

ACE (CSELT) to capture ODL specifications and perform simulations based on OMT and IDL

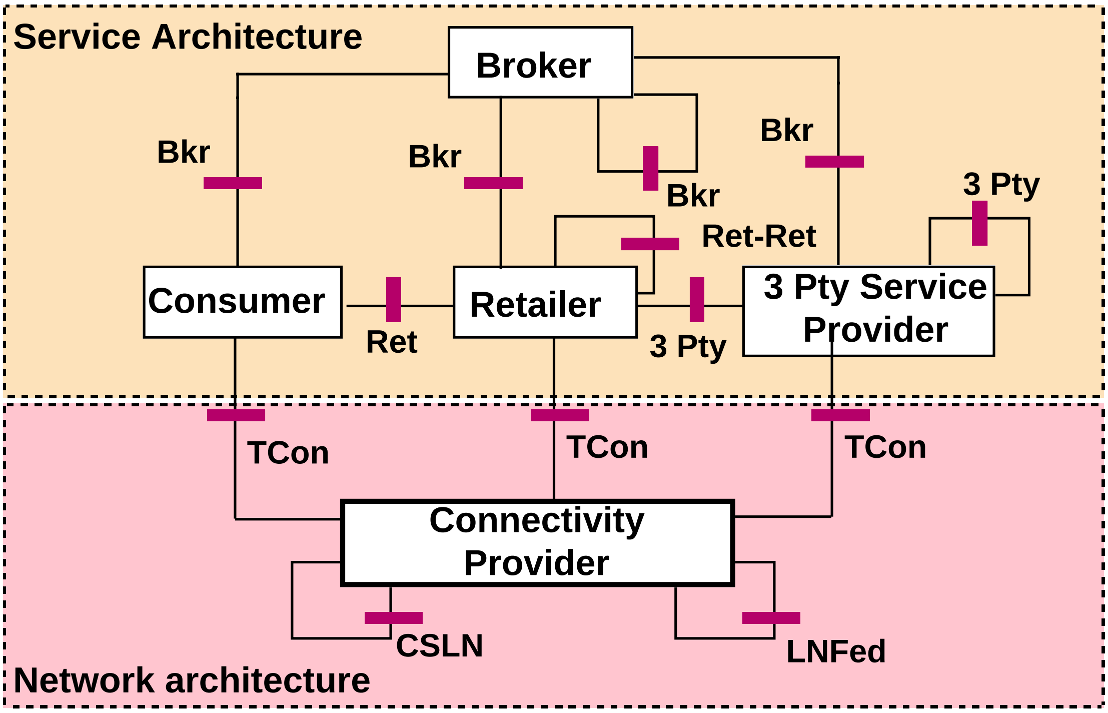
Basic TINA Components



30

This slide visualises the relationships between the different TINA components

Business Roles and Reference Points



The following types of businesses are identified:

• Consumer: a stakeholder that consumes various services.

• Retailer: sells different kinds of services to the consumers (can be compared with supermarket).

• Broker: provides a specific service for getting information how to reach other stakeholders and how to find certain services. The service which is provided by the broker is comparable with search engines in the Internet. The broker is responsible for commercial assignments, it handles subscription, accounting, security etc. (can be compared with an independent regulator or yellow pages).

•Third-party service provider: provides services to stakeholders other than consumers (retailers or other third-party providers). A third-party service provider could be a content provider. (can be compared with wholesale trader).

• Connectivity (network) provider: provides transport services. The connectivity provider offers an interface the retailers and third-party service provider businesses which enables them to request connections for offering the services to their customers.

The following Reference Points are defined between the roles:

• Ret	Retailer relationship
• 3pty	Third-party service relationship
• Bkr	Broker relationship
• TCon	Terminal connection relationship
• ConS	Connectivity service relationship
• LNFed	Layer network federation relationship
• CSLN	Client-server layer network relationship

Service Architecture

TINA Service Architecture provides

- **a generic re-usable set of components for rapid service development and provisioning**
- **support for a wide range of services (communication, information, multi-party, multi-media)**
- **management and customization of services**
- **open interfaces supporting an open (multi-player) environment to allow:**
  - **third party development of applications**
  - **interoperability amongst stakeholders**
  - **service, personal, terminal and session mobility**
- **integration of non-TINA systems and services**

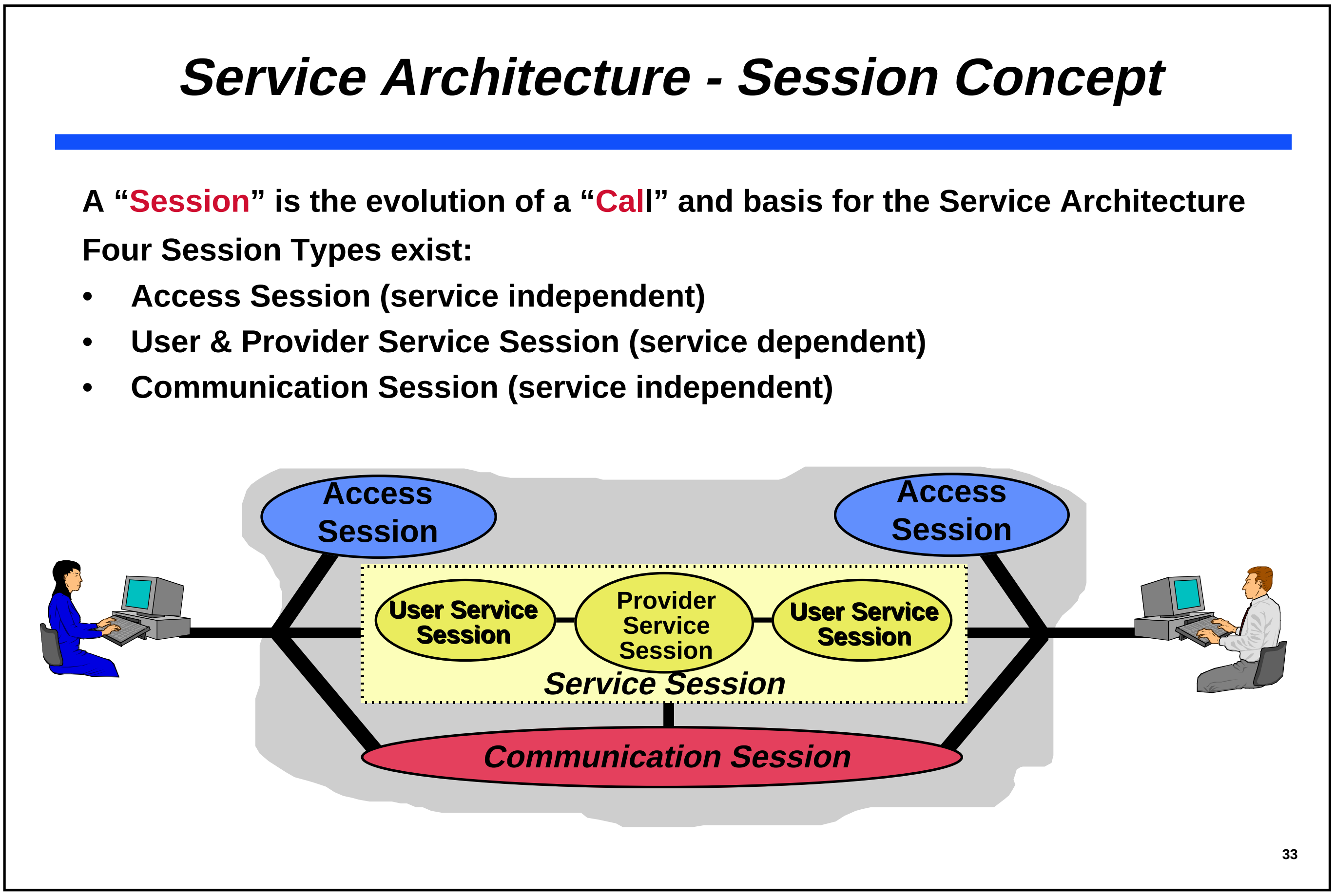
The Service Architecture defines a set of principles for providing services. It is based on the session concept which is an important original concept of TINA. A session represents the information used by all processes involved in the provision of a service for a certain duration. For example, in a multipoint conference, the information about connections, charging and user profiles may change during the onference as participants join and leave. The session helps keep such information coherent throughout the conference. A session can also be very simple (e.g., a web search).

The notion of session is further refined to allow for separation of access, service usage, and communications, Objects are separated into generic objects (common to all services) and service-specific objects (representing service logic, data, management, etc.). Several sessions are defined, corresponding to different types of activities.

The access session corresponds to the establishment of the terms and conditions of the session (e.g., authentication, selection of service profile) during the connection of a user to a system. It allows the user to start service sessions, combine sessions, and participate in several services.

The service session corresponds to the provision of the service itself (e.g., moderated multiparty conference with information retrieval capabilities) and insures overall coherence of control and management. It is divided into: the user service session that manages the state of each user's activity (local view) and resource attributes (e.g. charging context, current page), and the provider service session that contains the service logic and offers the functions allowing the user to join a session, to be invited in a session, etc. (global view).

Finally the communication session provides an abstract view of the actual transport network connections.



Service Architecture - Session Concept

A “Session” is the evolution of a “Call” and basis for the Service Architecture

Four Session Types exist:

- **Access Session (service independent)**
- **User & Provider Service Session (service dependent)**
- **Communication Session (service independent)**

Access Session

Access Session

User Service Session

Provider Service Session

User Service Session

Service Session

Communication Session

Access Session:

- provide initial entry for the user (e.g......... default configuration)
- Manages over services sessions (e.g......... subscription data)
- allows migration (downloading) of objects (software) for the usage phase

User Service Session:

- manages specific resources for one user in a service session

Provider Service Session:

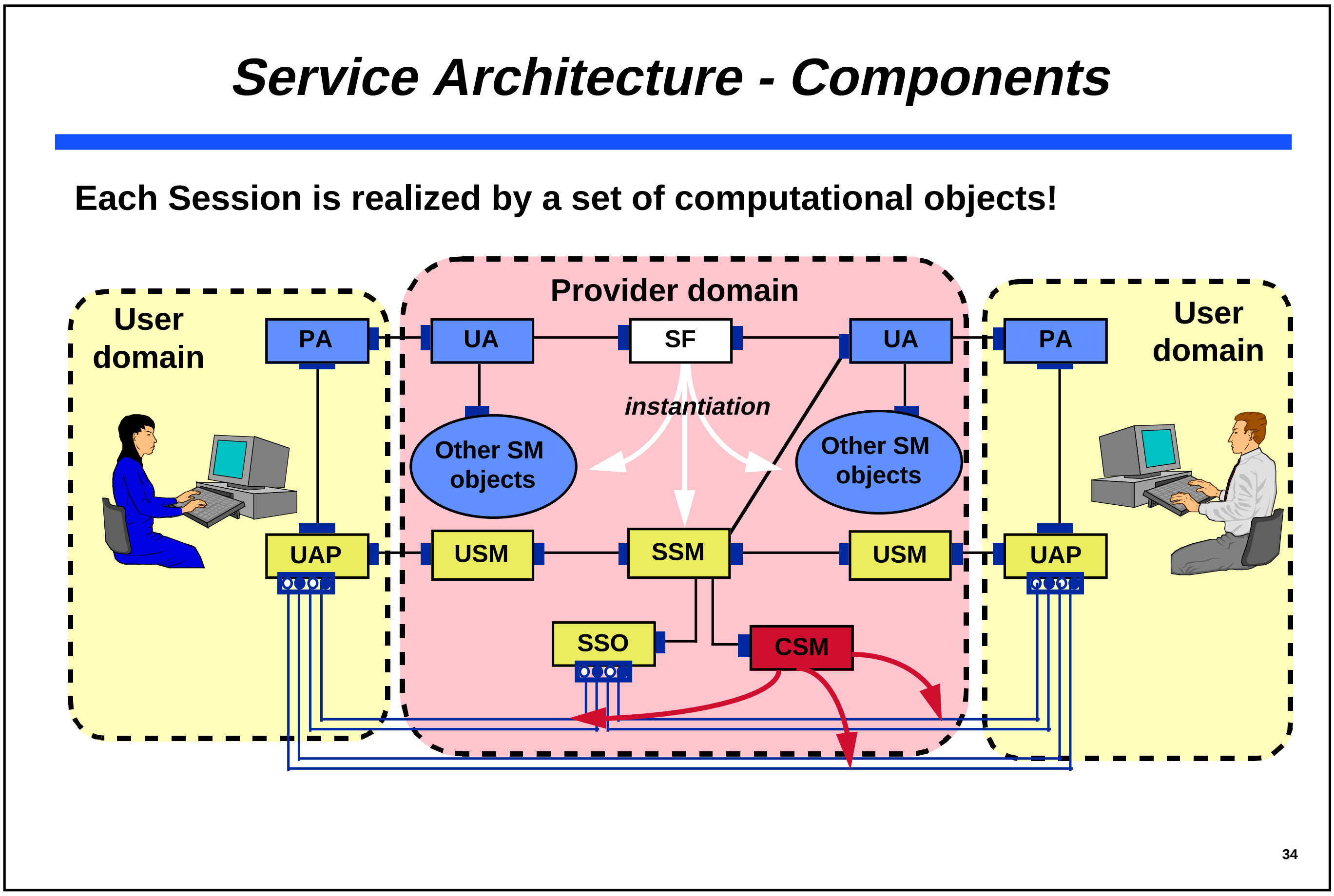
- manages resources specific to a service, manages parties and roles and their interaction (e.g......... negotiation)
- manages constrains and QoS
- manages communication

Communication Session:

- supports the communication needs of the service session
- manages communication resources

Difference between Call (ITU-T) and Session:

- session is independent of connection (e.g........ WWW page retrieval)
- session is inherently multi party
- session can persist over time without the presence of a connection (maintains context in which communication takes place)



Service Architecture - Components

Each Session is realized by a set of computational objects!

- ▬ **operational interface**
- ▣ **stream interface**
- 🟦 **computational object**
- 🟪 **comp. object group**
- — **operational interface binding**
- ▬ **stream interface binding**

The main Service components (and their properties) are displayed above:

UAP: User Application (e.g teleconferencing user part, information browsing)

UA: User Agent (forms the user part of the Access Session, contains user profiles (for e.g. access rights and security))

PA: Provider Agent (forms the provider part of the Access Session, contains initial interaction establishment functions, can be downloaded)

SF: Service Factory (user to set establish service session components

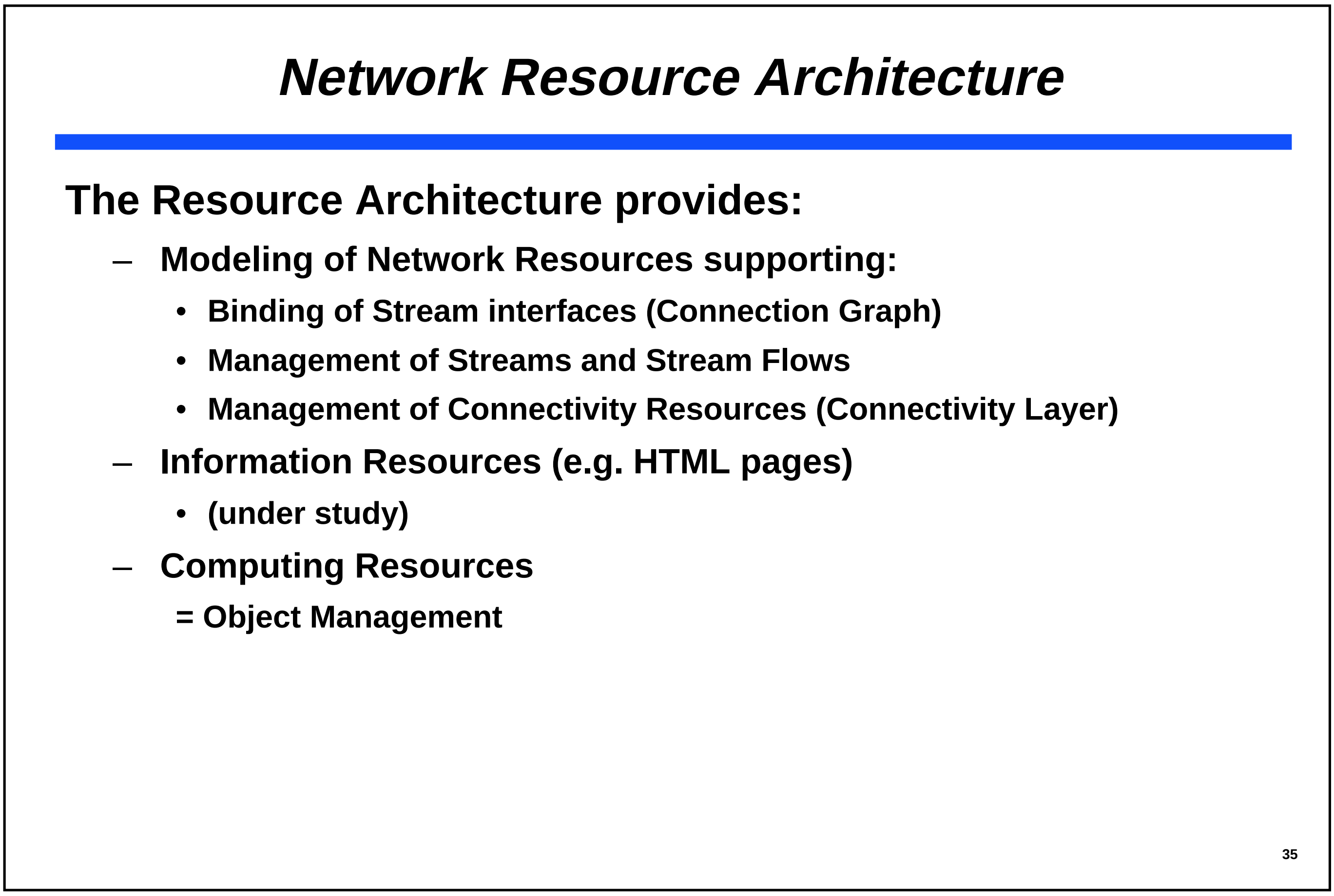
USM: User Service Session Manager (manages the user part of the service session, contains the local view of the service, provides user environment settings and customization for the user)

SM: Service Management related objects

SSM: Service Session Manager (manages the provider part of the service session, contains the global view of the service)

SSO: Service Support Object (objects needed for special service functions e.g. a video bridge control)

In the following we illustrate the necessary interactions between these objects for a simple two party multi media service.



Network Resource Architecture

The Resource Architecture provides:

- Modeling of Network Resources supporting:
  - Binding of Stream interfaces (Connection Graph)
  - Management of Streams and Stream Flows
  - Management of Connectivity Resources (Connectivity Layer)
- Information Resources (e.g. HTML pages)
  - (under study)
- Computing Resources

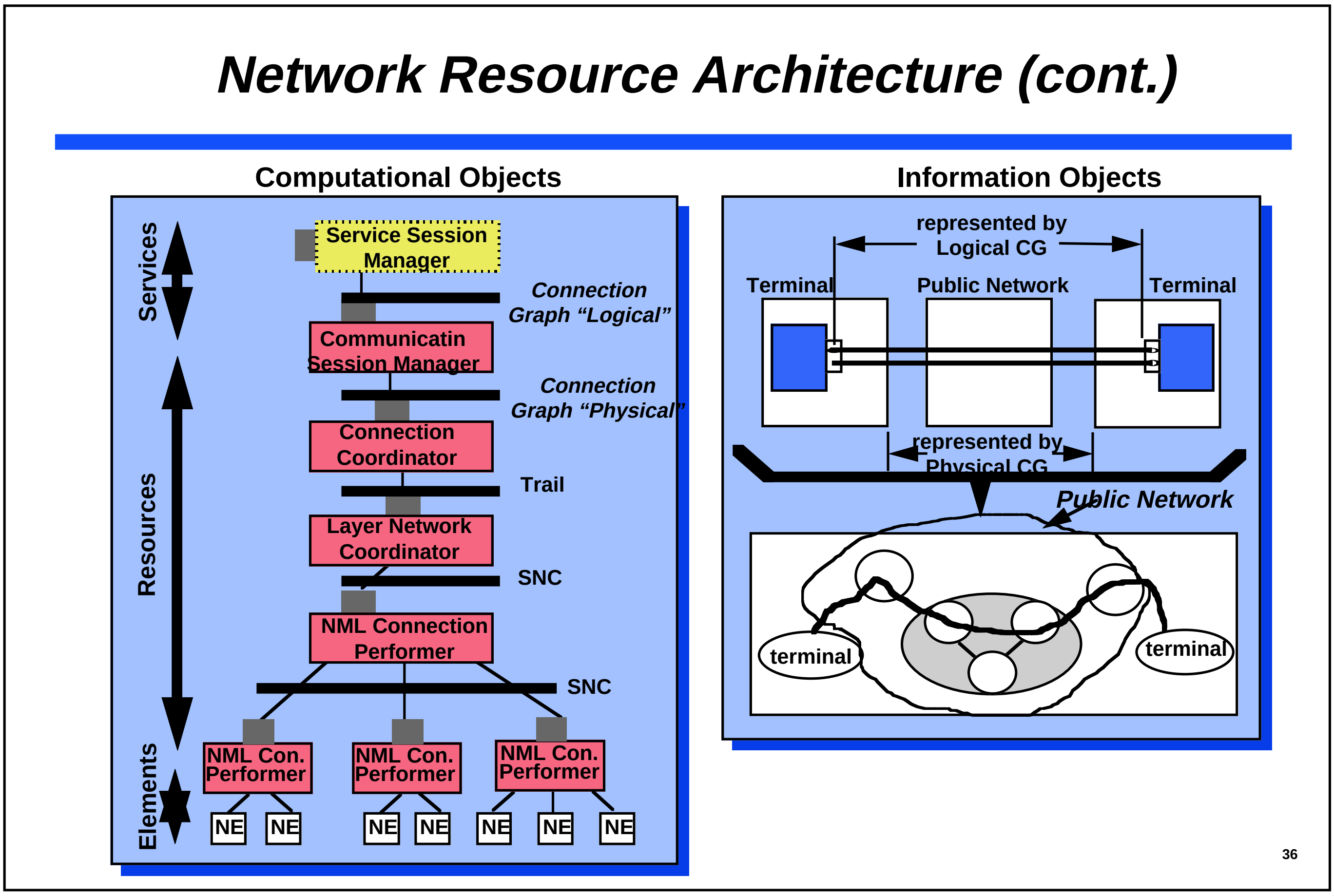
  = Object Management

The Network Architecture defines a generic technology independent model for setting up connections and managing telecommunications networks. It inherits concepts used in ITU-T and other standards bodies. It extends these concepts to integrate network control and management software for different network technologies. The Network Architecture has the following three layers:

The Communication Session layer provides service-independent interfaces for the service components to manage end-to-end communication at an abstract level. Therefore, this layer concerns both network and terminal parts of the communication.

The Connectivity Session layer abstracts all the different network technologies and provides technology-independent interfaces for the communication session layer to interconnect network-level termination points. It also handles interworking between different network technologies.

Finally the Layer Network is an abstract generalisation of each specific network technology. It deals with the set-up and management of connection within a specific network technology.



Network Resource Architecture (cont.)

The NRA Computational Objects perform the following functions:

Service Session Manager (SSM):

Establishes end to end connectivity requested bases on the user services. this request is passed on in the form of a Logical Connection Graph.

Communication Session Manager (CSM):

Coordinates the end to end stream binding between the network part (CC) and the consumer domain part (TCSM)

Connection Coordinator (CC):

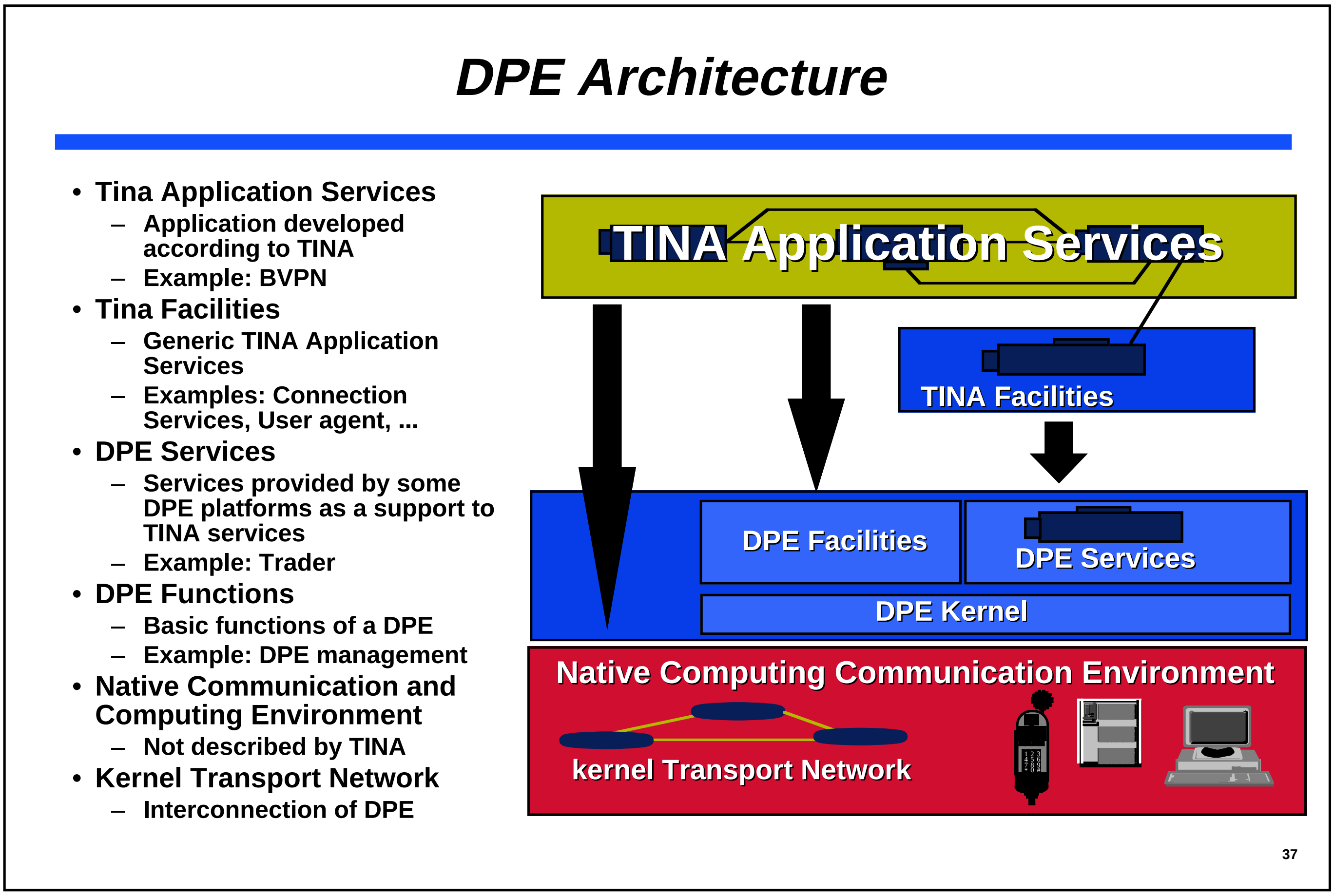
Sets up end to end connectivity over several subnetworks both in sequential (subnetworks federate) and hierarchical (subnetworks utilize connections set up by lower layer subnetworks).

Layer Network Coordinator (LNC):

Sets up the connection within a subnetwork. A subnetwork always comprises of a particular technology (e.g. SDH, ATM).

Connection Performer (CP):

Sets up subnetwork connections, e.g. a path through a switch of network of switches.



DPE Architecture

- Tina Application Services
  - Application developed according to TINA
  - Example: BVPN
- Tina Facilities
  - Generic TINA Application Services
  - Examples: Connection Services, User agent, ...
- DPE Services
  - Services provided by some DPE platforms as a support to TINA services
  - Example: Trader
- DPE Functions
  - Basic functions of a DPE
  - Example: DPE management
- Native Communication and Computing Environment
  - Not described by TINA
- Kernel Transport Network
  - Interconnection of DPE

37

TINA Application Services

- •Application developed according to TINA-C Architecture
- •Example: BVPN

TINA Facilities

- •Generic TINA Application Services
- •Examples: Connection Services, User agent, ...

DPE Services:

- Trader: Supports dynamic location & binding
- Notification: Supports of multicast, subscribed notification
- Factory: Supports instantiation of objects

DPE Kernel: support of distribution transparencies

- Repository Functions: Storage/access to permanent information
- Management Functions: Control of DPE processing functions
- Coordination Functions: Coordination of resource usage
- Security Functions: Support to security policies
- Communication Functions: Support for inter-node communication

Native Communication and Computing Environment

- •Computing nodes and peripherals (not described by TINA)

Kernel Transport Network

- •Provides interconnection of DPE

From IN to TINA

Introduction of TINA /DPE technology in specific IN domains

- **management**
- **control**
- **switching**

Advantages

- **Integration of service control and management**
- **Reuse and extensibility of service components**

Step-by-step *"TINA-risation"* of the IN

- **replacement of IN elements by means of TINA computational objects**
- **stepwise extension of DPE (CORBA) domain is prerequisite!**

TINA concepts and the inherent usage of DPE/CORBA technology are regarded as enabling technology, to provide the necessary flexibility for the emerging telecommunication service market. The advantages of introducing CORBA / TINA based solutions within the IN are mainly related to the possible rationalisation of the service aspects (e.g., the integration of service management and control), to a higher level of interoperability between applications running on DPEs, to the ability to extend service related capabilities (e.g., several points of control for a service), scalability of the service platform, vendor independence, etc.

In general, TINA concepts may be used in the following IN areas:

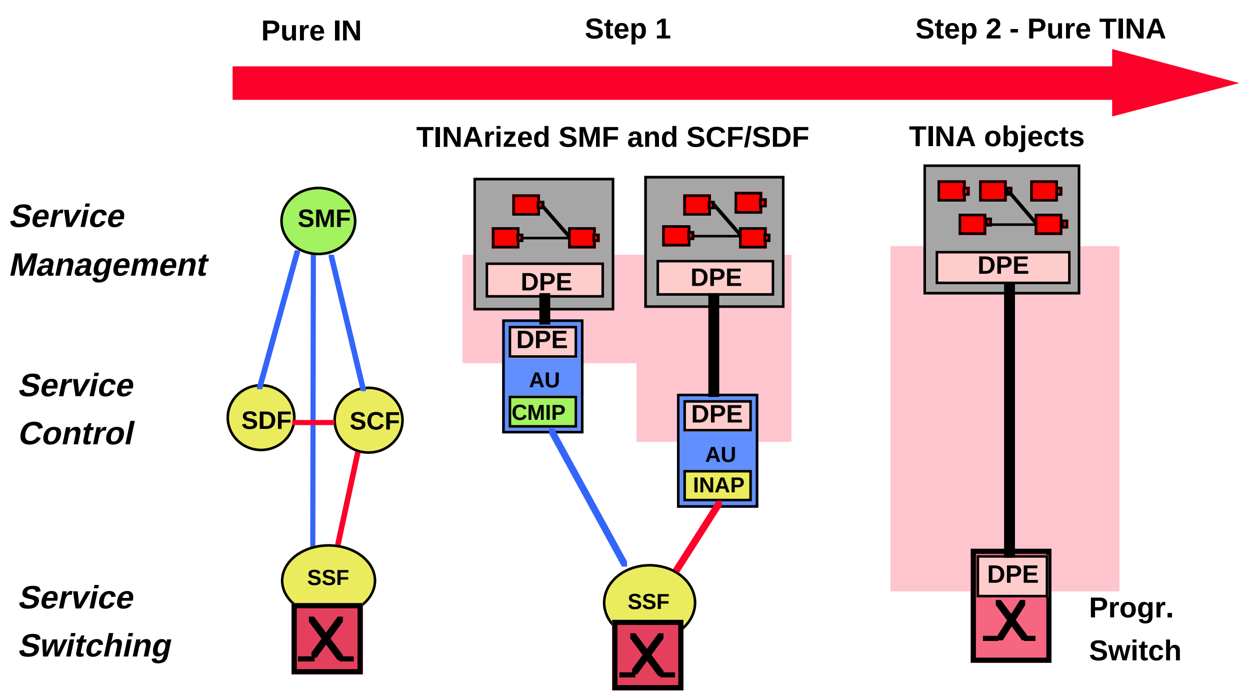
• Service Management: The introduction of TINA in the Service Management area seems to be promising because there is a lack of standardised IN management solutions and TINA offers the ability to integrate service management and control aspects by means of common objects and protocols.

• Service Data: TINA could be useful for supporting distributed incall and outcall signalling related personal profile access, in particular for personal and terminal mobility support.

• Service Control: Access Session and Service Session mechanisms could be usefully adopted in order to provide enhanced flexibility for supporting multi-party/multi-connection capabilities.

Migration of an IN-based platform toward a TINA platform means a step by step 'TINA-risation' of the IN platform elements, i.e. specific IN functional entities will be replaced by means of appropriate TINA service components, i.e. COs on top of a DPE.

From IN to TINA (cont.)



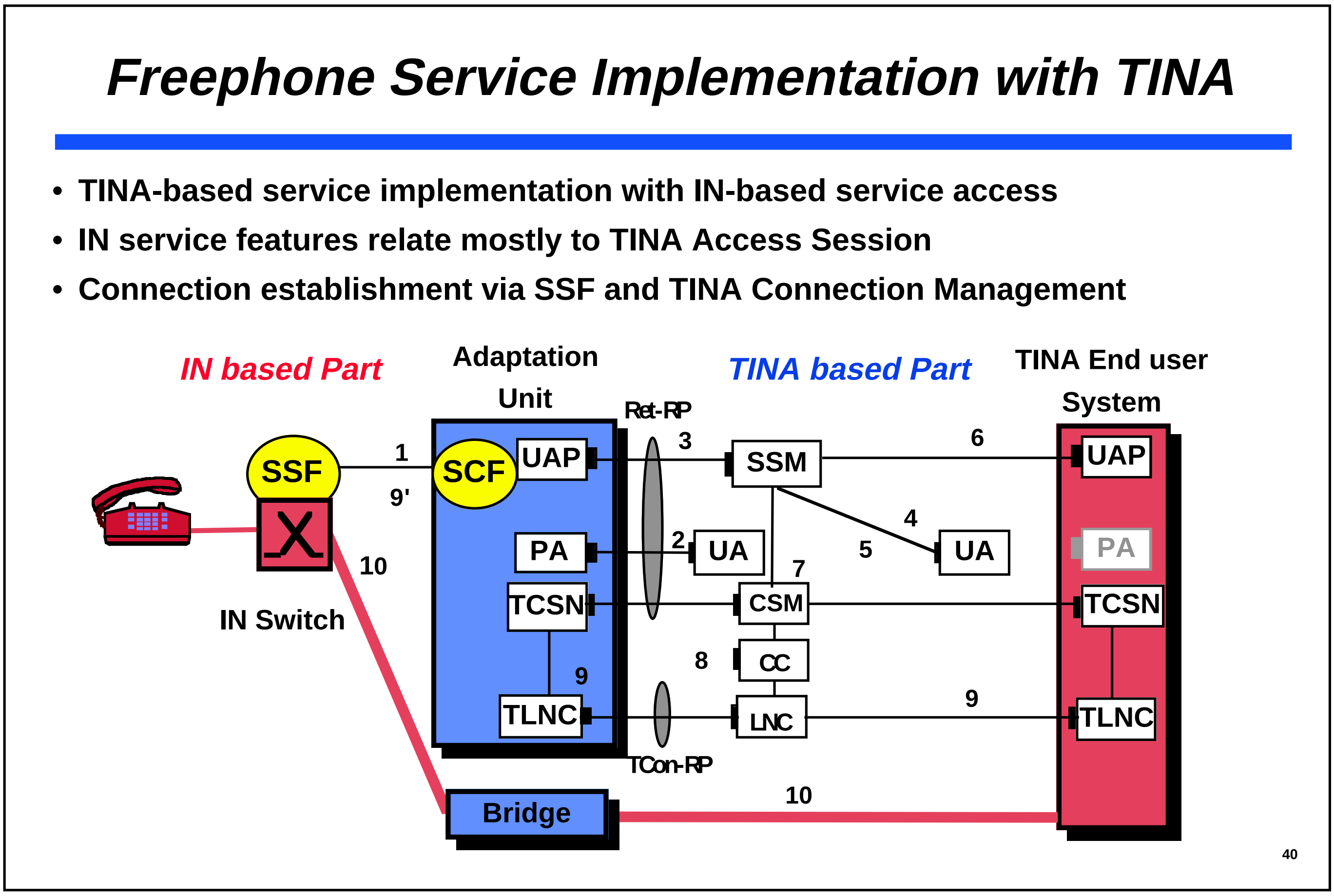
The following evolution steps seem to be most likely in this context, as illustrated above:

1. Harmonised TINArisation of IN Service Control, Data and Management

This scenario promotes the harmonised TINArisation of the SMF, SCF, SDF in order to take full advantage of the TINA service architecture and the object-oriented approach. Only the SSF remains IN compliant, since it represents the biggest IN investment. This scenario allows network operators to keep the existing Basic Call State Model (BCSM) and the INAP interfaces in operation as long as possible/required. This means that IN service control and service management are TINArised, i.e. are implemented by means of appropriate TINA COs on top of a DPE. Note that this could be realised in a TINA-based IN Service Node or in a real distributed manner by a collection of distributed TINA nodes. Correspondingly dedicated AUs are required in order to enable the interworking between the TINA COs implementing IN SCF/SDF and IN SMF capabilities, and the IN SSF for the sake of service control and management. One basic rationale for an operator to adopt this migration step is to introduce TINA concepts for the realisation of IN services on top of an existing narrowband network, particularly for new operators without existing IN service infrastructure.

2. TINA to the switch

This scenario goes one step beyond the previous one by also replacing the IN SSF. Therefore it represents the ultimate evolution step toward TINA. This scenario will be interesting with the introduction of new (broadband) switch generations, such as emerging programmable switches, which offer a very granular interface in order to control switching functions. This (proprietary) interface is described in terms of events (sent from the switch to the host controller) and commands (sent from the host computer to the switch). In this sense the "call processing" could be aligned to the TINA "call control" capabilities (i.e., the TINA service and connection management architecture). By this approach the INAP interface is no longer required and the exchanges can directly invoke the operations provided by the TINA computational objects via the DPE.



Freephone Service Implementation with TINA

- TINA-based service implementation with IN-based service access
- IN service features relate mostly to TINA Access Session
- Connection establishment via SSF and TINA Connection Management

Below we assume that IN services, i.e. IN service logic and data, will be realised by appropriate COs in the TINA environment. In this context we concentrate on the AU between an IN SSF and the TINA part, also called IN(-SCF)-AU. The AU looks from the IN side as an SCF, whereas on the TINA side the AU behaves like a TINA (end user) system comprising multiple COs and maps SS7-based INAP operations invoked by the IN SSF into TINA CO method invocations via a DPE and vice versa. Thereby, the AU enables the establishment of a (service) control relationship between an SSF in the IN domain and a Service Session Manager (SSM) in the TINA domain, i.e. the TINA SSM controls the IN SSF and the related Connection Control Function (CCF) in the switch through the AU.

Freephone Example

1.-2. Call will be initiated by a TINA user from a non-TINA terminal identical to the previous scenario.

3. SSM will receive from the UAP a service invitation for the called party.

4.-5. SSM contacts the corresponding named UA of the called party, the UA of the called party interacts with the other access session objects, such as a personal profile CO, which returns the appropriate terminal information (in accord to time of day or call origin, etc.).

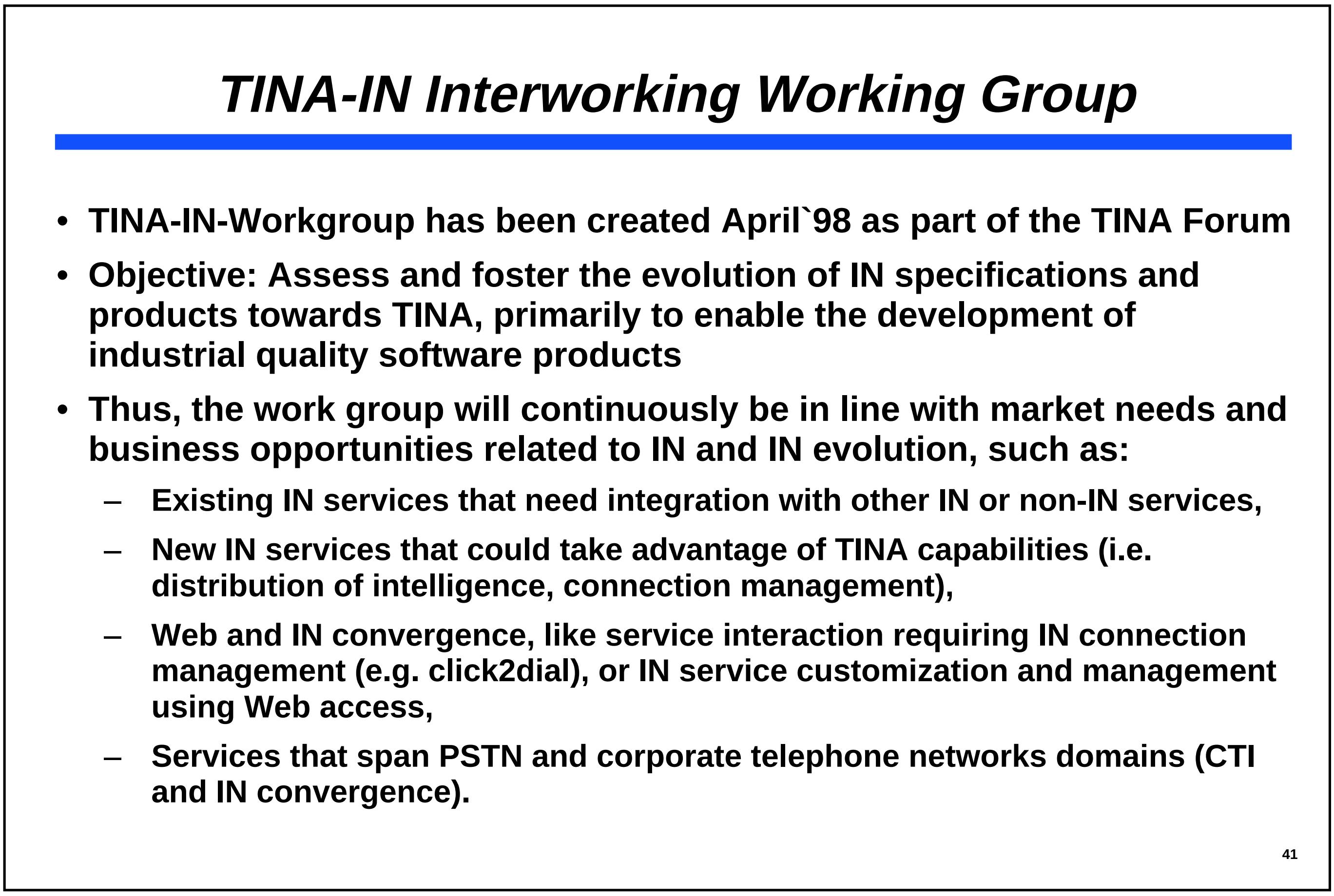
6. This information is used by the SSM to deliver the invitation to the UAP of the called party.

7. In case of an acceptance, the SSM generates a logical connection graph, from the AU Bridge to the end system of the called TINA User, which will be passed to the CSM.

8.-9.The CSM creates a physical connection graph and delivers it to the CC. The CC establishes via the LNC and TLNCs the connection from the AU Bridge to the end system of the called TINA User.

9’. In parallel the AU sends a Connect INAP Operation to the SSF in order to establish via the CCF a connection from the initiating switch to the AU bridge.

10. The connection is established. Charging treatment is not considered in the example!



TINA-IN Interworking Working Group

- **TINA-IN-Workgroup has been created April`98 as part of the TINA Forum**
- **Objective: Assess and foster the evolution of IN specifications and products towards TINA, primarily to enable the development of industrial quality software products**
- **Thus, the work group will continuously be in line with market needs and business opportunities related to IN and IN evolution, such as:**
  - **Existing IN services that need integration with other IN or non-IN services,**
  - **New IN services that could take advantage of TINA capabilities (i.e. distribution of intelligence, connection management),**
  - **Web and IN convergence, like service interaction requiring IN connection management (e.g. click2dial), or IN service customization and management using Web access,**
  - **Services that span PSTN and corporate telephone networks domains (CTI and IN convergence).**

The best forum to watch out for the IN migration towards TINA is the TINA-IN Working Group.

The TINA-IN-Workgroup has been created beginning of April 1998 as part of the structure of the TINA Forum, with the objective of assessing and fostering the evolution of Intelligent Network specifications and products towards TINA. The ultimate goal of the work group is to enable the development of industrial quality software products, by proposing implementable specifications to real business needs. Thus, the work group will continuously be in line with market needs and business opportunities related to IN and IN evolution, such as:

• Existing IN services that need integration with other IN or non-IN services,

• New IN services that could take advantage of TINA capabilities (i.e. distribution of intelligence, connection management),

• Web and IN convergence, like service interaction requiring IN connection management (e.g. click2dial), or IN service customization and management using Web access ;

• Services that span PSTN and corporate telephone networks domains (CTI and IN convergence).

Membership:

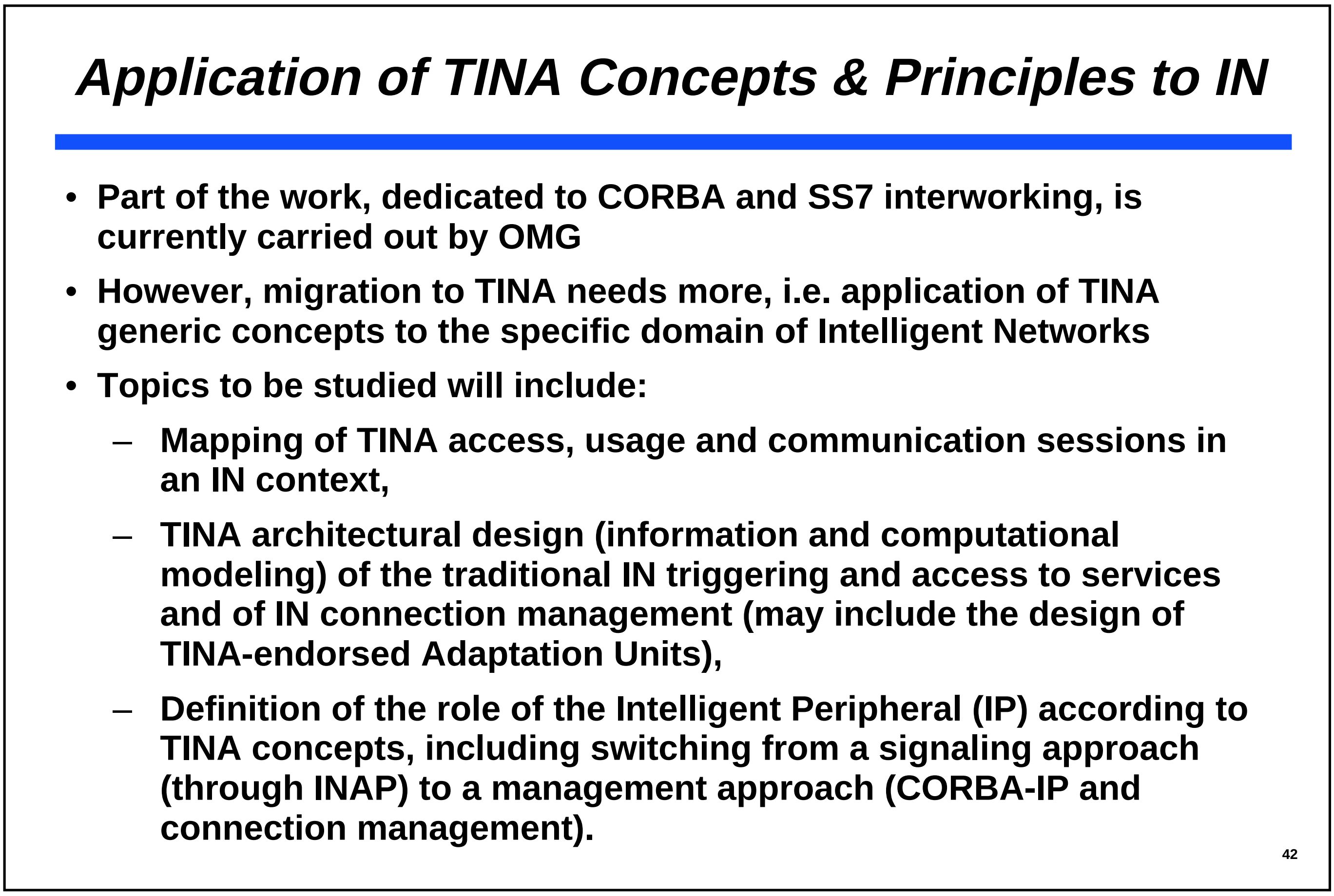
The three traditional types of actors of TINA can find interest in joining and influencing the TINA-IN-WG, and in using its results : Telecom operators, Telecom vendors and IT vendors.

A web page dedicated to the TINA-IN-WG is available at “http://www.tinac.com/98/WORKGROUPS/IN/Main.htm”.

The charter (Green Paper) can be found under “http://www.tinac.com/98/WORKGROUPS/IN/General/GreenPaper.html”

The group is open to any TINA member company, and any company (whether TINA member or not) is invited to answer to any RFP issued (see next slide).

Contact: Didier GUY - CNET DAC/ARP, FRANCE TELECOM (Didier.Guy@cnet.francetelecom.fr)



Application of TINA Concepts & Principles to IN

- **Part of the work, dedicated to CORBA and SS7 interworking, is currently carried out by OMG**
- **However, migration to TINA needs more, i.e. application of TINA generic concepts to the specific domain of Intelligent Networks**
- **Topics to be studied will include:**
  - **Mapping of TINA access, usage and communication sessions in an IN context,**
  - **TINA architectural design (information and computational modeling) of the traditional IN triggering and access to services and of IN connection management (may include the design of TINA-endorsed Adaptation Units),**
  - **Definition of the role of the Intelligent Peripheral (IP) according to TINA concepts, including switching from a signaling approach (through INAP) to a management approach (CORBA-IP and connection management).**

Application of TINA principles and concepts to IN evolution

As already mentioned, the TINA-IN-WG deals with IN to TINA evolution. Part of the work, dedicated to CORBA and SS7 interworking, is currently carried out by OMG (Object Management Group) with the CORBA/IN RFP (Request For Proposal). However, migration to TINA needs more than the ability for a CORBA application (e.g. CORBA SCP) to interwork with SS7-enabled switching equipments.

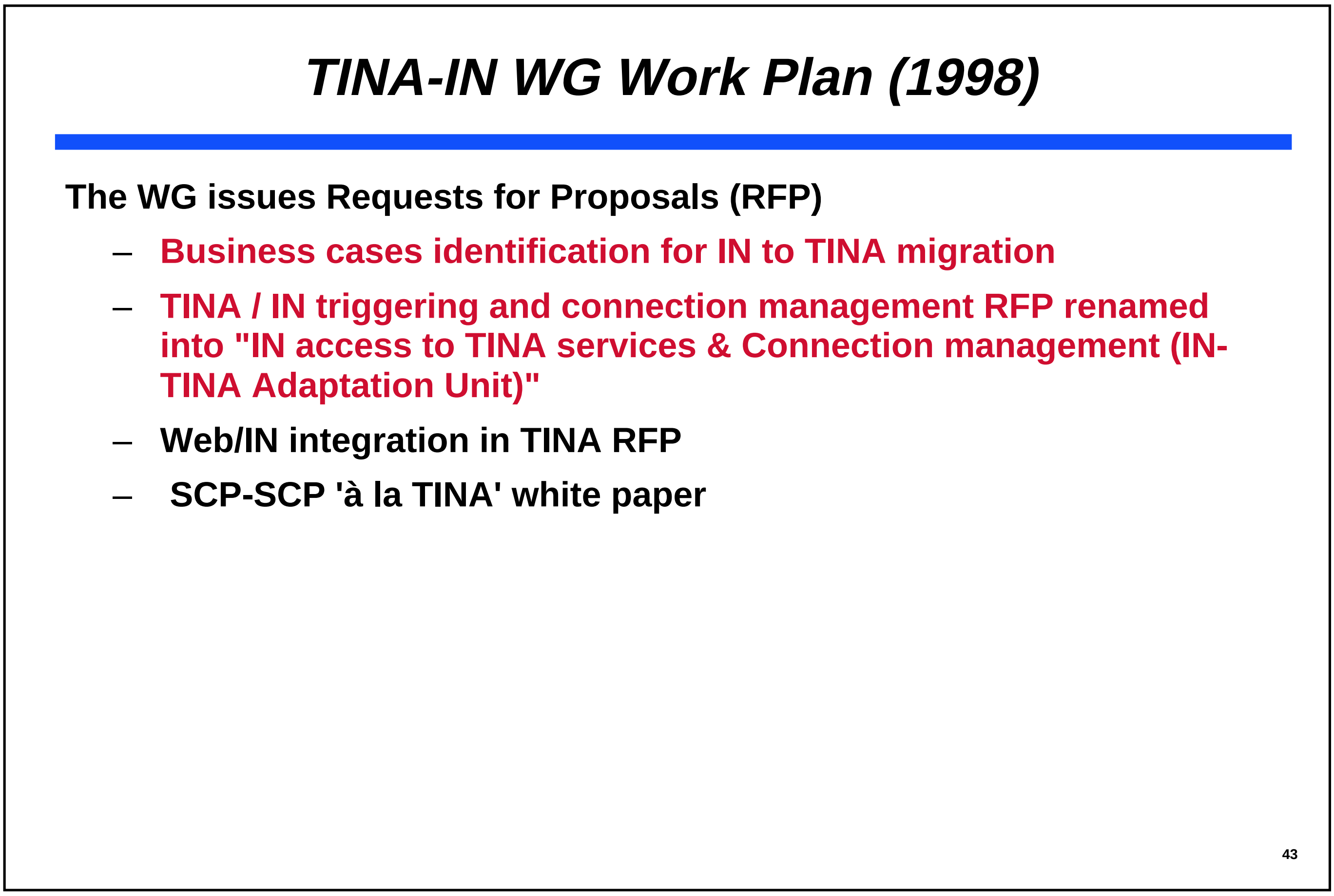
Evolution from IN to TINA also deals with applying TINA generic concepts to the specific domain of Intelligent Networks (including current status and foreseen evolutions), which means for instance assessment of the impact of TINA on IN business model and on IN elements design (SCP, IP...).

To be more specific, topics to be studied will include:

• mapping of TINA access, usage and communication sessions in an IN context ;

• TINA architectural design (information and computational modeling) of the traditional IN triggering and access to services and of IN connection management ; this topic may include the design of TINA-endorsed Adaptation Units ;

• definition of the role of the Intelligent Peripheral (IP) according to TINA concepts, including switching from a signaling approach (through INAP) to a management approach (CORBA-IP and connection management).



TINA-IN WG Work Plan (1998)

The WG issues Requests for Proposals (RFP)

- Business cases identification for IN to TINA migration
- TINA / IN triggering and connection management RFP renamed into "IN access to TINA services & Connection management (IN-TINA Adaptation Unit)"
- Web/IN integration in TINA RFP
- SCP-SCP 'à la TINA' white paper

Work Progress

To achieve the above-mentioned objectives, the TINA-IN-WG will issue Requests For Refinements & Specifications (RFR/Ss) related to the evolution of IN towards TINA, evaluate responses and submit them to the TINA Architecture Board (TAB) and TINA Forum (TF) for endorsement as TINA specifications.

According to the industry target mentioned above, TINA-IN-WG specifications should be based on prototypes or already available products, in order to meet current market demand, or have a significant influence on industry standards. Moreover, specifications should reach long-term stabilization by cooperating with recognized standard bodies.

This means that the TINA-IN-WG will coordinate its work with bodies such as ETSI NA6, ITU_T SG11, IETF PINT, OMG Telecom Domain Task Force, JAIN (Java Advanced Intelligent Network) and IN Forum.

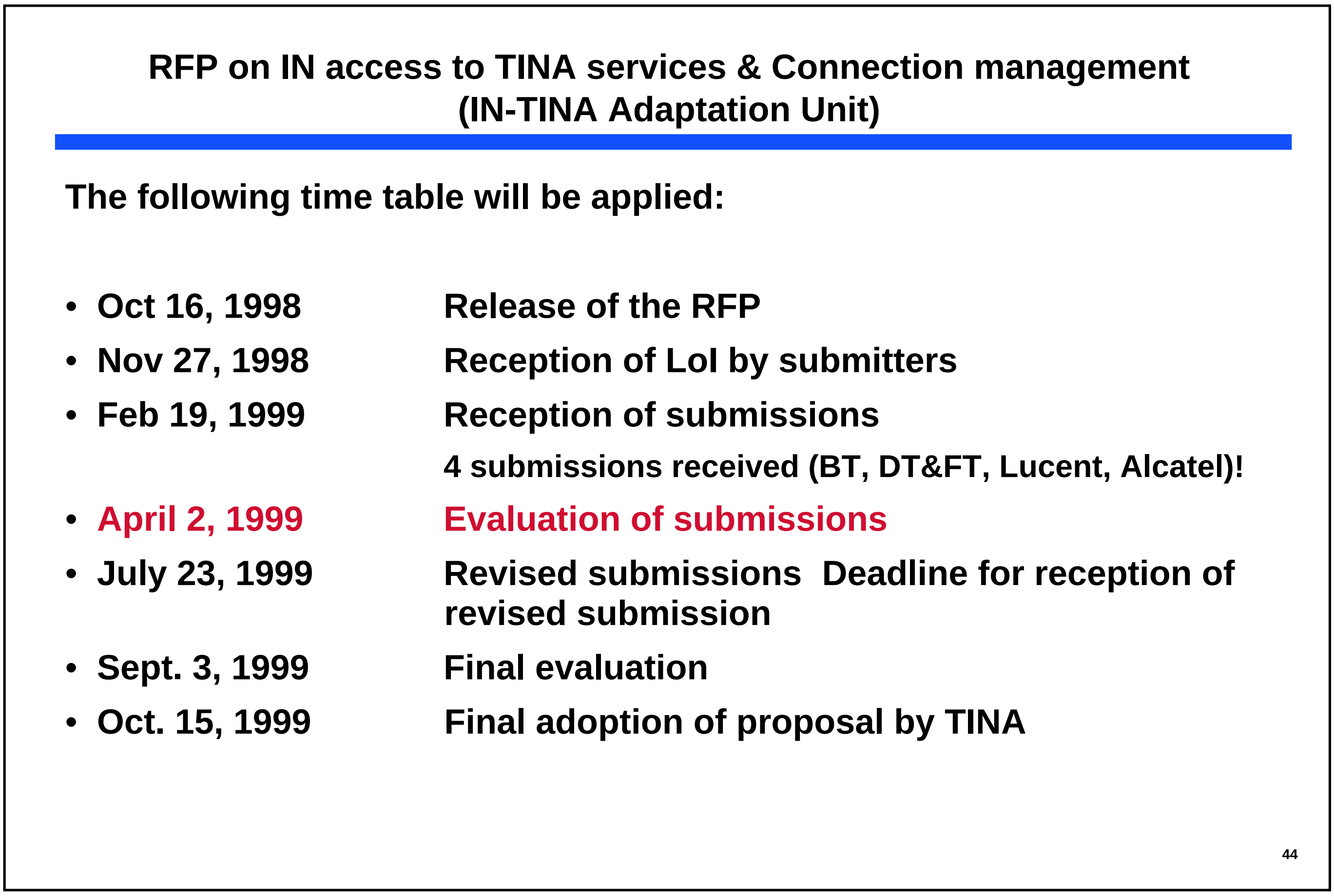
Workplan and results achieved in 1998:

• Business cases identification for IN to TINA migration (http://www.tinac.com/98/WORKGROUPS/IN/Work/Business/TINA-IN-BusCases.doc)

• TINA/IN triggering and connection management RFP release recently renamed into RFP on "IN access to TINA services & Connection management (IN-TINA Adaptation Unit)" in October 1998 (http://www.tinac.com/98/WORKGROUPS/IN/RFPs/IN-TINA-AU/IN-TINA-AU.html)

• Web/IN integration in TINA RFP release (http://www.tinac.com/98/WORKGROUPS/IN/Work/Web-IN/Web-IN.html)

• SCP-SCP 'à la TINA' white paper



RFP on IN access to TINA services & Connection management (IN-TINA Adaptation Unit)

The following time table will be applied:

• Oct 16, 1998	Release of the RFP
• Nov 27, 1998	Reception of LoI by submitters
• Feb 19, 1999	Reception of submissions 4 submissions received (BT, DT&FT, Lucent, Alcatel)!
• April 2, 1999	Evaluation of submissions
• July 23, 1999	Revised submissions Deadline for reception of revised submission
• Sept. 3, 1999	Final evaluation
• Oct. 15, 1999	Final adoption of proposal by TINA

RFP on "IN access to TINA services & Connection management (IN-TINA Adaptation Unit)" has been issued in October 1998 (http://www.tinac.com/98/WORKGROUPS/IN/RFPs/IN-TINA-AU/IN-TINA-AU.html), Editor: Didier Guy, France Télécom, Contributing companies: France Télécom, GMD-Fokus, Alcatel, CSELT, Deutsche Telekom, Nortel.

The introduction of TINA concepts and products in IN will certainly raise issues related to interactions between legacy-IN world and emerging TINA world. One of these issues relates to interactions between DPE-based software (i.e. CORBA-based SCP) and SS7-INAP enabled network elements (SSPs). This issue is currently tackled by the OMG Telecom Domain Task Force. A second issue, much closer to what TINA architecture deals with, relates to the service layer itself. According to TINA, telecommunication services are designed following Business Model and Service Architecture principles, and relying on the Network Resource Architecture for all that concerns network issues. In object oriented terms, TINA defines an object-framework for distributed-object implementations of telecommunication services, whatever these services might be. In particular, TINA has identified business roles and reference points that help designing and reusing software telecommunication service elements.

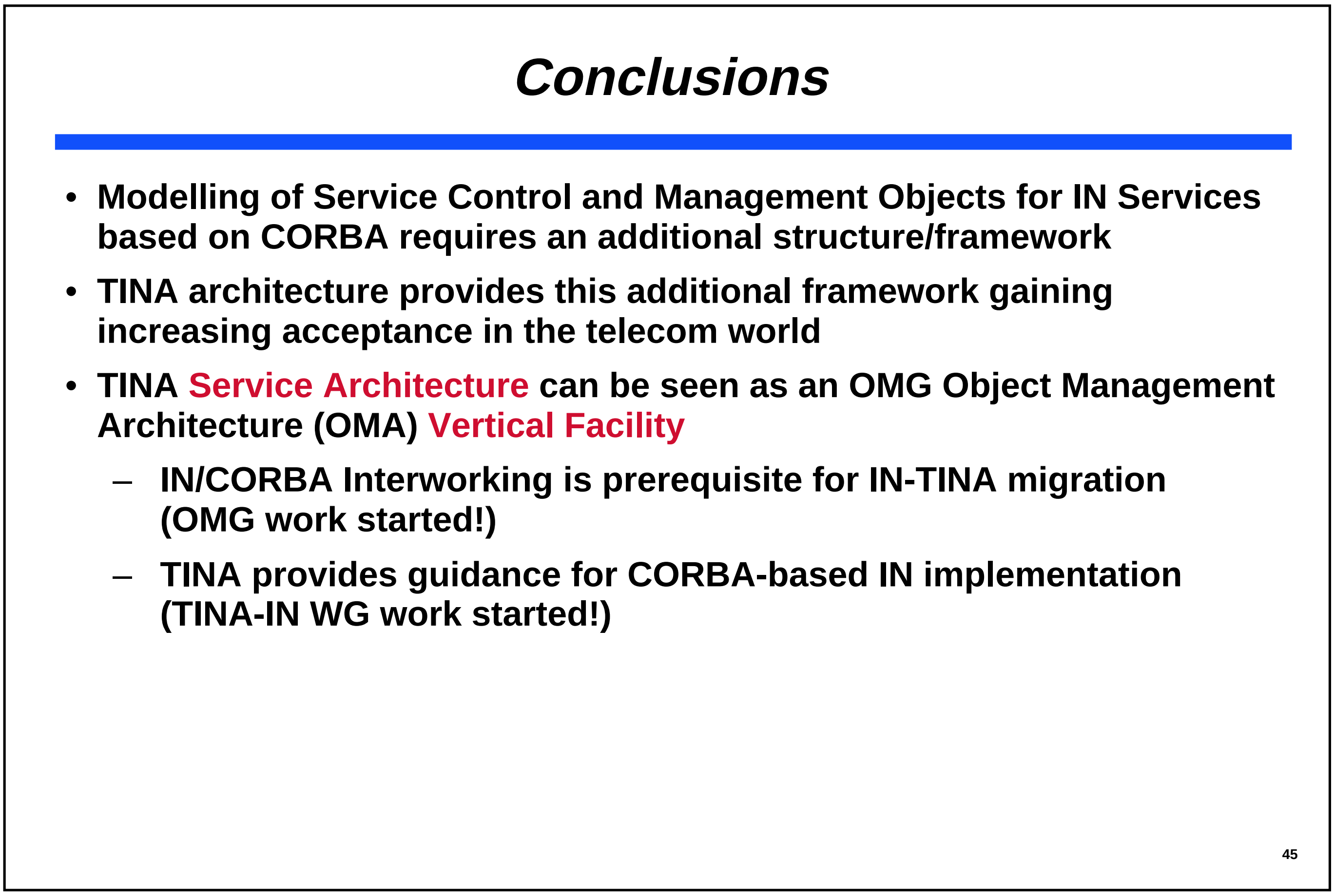
However, this framework has to be adapted in order to be used in a legacy IN world, since some of the concepts of TINA do not yet map well with IN, and vice-versa. In fact, TINA clearly identifies concepts like access session, service session and communication session, where IN deals with triggering, Basic Call State Model and call management.

When evolving from IN to TINA, two functions are to be dealt with, which can be provided by the Adaptation Unit this RFP is seeking solutions for:

1. Managing and/or controlling access to TINA services through IDL-defined operations invocations that correspond to INAP primitives; as an example, establishing access and service sessions from INAP InitialDP and other primitives;

2. Managing narrow-band connections (i.e. telephone calls) from a TINA service layer; this means end-to-end connections establishment, release and monitoring via IDL-defined operations invocations and receptions.

For both functions, it is assumed that the IN-TINA Adaptation Unit will manage an SS7/INAP - IDL translation.



Conclusions

- Modelling of Service Control and Management Objects for IN Services based on CORBA requires an additional structure/framework
- TINA architecture provides this additional framework gaining increasing acceptance in the telecom world
- TINA Service Architecture can be seen as an OMG Object Management Architecture (OMA) Vertical Facility
  - IN/CORBA Interworking is prerequisite for IN-TINA migration (OMG work started!)
  - TINA provides guidance for CORBA-based IN implementation (TINA-IN WG work started!)

Useful References for further studies

• B. Kitson: "CORBA and TINA: The Architectural Relationship", in: Proceedings of the 5th TINA Workshop, pp. 371-386, Melbourne, Australia, February 1995

• C.A.Licciardi et.al.: „Would you use TINA in your IN based Network?“, pp. 35-40, Int. Conference on IN (ICIN), Bordeaux, France, November 1996

• C. Capellmann et.al.: "Migration Scenarios for Evolving to TINA", Proceedings of the 6th TINA Conference, ISBN: 3-8007-2201, Heidelberg, Germany, September 3-5, 1996

• U. Herzog, T. Magedanz: "From IN toward TINA- Potential Migration Steps", 4th Int. Conference on Intelligence in Services and Networks (IS&N), Como, Italy, May 1997, pp. 219-228, A. Mullery et al. (Eds), ISBN: 3-540-63135-6, Springer Verlag, 1997

• U. Herzog, T. Magedanz: "Intelligent Networks and TINA - Migration and Interworking Issues", Int. Switching Symposium (ISS), Toronto, Canada, September 21-26,1997

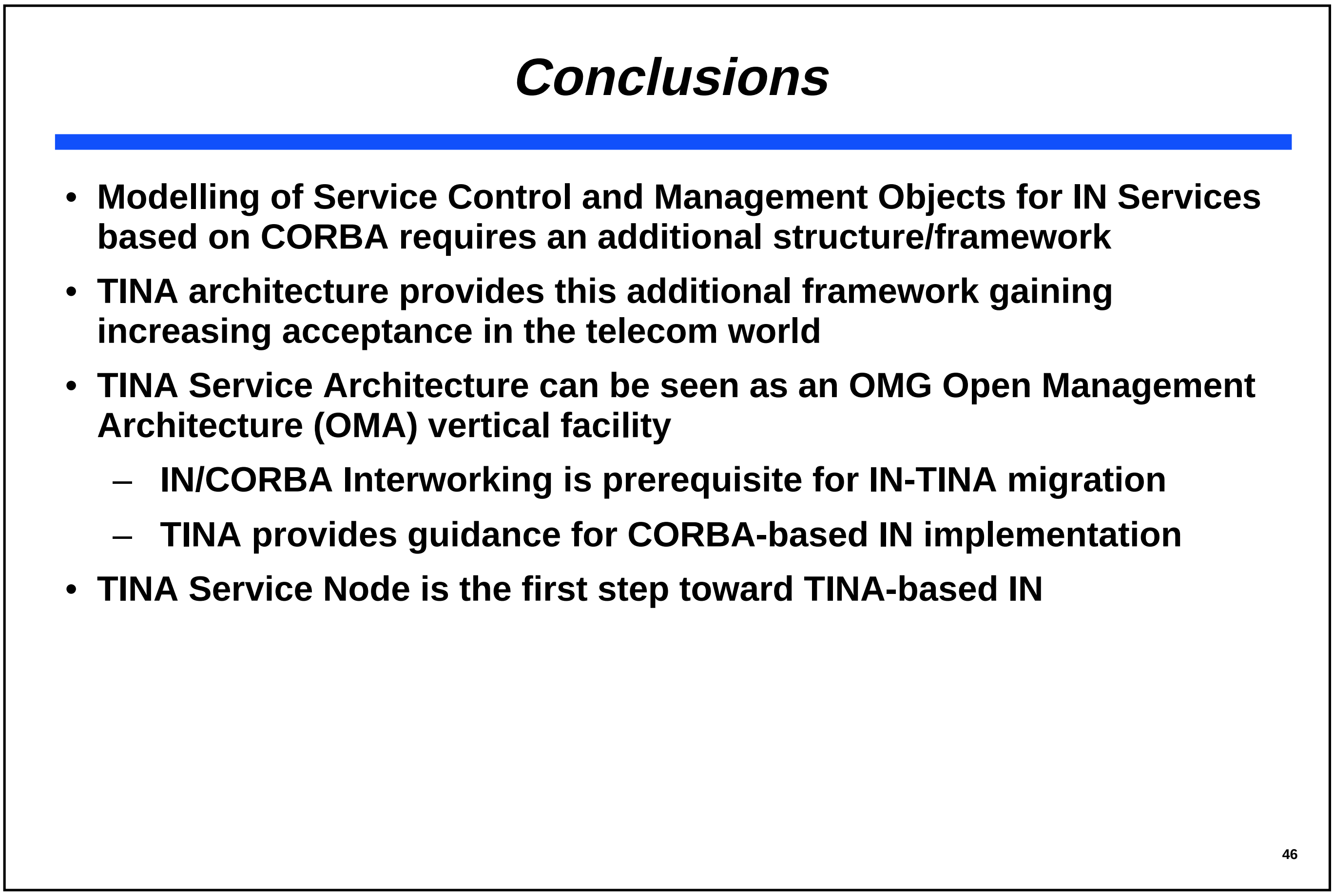
• EURESCOM Project P508 White Paper: "CORBA as an Enabling Factor for Migration from IN to TINA: A P508 Perspective", OMG DTC Document: telecom/97-01-01.xxx, January 1997

• A.Couturier, L.-P. Anquetil, "Implementing an IN Service 'à la TINA'", Proc. of 7th IEEE Intelligent Network Workshop, pp. 195-206, May 1998

• T.Eckardt, D.Guy, P.Kielhöfer, V.Pérébaskine, A.Akram, "A TINA Trial: Interworking Experience with the Legacy Telephone System", Proc. of TINA'97, pp. 70-77, Nov 1997

• P.Kielhöfer, T.Magedanz, U.Scholz, P.Schoo, "A TINA Service Node for IN Environments", Proc. of ICIN'98, pp.133-138, May 1998

• Y.Lu, P.Foliant, J.P.Van der Bijl, E.Teseling, R.SChiellius, W. Levelt, J.Kroeze, "Universal Services Platform: Implementing TINA-based service platform over the existing infrastructure", Proceedings of ICIN'98, pp.144-148, May 1998



Conclusions

- Modelling of Service Control and Management Objects for IN Services based on CORBA requires an additional structure/framework
- TINA architecture provides this additional framework gaining increasing acceptance in the telecom world
- TINA Service Architecture can be seen as an OMG Open Management Architecture (OMA) vertical facility
  - – IN/CORBA Interworking is prerequisite for IN-TINA migration
  - – TINA provides guidance for CORBA-based IN implementation
- TINA Service Node is the first step toward TINA-based IN

46

Useful References for further studies

• D.K. Brown: „Practical Issues Involved in the Architectural evolution from IN to TINA“, pp. 270-275, Int. Conference on IN (ICIN), Bordeaux, France, October 1994

• L. Demounem, H. Zuidweg: “On the Co-existence of IN and TINA“, pp. 131-148, TINA 95, Melbourne, Australia, February 1995

• B. Kitson: "CORBA and TINA: The Architectural Relationship", in: Proceedings of the 5th TINA Workshop, pp. 371-386, Melbourne, Australia, February 1995

• C.A.Licciardi et.al.: „Would you use TINA in your IN based Network?“, pp. 35-40, Int. Conference on IN (ICIN), Bordeaux, France, November 1996

• C. Capellmann et.al.: "Migration Scenarios for Evolving to TINA", Proceedings of the 6th TINA Conference, ISBN: 3-8007-2201, Heidelberg, Germany, September 3-5, 1996

• U. Herzog, T. Magedanz: "From IN toward TINA- Potential Migration Steps", 4th Int. Conference on Intelligence in Services and Networks (IS&N), Como, Italy, May 1997, pp. 219-228, A. Mullery et al. (Eds), ISBN: 3-540-63135-6, Springer Verlag, 1997

• U. Herzog, T. Magedanz: "Intelligent Networks and TINA - Migration and Interworking Issues", Int. Switching Symposium (ISS), Toronto, Canada, September 21-26,1997

• EURESCOM Project P508 White Paper: "CORBA as an Enabling Factor for Migration from IN to TINA: A P508 Perspective", OMG DTC Document: telecom/97-01-01.xxx, January 1997

• T. Magedanz, U. Scholz, P. Schoo: “A TINA Service Node for IN Environments - A First Step in IN-TINA Evolution”, International Conference on Intelligent Networks (ICIN), Bourdeaux, France, May 13-15, 1998

IN and Mobile Agents

- Mobile Agent Basics
- MA-based IN Environments

This part introduces emerging Mobile Agents (MAs) concepts, technologies and standards. In contrast to the early days of agent mania, today agent technology will be integrated with CORBA technology , where the OMG Mobile Agent System Interoperability Function (MASIF) represents an important step.

An application area which could potentially benefit greatly from the success of agent technology, but is also among the most challenging application fields possible, is the field of telecommunications. The rationale behind this viewpoint is the assumption that intelligent mobile agents may enable new and much more flexible and efficient ways for the distribution of control and management in telecommunication systems.

We will see how an MA-based IN environment can be constracted by introducing MA platforms into IN elements, thereby enabling the concept of intelligence on demand.

Introduction

- Agent paradigm has gained momentum in th e early 90s
- NO unique Agent definition
  - many languages, platforms, application areas
- Basic agent issues
  - Intelligent Agent + Mobile Agents = Intelligent Mobile Agents?
  - Agent versus Distributed Object Computing platforms
- Agent standards appeared in 1997 (FIPA, OMG MASIF)
  - MASIF conform platforms are coming integrating MA and CORBA
- Telecommunications is a promising MA application area:
  - Intelligence on demand, flexible intelligence (logic) distribution
- IN and TMN will be impacted by MA technology
- MA-Based IN Enviroments are going to be prototyped right now!

Within the last five years the paradigm of Intelligent and Mobile Agents has gained momentum in the field of software engineering. Agents are autonomous, asynchronous and intelligent software entities which, in order to fulfil their tasks, can migrate to and reside in a number of networked nodes. A variety of agent languages, architectures and platforms have been developed within the last three years in academia and industry, including AgentTCL, Telescript, Tabriz, Active X, Java and a variety of Java enhancements, such as Aglets, Voyager, Odyssey, etc. However, all of these developments are incompatible with each other and designed for specific environments and application domains. One main reason was the lack of standards and consequently a lack of a unique agent definition. It is to be noted that the work on the standardization of more generic agent platforms and facilities has started late in 1996/7 within the FIPA, OMG and IETF.

The application of the agent technology, however, has already a history and spreads over many areas, including user interface/personal assistance, mobile computing, information retrieval & filtering, data mining, smart massaging, the electronic marketplace, and telecommunication services control and service / network management. The last two areas represent still new application fields for agent technology, now rapidly gaining importance in the problem area addressed by the emerging open telecommunication and information environments.

Fortunately there is today a common understanding that mobile agent technology should be seen as an enhancement of distributed object technology rather than a replacement. The reason is that both technologies offer advantages in different aspects, which only collectively provide the maximum flexibility in application design.

Therefore the recent OMG work on a Mobile Agent System Interoperability Facility (MASIF) specification can be regarded as a milestone on the road toward a unified distributed mobile object middleware, which enables technology and location transparent interactions between static and mobile objects. However, the MASIF is just the starting point for implementation of common agent platforms, since it concentrates only on the interoperability support between different agent systems.

In the following we provide a brief introduction to mobile agent concepts, illustrates what basic functionalities are covered by today's mobile agent platforms, what functionality is covered by the MASIF specification and upcoming MASIF conform mobile agent platform. In particular, we illustrate how such MA platform could be used to establish an MA-based IN environment, which provides intelligence on demand.

Agent Definition

- There is no unique Agent Definition!
- BUT: Agents can be characterised by their attributes / properties
- The only common attribute is Autonomy
  - an agent exercises control over its own activities
- Complementary (but not mandatory) attributes
  - Intelligence (goals, reasoning, planning, learning)
  - Mobility (remote execution vs. migration)
  - Social ability (communication vs. cooperation)
  - many *others* depending on application domain and origin

The term "Agent" is associated with various expectations and is used in many different contexts. Thus no Agent definition exists.

However, the following attributes are typical for agents: An agent is a self-contained software element responsible for performing part of a programmatic process. Therefore, it contains some level of intelligence, ranging from predefined rules up to self-learning AI inference machines. It acts typically on behalf of a user or a process enabling task automation.

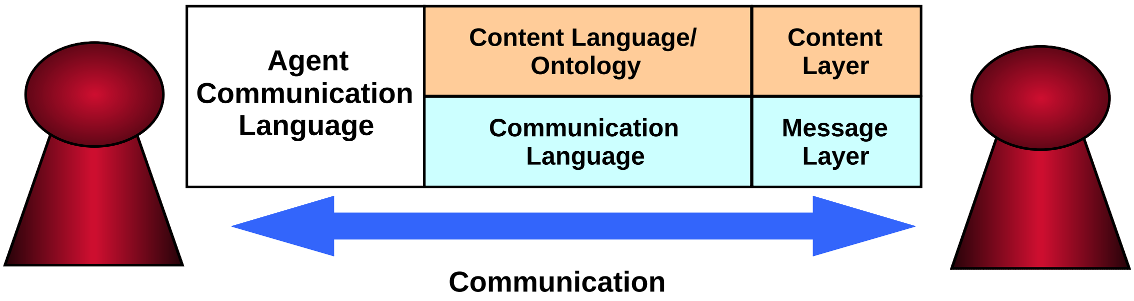
Agents operate rather autonomously and asynchronously to the user (they are often event or time triggered) and may communicate with the user, system resources and other agents as required to perform their task.

Moreover, more advanced agents may cooperate with other agents to carry out tasks beyond the capability of a single agent.

Finally, as mobile or even active objects, they may move from one system to another to access remote resources or even meet other agents and cooperate with them.

Intelligent Agent Principles

- **Intelligent Agents origin is Distributed Artificial Intelligence (DAI)**
- **Believe - Desire - Intention (BDI) Agents**
- **Reactiveness vs. Proactivenes**
- **Most important aspect is inter-Agent Communication (based on Speech Act Theory)**
  - **Agent Communication Language (ACL)**
  - **Communication Language versus Content Language**



Agent Communication Language	Content Language/ Ontology	Content Layer
	Communication Language	Message Layer

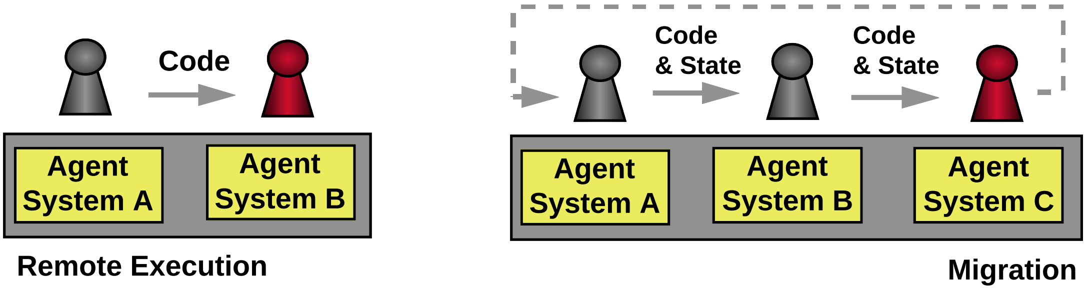
Communication

Benefits of Intelligent Agents

- **Distributed problem solving**
- **Negotiation capabilities**
- **Proactiveness - Learning capabilities**
- **Adaptability of functionality and interfaces**
- **Interoperability via high level Agent Communication**

Mobile Agent Principles

- Trend coined by Telescript and Java!
- Idea is shipping code to data vs. remote communication (RPC)
- Two approaches:
  - Remote execution
    - Agent transfer (code and data) before Task execution
  - Migration
    - Agent transfer during Task(s) execution, i.e. Agent performs specific task(s) at different nodes (Agent decides when and where to go)
    - In addition to *code/data*, also *execution state* has to be transferred!

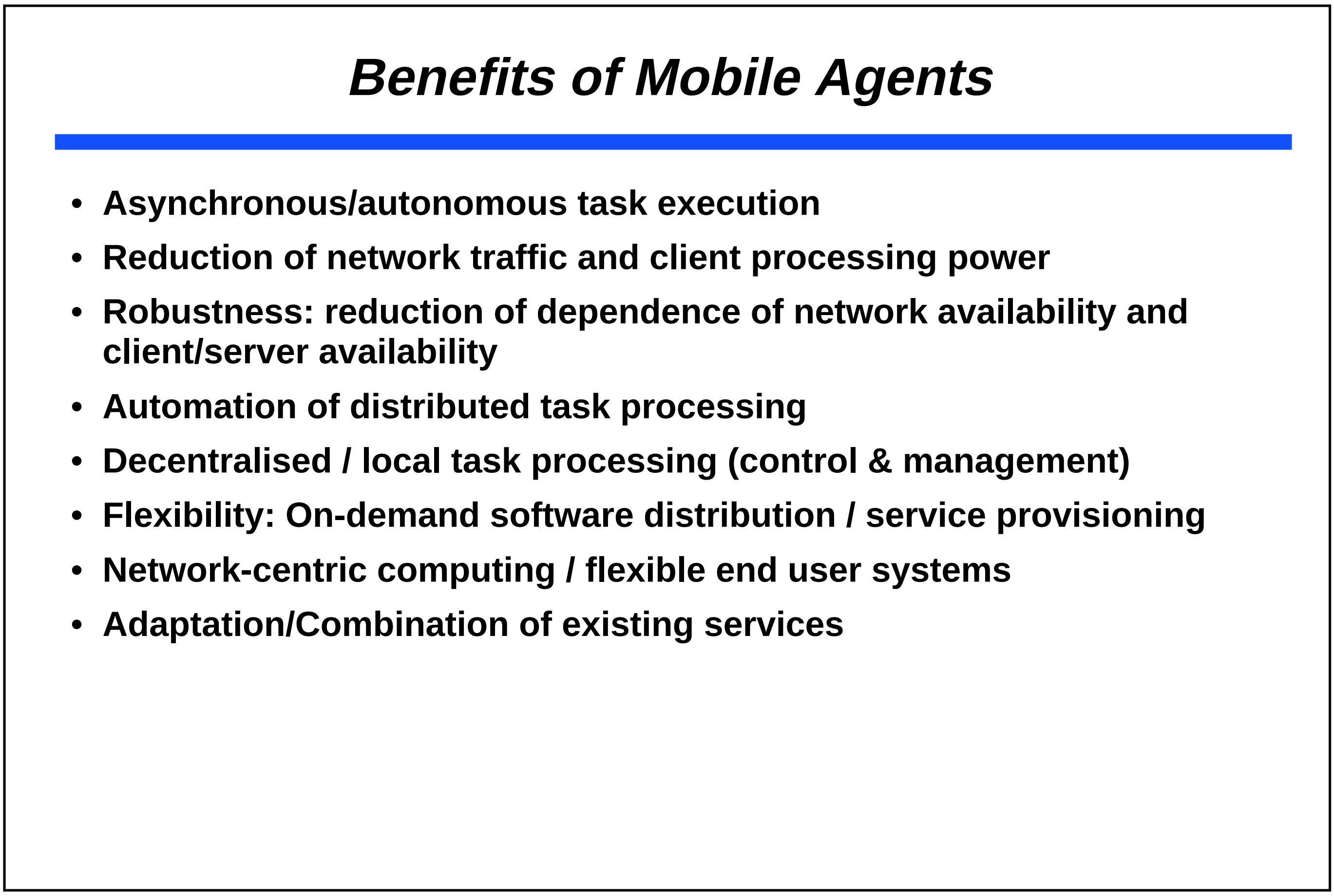


In contrast to static agents, which remain in a single location throughout their entire life time, mobile agents roam the network to perform their tasks. All aspects of an agent's behavior including execution, communication and agent mobility are supported by respective support services of an agent run-time environment. Allowing agents to move around enables them to perform their tasks locally, i.e. at the location where the involved resources/entities are located. This concept is often referred to as „Remote Programming". Remote programming represents an alternative to the traditional client/server interaction paradigm, where static components located at different computers communicate remotely via a Remote Procedure Call (RPC) mechanism.

Generally, two levels of agent mobility can be identified:

Remote Execution: An agent (i.e. program code and data) is transferred to a remote system, where it is activated and executed in its entirety. In the context of its remote execution the agent will then interact with resource objects located at the receiving server host (e.g. at agent system B - not illustrated). After performing its task(s), the agent may either be destroyed or it may stay at the server host in a dormant state in order to be re-activated at a later time.

Agent Migration: In the course of its execution, an „active" agent may move from one node to another node in order to accomplish its task progressively. An agent environment supporting agent migration allows active agents to suspend their execution, to move to another node in the network, and to resume execution from the point where it was suspended. This requires the preservation of the agent's execution state. Migrating agents may return to the originating host in order to deliver specific results.



Benefits of Mobile Agents

- **Asynchronous/autonomous task execution**
- **Reduction of network traffic and client processing power**
- **Robustness: reduction of dependence of network availability and client/server availability**
- **Automation of distributed task processing**
- **Decentralised / local task processing (control & management)**
- **Flexibility: On-demand software distribution / service provisioning**
- **Network-centric computing / flexible end user systems**
- **Adaptation/Combination of existing services**

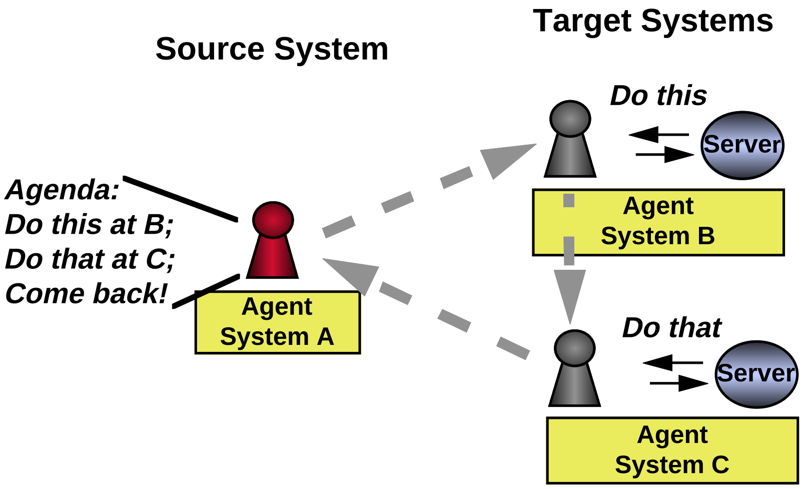
The envisaged advantages of Intelligent Mobile Agents:

- Asynchronous/autonomous task execution, i.e. client systems releasing agents may terminate, while agent controls task execution
- Reduction of network traffic and client processing power (depends on agent size and interaction pattern), ie. weak clients may outsource tasks and low bandwidth environments (e.g. wireless) may be used, since interactions are performed locally and only results may be trasnfered via the network.
- Robustness: reduction of dependence of network availability and client/server availability, i.e. already migrated agents are not anymore affected by network failures/breakdowns
- Automation of distributed task processing, i.e. agent performs predefined task at different places
- Decentralised / local task processing (control & management), i.e. dedicated agents are travelling close to the resources and thereby enabling better performance through load balancing
- Flexibility: On-demand software distribution / service provisioning
- Network-centric computing / flexible end user systems, i.e. modular service structure allows to download required or new functionality to the end system
- Adaptation/Combination of existing services, i.e. agents may act as gateways or filters integrating even multiple services

MA Benefits (cont.)

Automation of distributed task processing

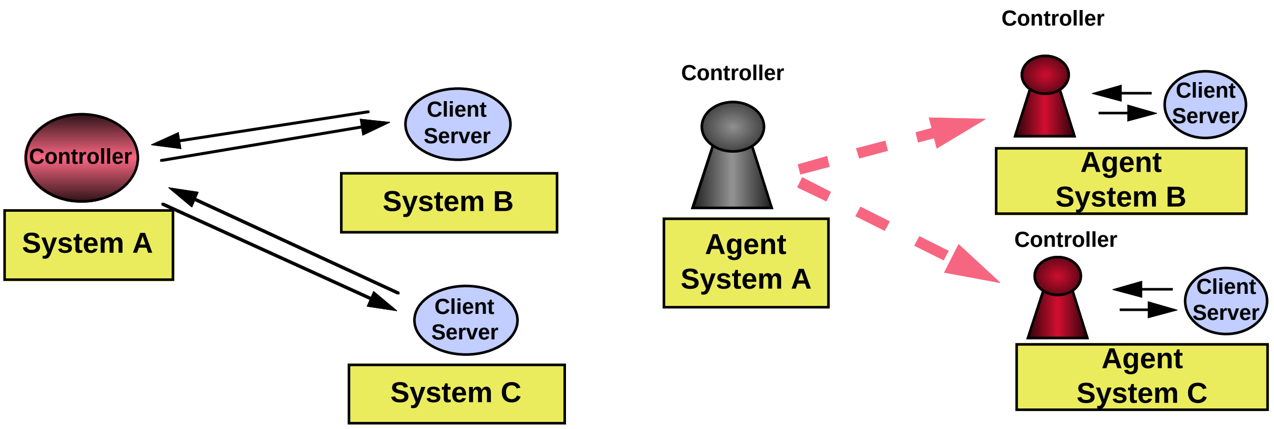
- **MA performs specific activities at different places**



MA Benefits (cont.)

Decentralised / local task processing (control & management)

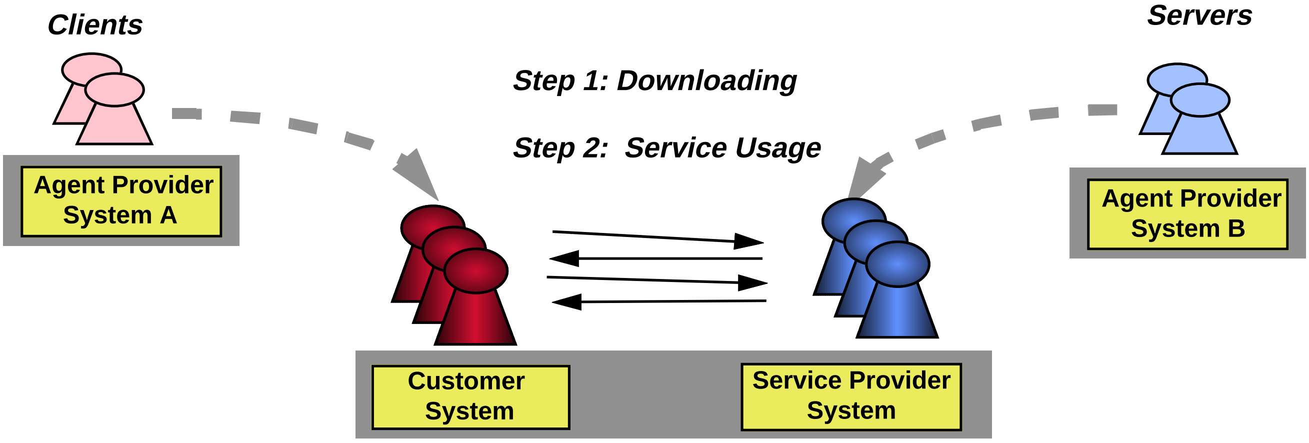
- **enables better performance, reliability, security, etc.**



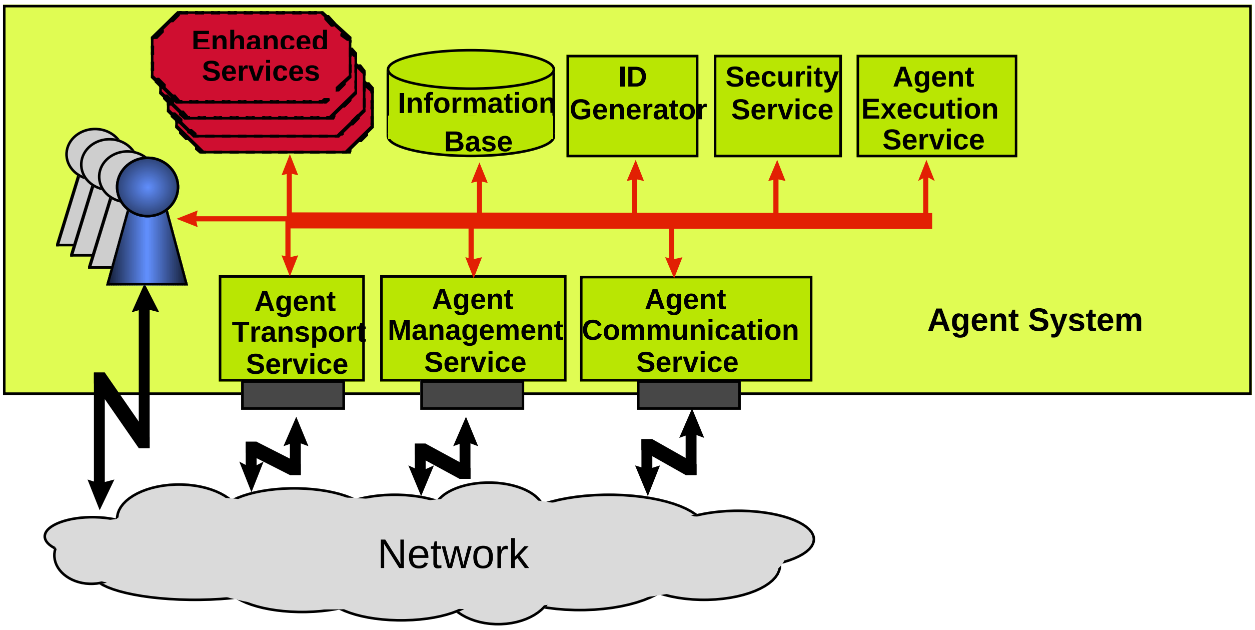
MA Benefits (cont.)

Flexibility: On-demand and customised service distribution

- **instant downloading of service clients from service providers to service users**
- **also servers may be extended or replaced dynamically**



Basic Agent Platform Services



Agent Execution Support: An agent platform must provide the capability to create mobile agents, taking into account agent-specific requirements regarding the runtime environment. Before the creation, the platform has to retrieve the agent's code, that may either be delivered during the creation request, or downloaded separately from an external code base.

Management Support: It is necessary for agent administrators to be able to monitor and control their agents. The control aspect comprises among others the temporary interruption of an agent's task execution, its premature termination, or the modification of its task. The monitoring of an agent is associated with its localization in the scope of the whole distributed environment. Regarding an agent system, all hosted agents as well as the occupied system resources have to be monitored and controlled by the system administrator.

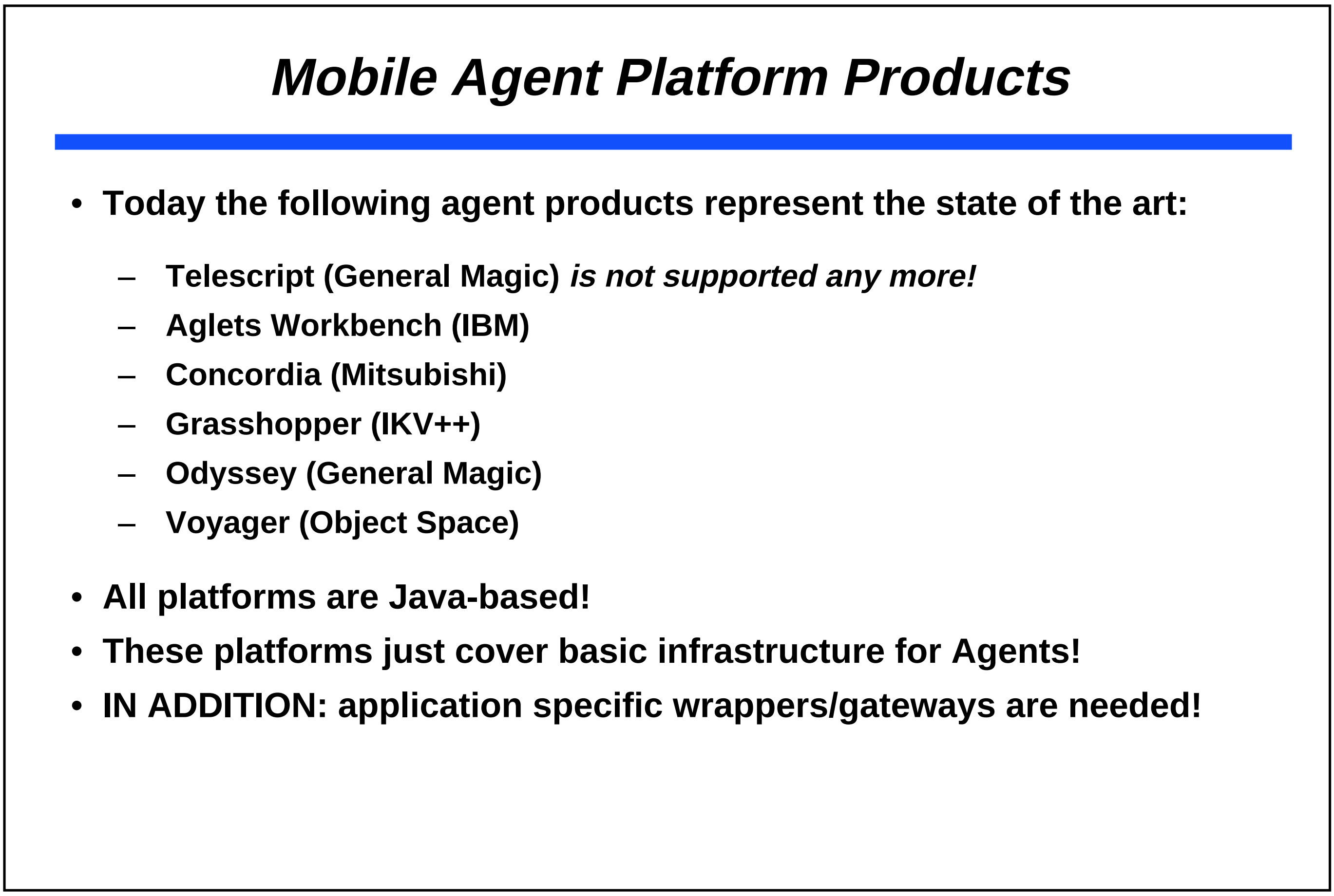
Security Support: The privacy and integrity of agents as well as agent systems must be guaranteed throughout their entire lifetimes. This requirement comprises the encryption and decryption of agents during migration, the authentication and authorization of agents and agent systems, and access control regarding the resources of an agent system or even of an agent offering some functionality.

Mobility Support: A special mobility support must be provided by the platform, supporting remote execution as well as migration. Note that the mobility aspect cannot be sufficiently handled without regarding the security support mentioned above.

Support for Unique Identification: Mobile agents as well as agent systems have to be uniquely identifiable in the scope of the entire agent environment. Thus, special support is required for the generation of unique agent identifiers.

Transaction Support: An important requirement is the support of correct and reliable execution of agents in presence of concurrency and occurrence of failures. Therefore, transaction support must be provided.

Communication Support: Agents should be able to communicate with each other as well as with platform services. Several mechanisms are possible, such as messages, method invocation or a blackboard mechanism. Communication through messages may be done point-to-point, by multicasting or broadcasting. Furthermore, agent communication includes support for semantic analysis.



Mobile Agent Platform Products

- **Today the following agent products represent the state of the art:**
  - **Telescript (General Magic) *is not supported any more!***
  - **Aglets Workbench (IBM)**
  - **Concordia (Mitsubishi)**
  - **Grasshopper (IKV++)**
  - **Odyssey (General Magic)**
  - **Voyager (Object Space)**
- **All platforms are Java-based!**
- **These platforms just cover basic infrastructure for Agents!**
- **IN ADDITION: application specific wrappers/gateways are needed!**

Telescript from General Magic: In 1991, General Magic provided one of the first operational agent environments, comprising an agent programming language (Telescript) and an execution platform. Despite the very good design and functionality, the lack of interoperability with other languages and platforms hindered the acceptance of Telescript.

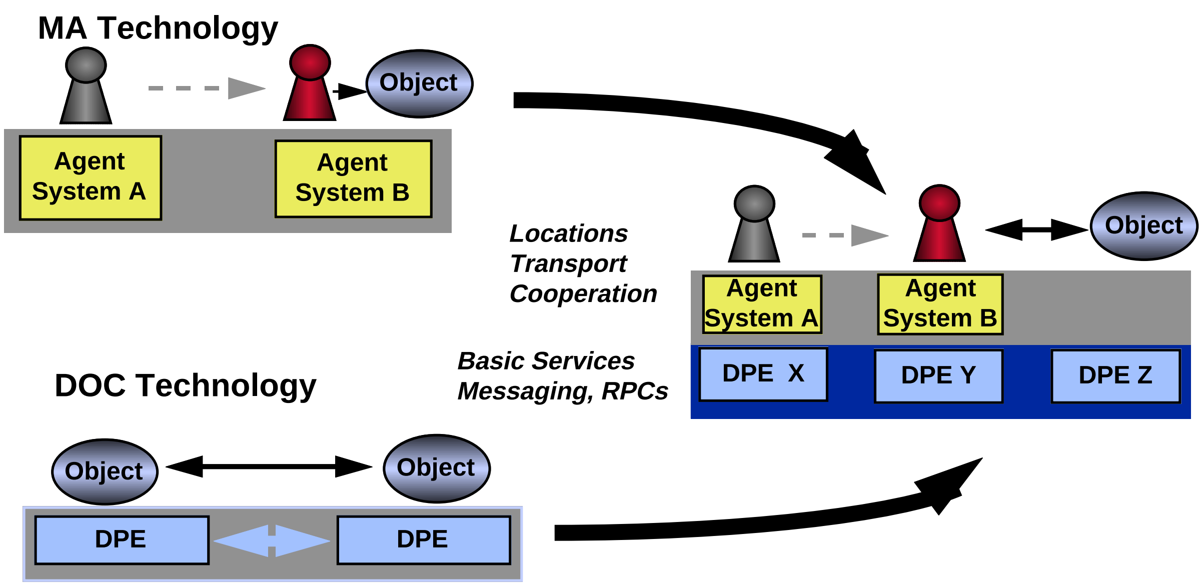
Odyssey from General Magic: After canceling the Telescript project, General Magic provided Odyssey which is a completely Java-based agent environment. Odyssey runs on any platform that supports JDK 1.1 or higher. Mobile agents are transferred via Java RMI, Microsoft's DCOM or CORBA IIOP. The Odyssey concepts are strongly related to Telescript. However, it is not possible to realize the entire Telescript functionality with pure Java. Odyssey will be enhanced in order to become compliant to the OMG MAF standard.

Aglets Workbench (AWB) from IBM: AWB is a Java-based environment for the construction and operation of mobile agents, consisting of development tools, libraries and ready-to-run agent system components. AWB runs on any platform that supports JDK 1.0.2 plus RMI preBeta2, JDK 1.1 or higher. For agent transport, AWB uses Java sockets and a special agent transfer protocol. The whole Workbench comprises the required Framework APIs, a visual agent manager, and an agent Web launcher. An enhancement of the AWB is planned in order to satisfy the OMG MAF standard .

Voyager from ObjectSpace: Voyager is a Java-based, agent-enhanced Object Request Broker (ORB), running on all platforms that support JDK 1.1. Voyager allows the development of network applications using both the traditional client/server paradigm and agent technology. Agents use the ORB for migration, and an integration of Microsoft's DCOM is planned. Besides, agent communication is supported by means of local and remote message transfer.

Grasshopper from IKV++: Grasshopper is an agent platform that is also entirely implemented in Java, thus running on platforms supporting JDK 1.1. Besides, it is built on top CORBA and thus, like Voyager, combines the client/server paradigm and mobile agent technology. Agents are transferred via CORBA IIOP or Java RMI. Apart from a distributed agent runtime environment, Grasshopper provides various „plug-in's" to enhance the core functionality of agents and agent systems. Grasshopper is developed compliant to the OMG MASIF (http://www.ikv.de/products/grasshopper.html).

Integration of DOC and MA Technology



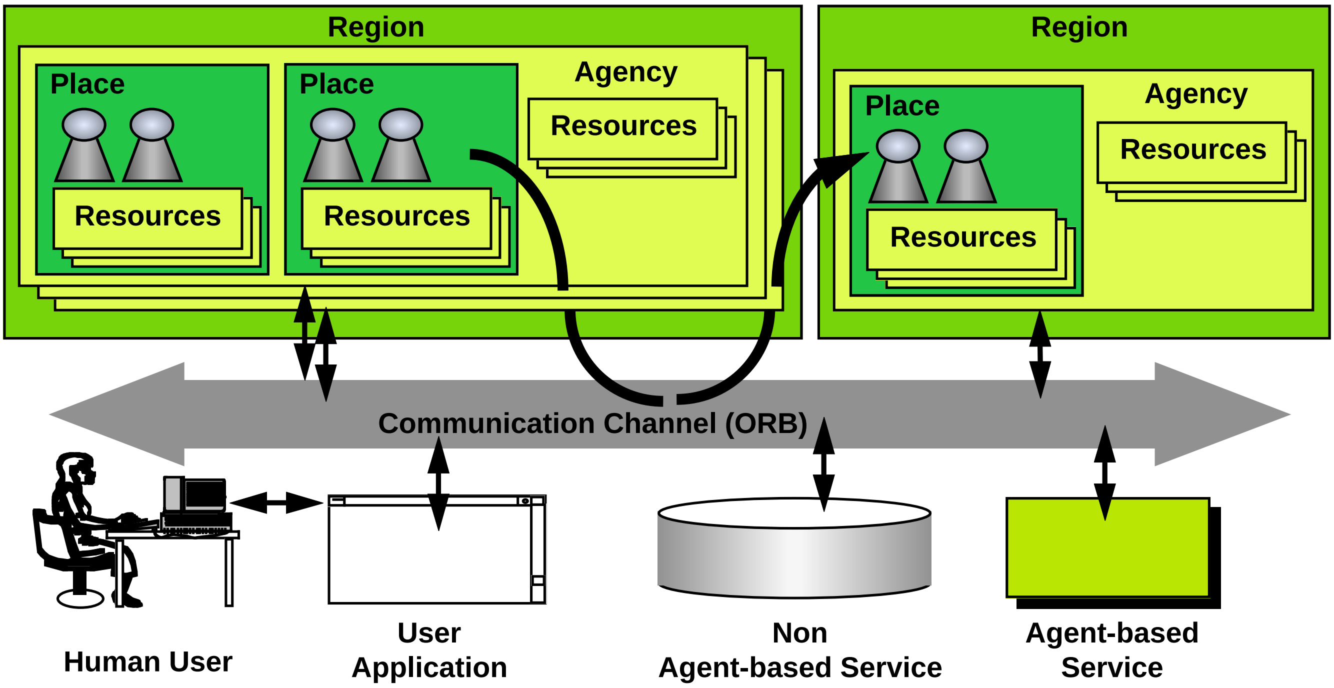
Today distributed object computing technology / middleware, such as OMG's Common Object Request Broker Architecture (CORBA), has gained considerable acceptance in the telecommunications systems. Particularly, TMN and IN are currently targeted for the introduction of CORBA technology, in order to achieve more flexibility in terms of software distribution, interoperability of heterogeneous platforms and particularly integration of legacy technologies.

On the other hand, mobile agent technology is currently gaining momentum in many domains (particularly in telecommunications), too. However, the mobile agent paradigm has been adopted by many people as some kind of religion, which prohibits somehow the usage of Remote Procedure Calls (RPCs) across a network for client/server communication.

Hence in the last years mobile agent technology had to be regarded as another technology incompatible with current distributed object technology. But in order to support various applications in a flexible and efficient way, sometimes RPCs are necessary or more sensible than shipping agents back and forth. On the other hand only mobile agent technology enables the distribution of „intelligence“ on demand.

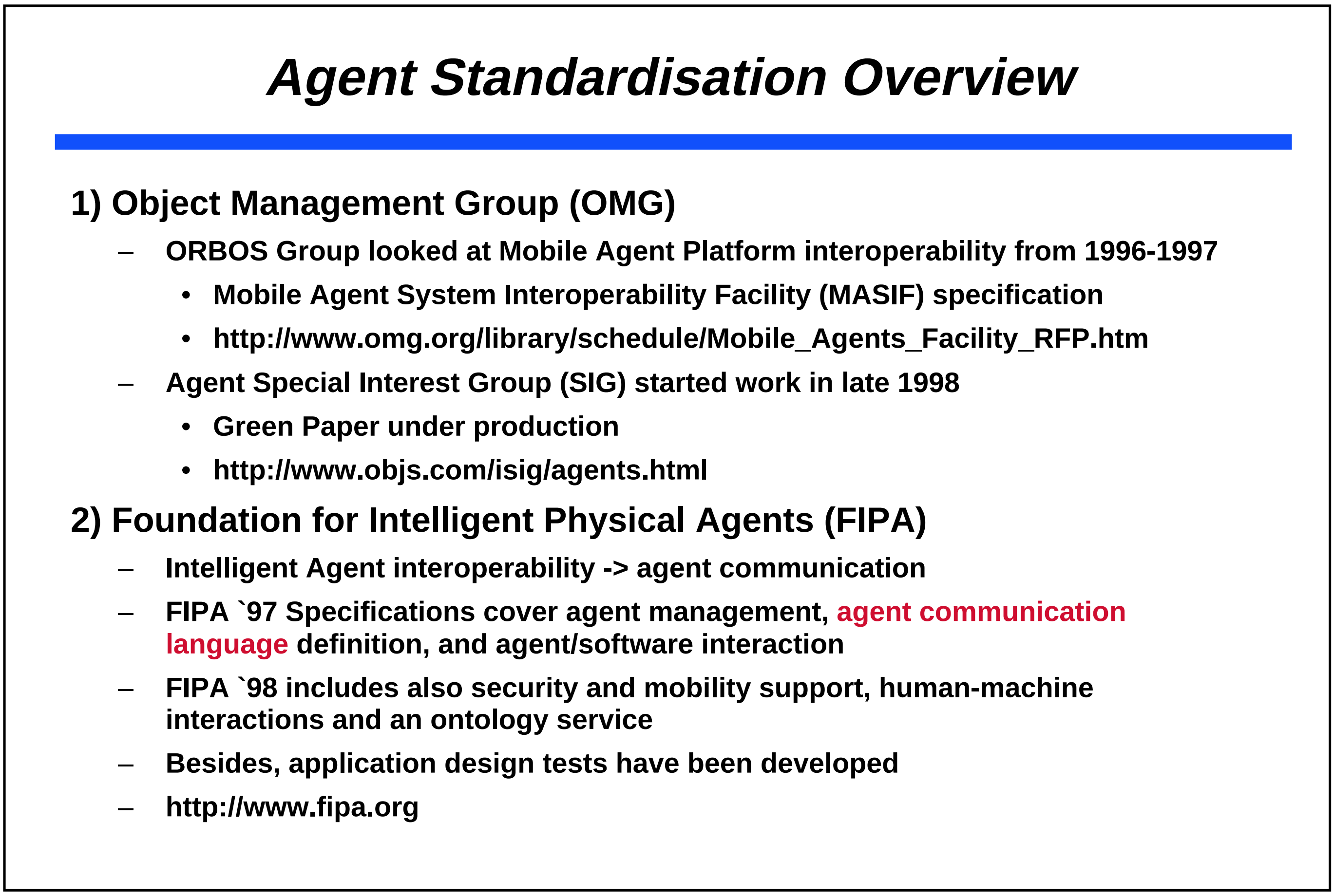
Fortunately there is today a common understanding that mobile agent technology should be seen as an enhancement of distributed object technology rather than a replacement. The reason is that both technologies offer advantages in different aspects, which only collectively provide the maximum flexibility in application design.

Integration of DOC and MA Technology (cont.)



This slide depicts a general structure of a CORBA-based MA platform. The actual runtime environments for mobile agents are called agencies. Each agency consists of one or more places that provide specific resources (e.g. services, memory, file system access, etc.). This concept has been proved to be valuable by several existing platforms, in the first place Telescript. Agencies may be grouped to regions in order to facilitate management operations. E.g. a region can be associated to a single authority, providing certain security policies for each comprised agency. The agent transport as well as further interactions between distinguished DAE and non DAE components are performed via CORBA. In this way, legacy services, like e.g. a CORBA trading, naming, or event service can be used to enhance the platform functionality in a very comfortable manner.

The core agency comprises only those capabilities that are inevitably necessary for the maintenance of the agency itself, as well as for the execution and migration of mobile agents. Additional, application-dependent functionality is realized by modular and reusable building blocks that can easily be plugged into a core agency. Examples of building blocks are adapter services for the access of telecommunication hardware or sophisticated communication services. In this way, any requirements regarding desired functionality or hardware restrictions can be met.



Agent Standardisation Overview

1) Object Management Group (OMG)

- ORBOS Group looked at Mobile Agent Platform interoperability from 1996-1997
  - Mobile Agent System Interoperability Facility (MASIF) specification
  - http://www.omg.org/library/schedule/Mobile_Agents_Facility_RFP.htm
- Agent Special Interest Group (SIG) started work in late 1998
  - Green Paper under production
  - http://www.objs.com/isig/agents.html

2) Foundation for Intelligent Physical Agents (FIPA)

- Intelligent Agent interoperability -> agent communication
- FIPA `97 Specifications cover agent management, agent communication language definition, and agent/software interaction
- FIPA `98 includes also security and mobility support, human-machine interactions and an ontology service
- Besides, application design tests have been developed
- http://www.fipa.org

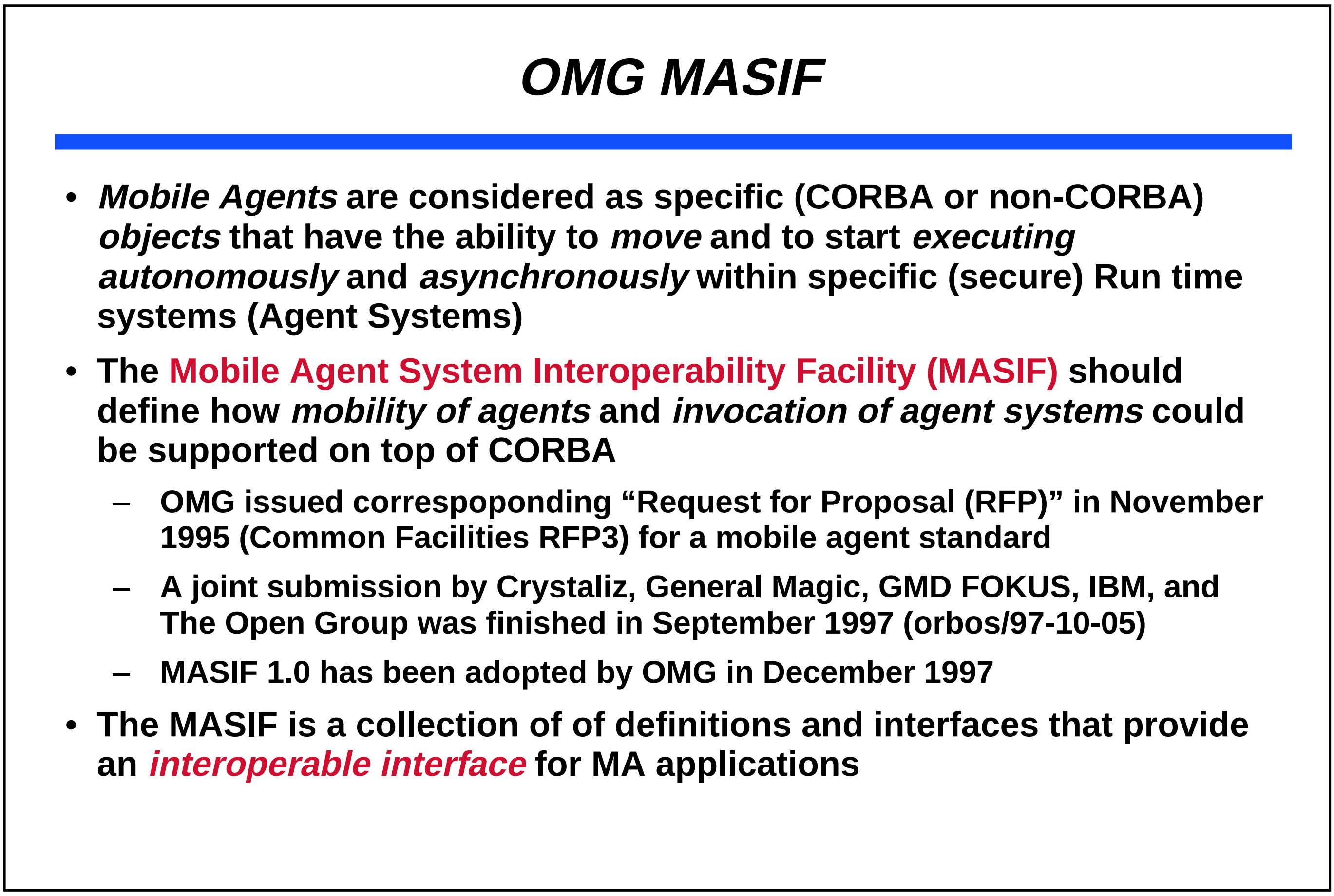
Agent Standardization Activities

1) Object Management Group (OMG)

According to an OMG request for proposal, Crystaliz, General Magic, GMD FOKUS, IBM, and The Open Group are developing the Mobile Agent System Interoprability Facility (MASIF) as a standard for mobile agent technology. The standard comprises, among others, CORBA IDL specifications supporting agent transport and management, including localization capabilities. The objective is to enable interoperability between agent platforms of different manufacturers. For the details see: http://www.omg.org/library/schedule/Mobile_Agents_Facility_RFP.htm. In autumn 1998, an Agent Special Interest Group (SIG) has been established. See for details: http://www.objs.com/isig/home.htm

2) Foundation for Intelligent Physical Agents (FIPA)

FIPA has developed in 1997 a seven-part specification called FIPA 97. The three main specification documents are focusing on agent management, defining an agent communication language, and dealing with agent/software interaction. Besides, four application design tests have been developed in order to validate these specifications. These comprise Personal travel assistance, Personal assistant, Audio-visual entertainment and broadcasting, and Network management and provisioning. In 1998 FIPA started work on an enhanced set of specfications, called FIPA`98. In addition to modifications and extensions of FIPA'97 specifications, FIPA'98 also contains six new parts. These are: Human Agent Interaction, Product Design and Manufacturing Agents, Agent Security Management, Agent Management Support for Mobility, Ontology Service, and a FIPA97 Developers Guide. FIPA URL is: http://www.fipa.org/



OMG MASIF

- ***Mobile Agents*** **are considered as specific (CORBA or non-CORBA)** ***objects*** **that have the ability to** ***move*** **and to start** ***executing autonomously*** **and** ***asynchronously*** **within specific (secure) Run time systems (Agent Systems)**
- **The Mobile Agent System Interoperability Facility (MASIF) should define how** ***mobility of agents*** **and** ***invocation of agent systems*** **could be supported on top of CORBA**
  - **OMG issued correspoponding "Request for Proposal (RFP)" in November 1995 (Common Facilities RFP3) for a mobile agent standard**
  - **A joint submission by Crystaliz, General Magic, GMD FOKUS, IBM, and The Open Group was finished in September 1997 (orbos/97-10-05)**
  - **MASIF 1.0 has been adopted by OMG in December 1997**
- **The MASIF is a collection of of definitions and interfaces that provide an** ***interoperable interface*** **for MA applications**

The Common Object Request Broker Architecture (CORBA) has been established as an important standard, enhancing the original RPC based architectures by allowing relatively free and transparent distribution of service functionality. Besides, mobile agent technology has been proved to be suitable for the improvement of today's distributed systems. Due to its benefits, such as dynamic, on-demand provision and distribution of services, reduction of network traffic and the reduction of dependence regarding server failures, various problems and inefficiencies of today's client/server architectures can be handled by means of this new paradigm. However, for several applications RPCs still represent a powerful and efficient solution. Thus, an integrated approach is desirable, combining the benefits of both client/server and mobile agent technology, and on the other hand eliminating or at least minimizing the problems that rise if one of these techniques is used as „stand-alone" solution. This slide shows this integrated approach by means of a distributed agent environment (DAE) that is built on top of a CORBA distributed processing environment (DPE).

Today, the CORBA standards gain high acceptance in the world of distributed computing. Various service specifications have been developed, providing standardized, implementation-independent interfaces and a common protocol, and thus enabling a high degree of interoperability between applications of distinguished manufacturers. In contrast to this, mobile agent technology is driven by a huge variety of different approaches regarding implementation languages, protocols, platform architectures and functionality. In order to achieve a sufficient integration with CORBA, a standard is required for mobile agent technology. This standard has to handle interoperability between distinguished agent platforms, and the usability of (legacy) CORBA services by agent-based components.

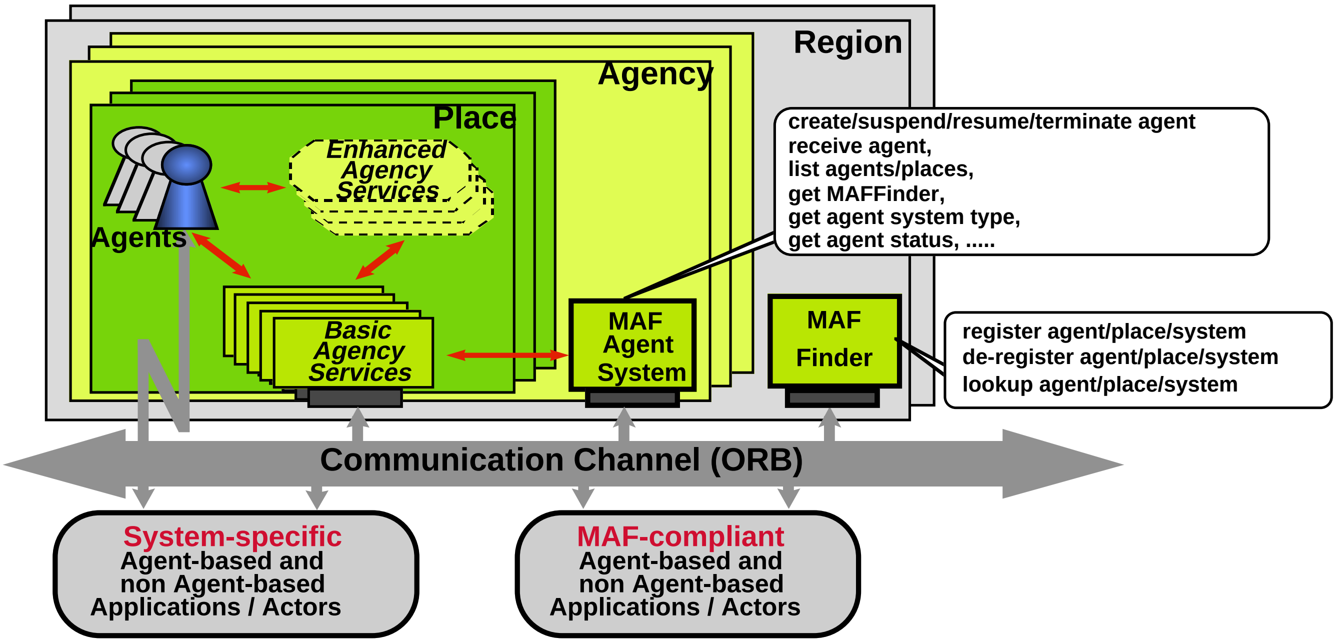
Therefore OMG issued a Request for Proposal (Common Facilities RFP3) for a mobile agent facility in November 1995. After the failure of former proposals, the Mobile Agent System Interoperability Facility (MASIF) submission, written by Crystaliz, General Magic, GMD FOKUS, IBM, and The Open Group, has been sent to the OMG for voting in November 1997. The OMG adopted the submission in spring 1998. Availability of MASIF-compliant agent platforms are expected for end of 1998.

The MASIF (formerly MAF) RFP is available through ftp://ftp.omg.org/pub/docs/1995/95-11-03.rtf.

The submitted MASIF specification is available through "ftp://ftp.omg.org/pub/docs/orbos/97-10-05.pdf".

For the MASIF progress look at "http://www.omg.org/library/schedule/Mobile_Agents_Facility_RFP.htm"

Basic Interfaces of the OMG MASIF


Region
Agency
Place
Agents
Enhanced Agency Services
Basic Agency Services
MAF Agent System
MAF Finder
Communication Channel (ORB)
System-specific
Agent-based and non Agent-based Applications / Actors
MAF-compliant
Agent-based and non Agent-based Applications / Actors

create/suspend/resume/terminate agent
receive agent,
list agents/places,
get MAFFinder,
get agent system type,
get agent status, ...

register agent/place/system
de-register agent/place/system
lookup agent/place/system

The idea behind the MASIF standard is to achieve a certain degree of interoperability between mobile agent platforms of different manufacturers without enforcing radical platform modifications. MASIF is not intended to build the basis for any new mobile agent platform. Instead, the provided specifications shall be used as an „add-on“ to already existing systems.

As shown in above, MASIF has adopted the concepts of places and regions that are used by various existing agent platforms, such as Telescript. A place groups the functionality within an agency, encapsulating certain capabilities and restrictions for visiting agents. A region facilitates the platform management by specifying sets of agencies that belong to a single authority.

Two interfaces are specified by the MASIF standard:

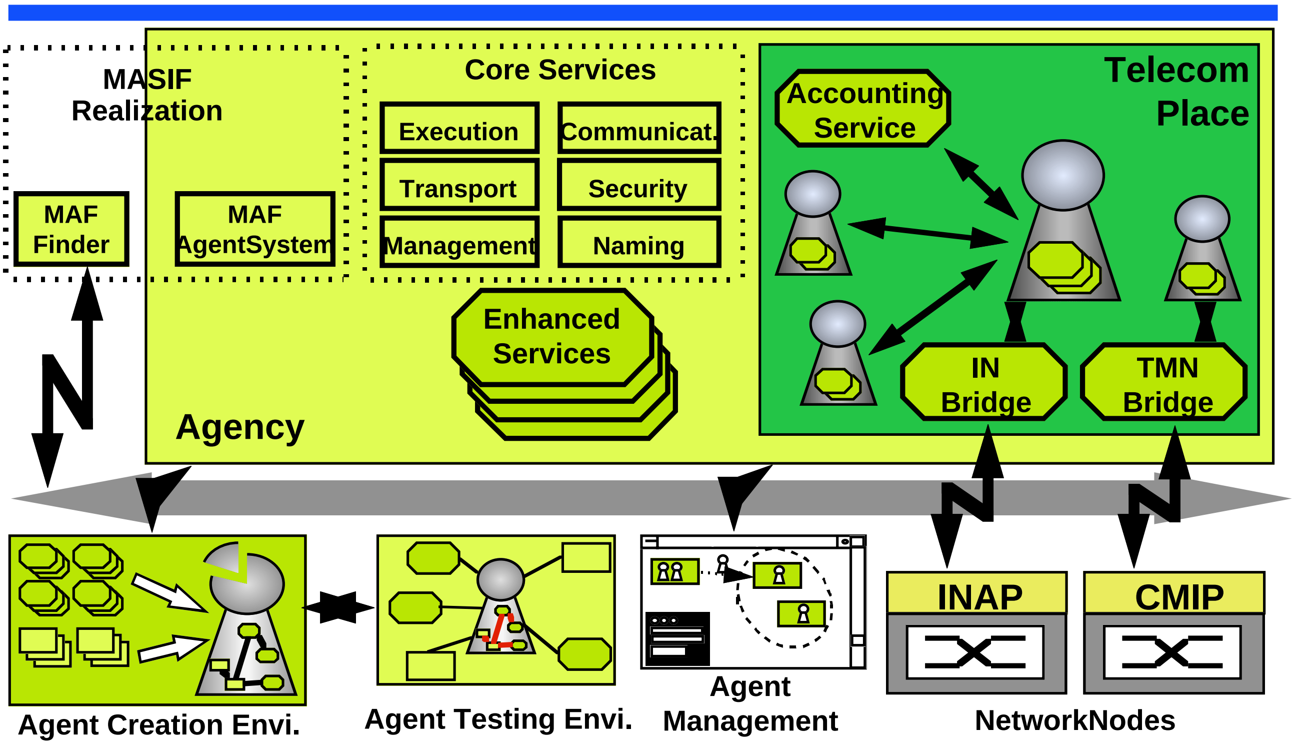
• the MAFAgentSystem interface provides operations for the management and transfer of agents, whereas

• the MAFFinder interface supports the localization of agents, agent systems, and places in the scope of a region or the whole environment, respectively.

As part of a MASIF-compliant agency, a MAFAgentSystem object interacts internally with agency-specific services and provides the associated CORBA interface to external users. In this way, it is possible to communicate with an agency either in a MASIF-compliant way (using the MAFAgentSystem interface) or in a platform-specific way (using platform-specific interfaces that may provide additional functionality, not handled by MASIF).

Apart from the agent-specific CORBA interfaces MAFAgentSystem and MAFFinder, the MASIF standard explains in detail how existing CORBA services, like e.g. the Naming, Life Cycle, Externalization, and Security Service, can be used by agent-based components to enhance the provided functionality. Note that the current MASIF submission only represents the first approach for a mobile agent standard. Enhancements are planned when sufficient experiences have been made with the implementation of the provided specifications.

MASIF conform MA Platform



Based on the previous considerations a generic mobile agent platform will look quite similar as the one depicted above. Three main objectives are associated with such a platform:

1) In order to support both the traditional client/server paradigm and mobile agent technology, the agent platform has to be built on top of RPC-based middleware. Regarding its high acceptance, CORBA should be chosen for this task.

2) To support various application areas with distinguished needs regarding the platform functionality, a separation is necessary between the core functionality of the platform and additional capabilities. The additional capabilities have to be designed by means of modular, reusable functional building blocks which can be „plugged“ into the platform, enhancing its core.

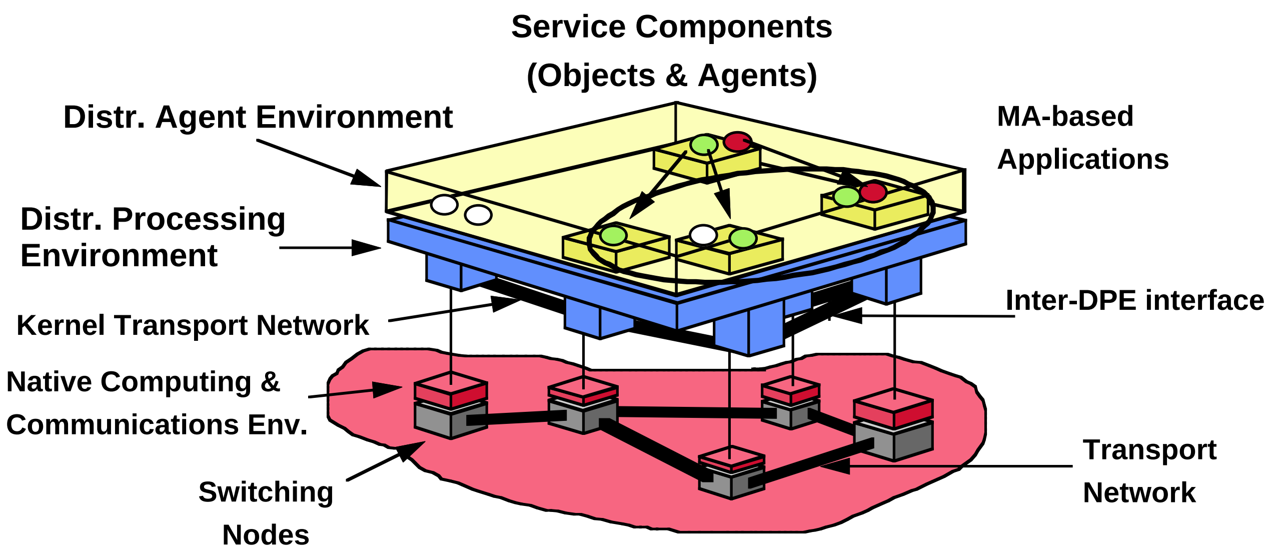
3) In order to take advantage of interoperability with further agent platforms of different manufacturers, compliance to the OMG MASIF standard has to be achieved.

Note that the platform architecture is similar to the one shown before. Special emphasis lies on the opportunity to easily enhance the platform capabilities in order to fulfill individual needs, depending on concrete application scenarios. To achieve this goal, the core agency comprises only those capabilities that are inevitably necessary for the maintenance of the agency itself, as well as for the execution and migration of mobile agents. Additional, application-dependent functionality is realized by modular and reusable building blocks that can easily be plugged into a core agency. Examples of such building blocks are adapter services for the access of telecommunication hardware or sophisticated communication services. In this way, any requirements regarding desired functionality or hardware restrictions can be met.

Apart from the agencies themselves, demands for various tools have to be provided. An agent creation environment enables the „plug and play“ composition of mobile agents out of reusable functional building blocks. In this way, new agent-based services can be created and provided very dynamically and on-demand. Similar to the agency, an agent is divided into a core and an application-specific part. An agent testing environment allows for the simulation of the whole distributed environment by means of a single agency. The entire execution of an agent, including its migration and access of sensible software and hardware, can be simulated „locally“ without endangering the real resources. Finally, a graphical agent management tool enables the monitoring and control of agents and agencies in the scope of one or more regions.

Target Flexible Telecom Middleware

Integration of Distributed Object and Mobile Agent Technologies



The agent transport as well as interactions between different Distributed Agent Enviroment (DAE) and non DAE components are performed via CORBA. In this way, existing services, e.g. a CORBA trading, naming, or event service can be used to enhance the platform functionality in a very comfortable manner. By providing the CORBA interfaces specified by the OMG MASIF standard, interoperability to other agent platforms can be achieved.

Mobile Agents for Telecommunications

MAs provide new opportunities for service control & management

- **reduction of traffic load and availability requirements of the underlying networks and systems**
  - **via the autonomy and asynchronous operations of the agents**
- **reduction of customer intelligence required during installation, operation and maintenance of resources**
  - **via the intelligence and autonomy of the agents**
- **enable “on demand” provision of customised services**
  - **via dynamic agent downloading from the provider system to the customer system and further on back to the provider system or directly to the resources**
- **enable more decentralised realisation of service control and management software**
  - **by means of bringing the control or management agent as close as possible to the resource(s)**

The heterogeneity, distribution, scale and dynamic nature of the emerging open Telecommunication environments, sometimes referred to as “information infrastructure”, is calling for new paradigms for the control and management of the open resources. In this context, the agent-based technology seems to offer a promising solution to cope with the complexity of this environment.

Within such an open environment, an agent-based solution can:

• reduce the requirement of traffic load and the availability on the underlying networks (via the autonomy and asynchronous operations of the agents);

• reduce the requirement on customer intelligence during the installation, operation and maintenance of the resources (via the intelligence and autonomy of the agents);

• enable „on demand“ provision of customized services (via dymanic agent downloading from the provider system to the customer system and further on back to the provider system or directly to the resources);

• increase the flexibility, reusability and effectiveness of the software-based problem solutions;

• allows for a more decentralized realisation of service control and management software, by means of bringing the control or management agent as close as possible or even onto the resources.

Toward an MA-Based IN Environment

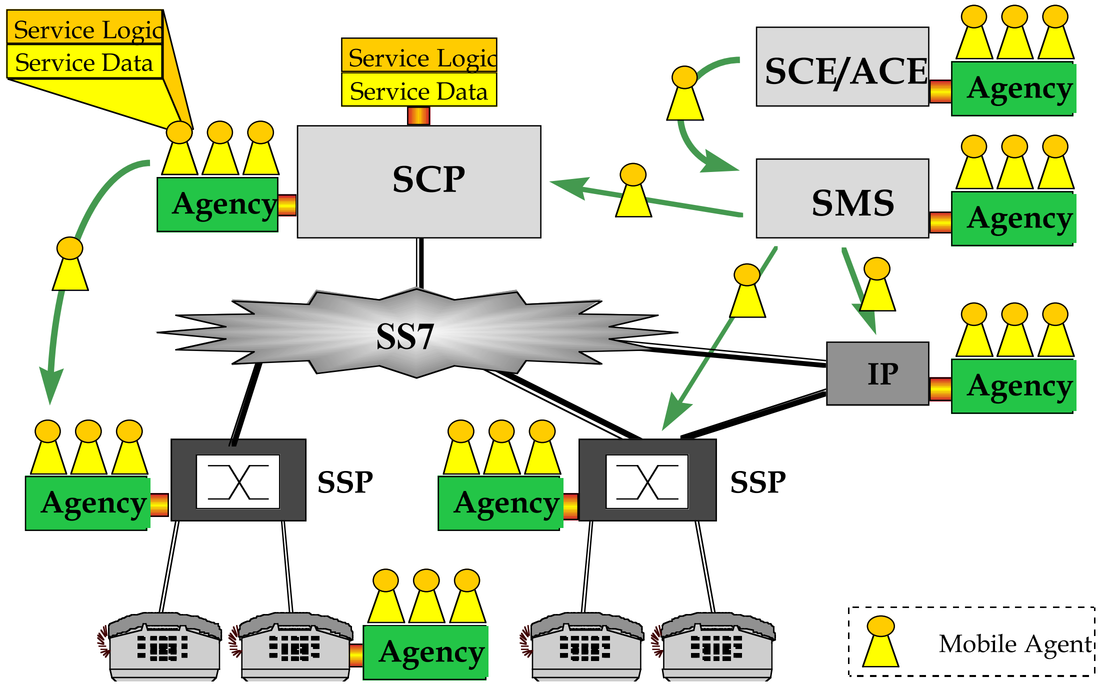
- **Open Service Market requires flexible service infrastructure**
- **IN architecture is limited in terms of flexibility and may become overloaded due to the centralized approach of service provision**
- **Introduction of distributed object technology, such as CORBA, is considered currently in the IN to increase flexibility**
- **Introduction of standard mobile agent technology into the IN environment would increase flexibilty even more!**
- **IN services are realized by means of mobile agents, the service provision can be improved in several ways:**
  - **Services can be provided time-dependent, i.e. installed for a limited time duration.**
  - **Distributed provision of services or service components enables load balancing in the network.**
  - **Finally, the management can be facilitated by automating the service provision.**

The IN architecture provides several important advantages, like e.g. the opportunity to create or modify network capabilities without any changes at the switches. However, due to the rising number of service users and the increasing number of provided IN services, the centralized SCPs are likely to become the bottleneck of the whole system. Even now the SCP capacity is temporarily overdrawn. Besides, the deployment and subscription of services by the SMS is not efficient and open enough to handle the demands of the emerging open service market. Finally, due to the centralized architecture, a server (i.e. SCP) failure would cause immense costs for the service providers.

Based on these limitations, the introduction of distributed object technology, such as CORBA, is considered currently in the IN world. Taking this into account, we propose the introduction of the previously described standard mobile agent technology into the IN environment in order to achieve ultimate flexibility. This means that IN services could be implemented by either CORBA objects and/or by mobile agents on top of CORBA. However, we focus on the last issue.

By realizing IN services by means of mobile agents, the service provision can be improved in several ways: Services can be provided time-dependent, i.e. installed for a limited time duration. Distributed provision of services or service components enables load balancing in the network. Finally, the management can be facilitated by automating the service provision, i.e. by dynamically installing and maintaining mobile service agents on those network nodes where they are needed.

MA-Based IN Environment



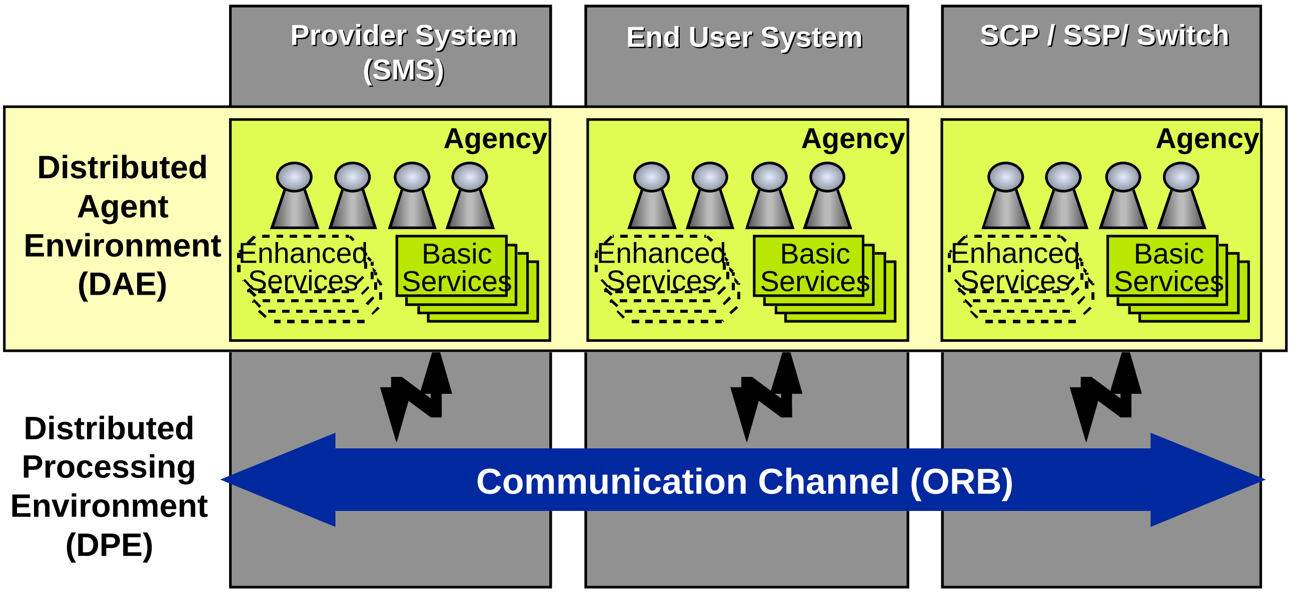
Within an MA-based IN envrionment the approach is to introduce services on demand and ultimately distribute service logic and data from the centralized SCP to switches and end user devices by means of mobile agents. To support agent technology, the distinguished network nodes are containing agencies (in line with those outlined before). In this way, agents representing IN service logic and data can be sent dynamically to those network locations where the functionality is currently required.

An agent-enhanced service creation environment (SCE) allows to develop appropriate IN service logic and data, including the envisaged itinerary of the agent.

An agent-enhanced service management system (SMS) allows to introduce the service agents into the MA-based IN infrastructure. It is a key component in respect to providing the service to potential customers and configuring service agents accordingly. After agent release the SMS keeps track of an agents location and it status.

The outlined integrated approach, combining both agent technology and client/server technology, allows to enhance the current IN architecture instead of completely removing it. Note that the agent transport shown above is not performed via the signaling system no.7 (SS7), but instead via a data network that interconnects the agencies.

Distributed Agent Environment within IN



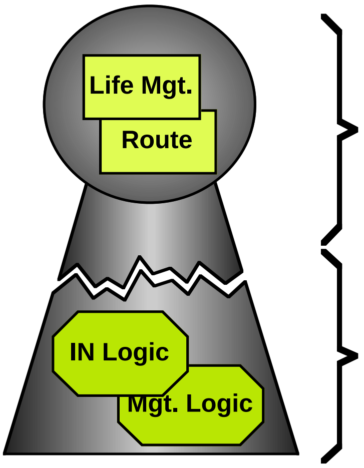
This slide depicts an agent oriented view of the IN environment. Basically each IN element has to provide an agency, which collectively form a distributed agent environment on top of a distributed processing environment.

Thus service agents can easily migrate in this “harmonized” environment from the provider system (i.e. the SMS) to the customer or end user system (for further service agent customisation) and furtheron to the best suited network location, which may be (depending on the telecommunications service to be offered) a local switch/SSP or a more central SCP.

In the following we look in more detail at the service provision process.

The two Faces of a Mobile Agent

Distinction between Agent Support Part and Application Part



Agent Functionality:
- **Life Cycle Control Functions**
- **Management GUI**
- **Mobility (e.g. Itinerary)**
- **Security**

Application Functionality:
- **Task Control (Service Logic/Data)**
- **Application-specific GUIs**
- **Use of Resource Interfaces (e.g. INAP, CMIP)**

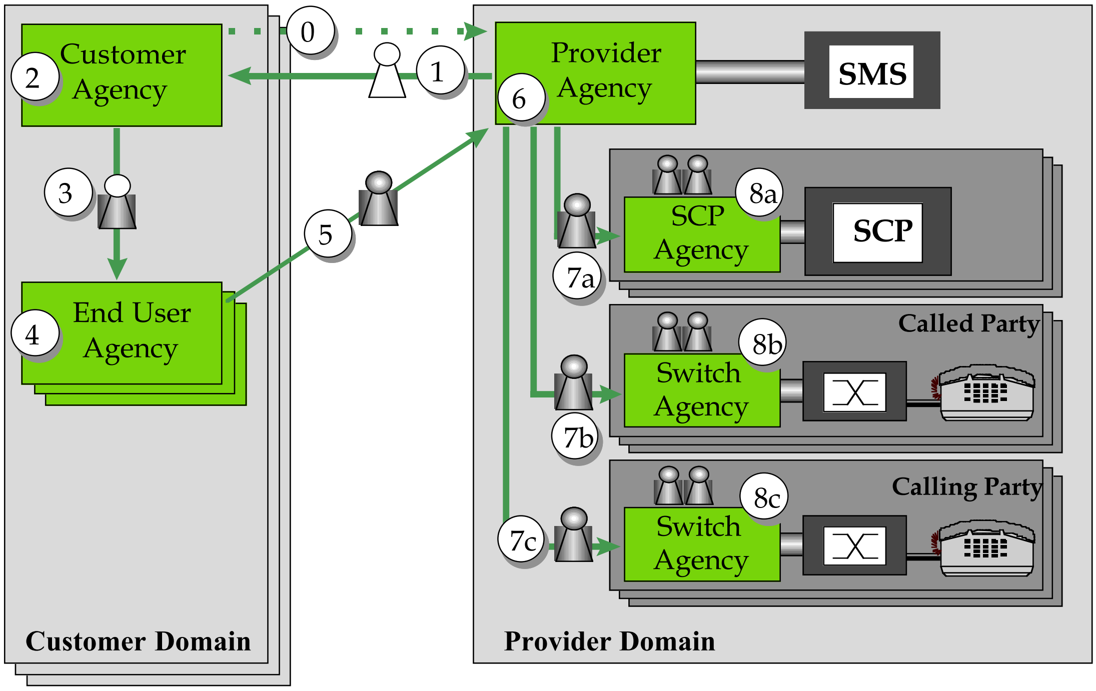
In this approach it is important to note that the a Service Agent comprises in principle two different parts:

The first part - the application part - is related to the provision of the real service, i.e. the control or management of switches. Therefore the agent contains appropriate (e.g. IN) logic and data and makes use of external interfaces (e.g. INAP interface at the switch or SCP) or adapted interfaces offered within the agency (e.g. a CORBA object which maps INAP operations into CORBA object invocations). In the first case traditional IN logic may be used, whereas in the second case advanced OO-based logic may be used. Note that this functionality may include also functionalities for service subscription, customisation and service logic maintenance (including appropriate GUIs).

The second part - the (core) agent part - is devoted to the agent's nature of being an autonomous mobile entity. This means that this part comprise all functionalities required for agent lifecycle management and mobility support (e.g. itinerary maintenance). These functions are realized by strongly interworking with the agency services.

It has to be stressed that these two parts are totally separated, since relationships exist between the agent's mobility and the provision of service subscription, customisation and real service control/management functionalities.

Mobile Agent-based Service Provision



Three kinds of actors are involved in the scenario: An agent provider who is accessible via the WWW, a customer representing a company or organization, and various end users. Each actor requires access to an agency. Additional agencies are connected to the distinguished network elements, i.e. SCP and switches.

The scenario is initiated by the customer accessing the provider via the WWW and requesting for an IN service agent (0).

The provider sends the requested agent to the customer (1) who is now able to pre-configure it (2).

The pre-configuration comprises the selection of end users who shall be able to use the agent, and the specification of various access restrictions concerning the selected end users. Afterwards, the service agent is sent to the end users (3) where it is supplied with individual service logic configurations (4).

Before the agent executes its designed task, it automatically migrates back to the provider (5) in order to allow security checks, e.g. the determination of code modifications (6).

Only if the security checks have been successful, the agent moves to a specific network node. Three possibilities are taken into account:

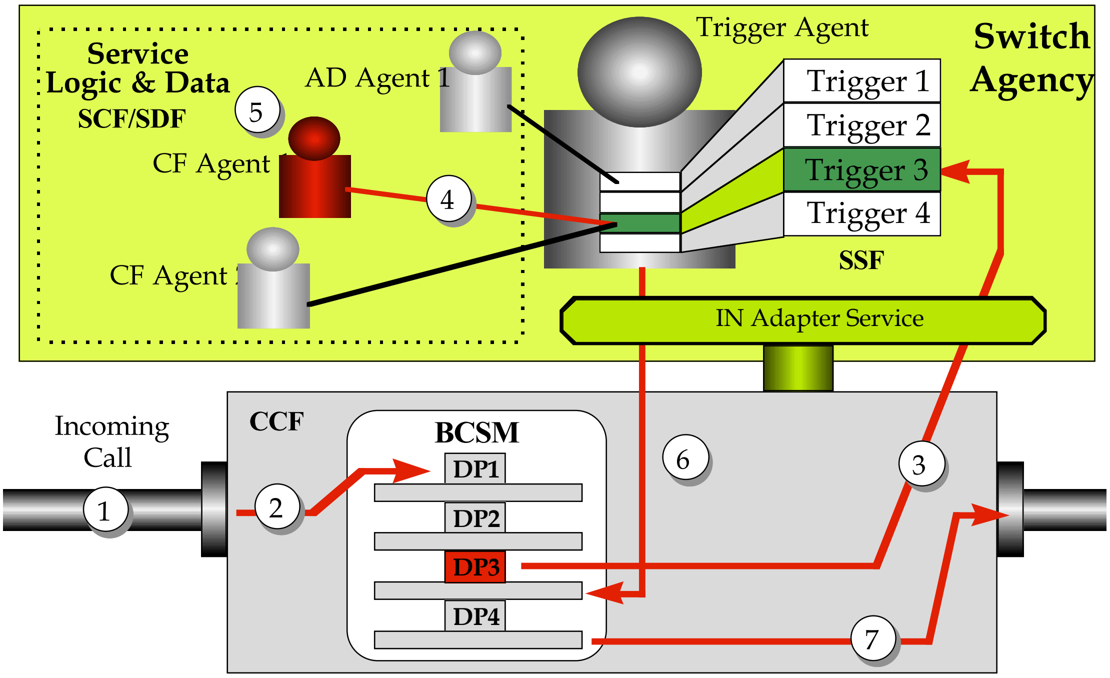
• Agents representing a global service, like e.g. freephone or premium rate, migrate to the agency connected to the centralized SCP (7a).

• Agents realizing a called party service, like e.g. call forwarding or originating call screening, move to the agency at the called party switch (7b).

• Agents representing a calling party service, like e.g. abbreviated dialing or termination call screening, move to the agency at the calling party switch (7c).

After reaching their destination agency, the agents connect themselves to specific IN service adapters which can either be realized by enhanced agency services or in turn by special (stationary) agents. Finally, the service execution starts (8a-8c).

Switch Agency (Detailed View)

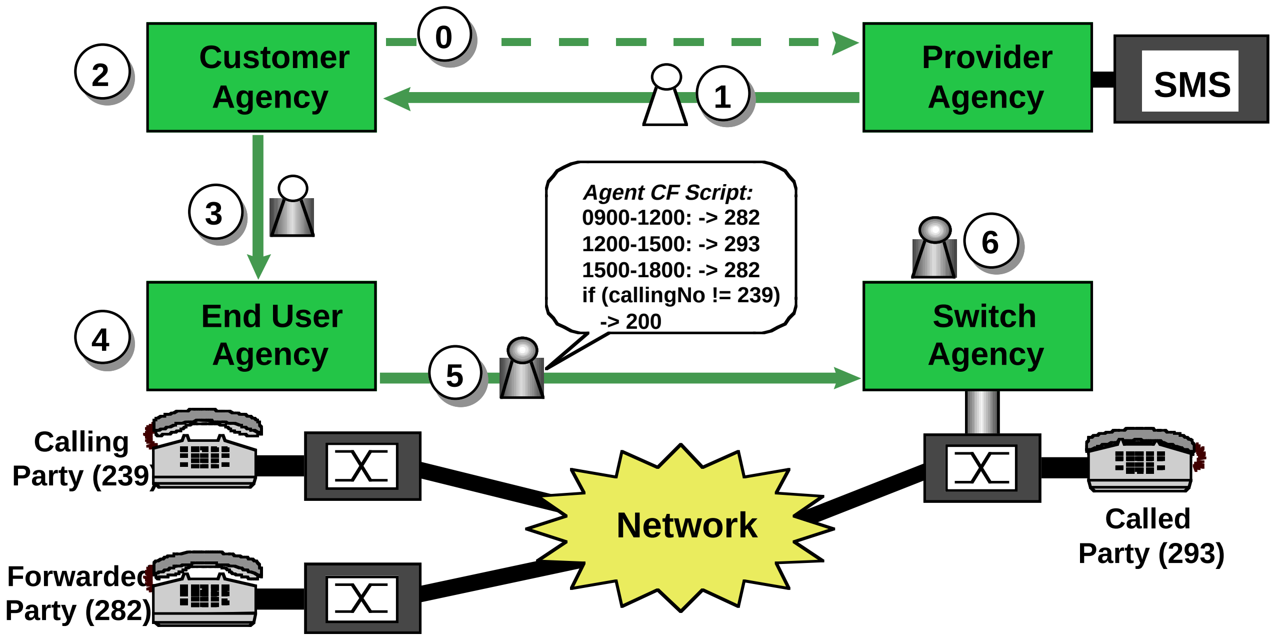


In order to relieve the centralized Service Control Points from network traffic, the basic call processing has to be performed without contacting the SCP. This can be achieved by mobile agents which represent the required service logic and data, and which reside on agencies connected to switching facilities. Thus we focus in the following on an „agent-based service node“ implementation. This slide depicts an approach of an agent-based call processing.

An enhanced agency service (named „IN adapter service“ above) represents the bridge between the switch and the agent environment. The trigger table that has formerly been maintained by the switch itself, is now provided by the stationary Service Trigger Agent (STA). The STA in turn is connected to the mobile IN service agents which comprise the actual service logic/data that has formerly been stored within the SCP. In this way, the switch agency hosts the SSF, SCF and SDF.

The service agents can be sent dynamically on-demand to those switches where they are currently required. After their arrival, they connect themselves to the STA, delivering the necessary trigger information. During the call processing, when a detection point is reached, the processing is suspended, the Service Trigger Agent is contacted and determines if there is any connected service agent responsible for the occurred event. If this is true, the service agent is accessed and performs the associated maintained logic program. After returning the result, the call processing is resumed.

MA-based Call Forwarding Service



The agent-based call forwarding service enables end users to initiate an automatic routing of their own telephone number to another device, dependent on the time of day, the calling party number, or specific events .

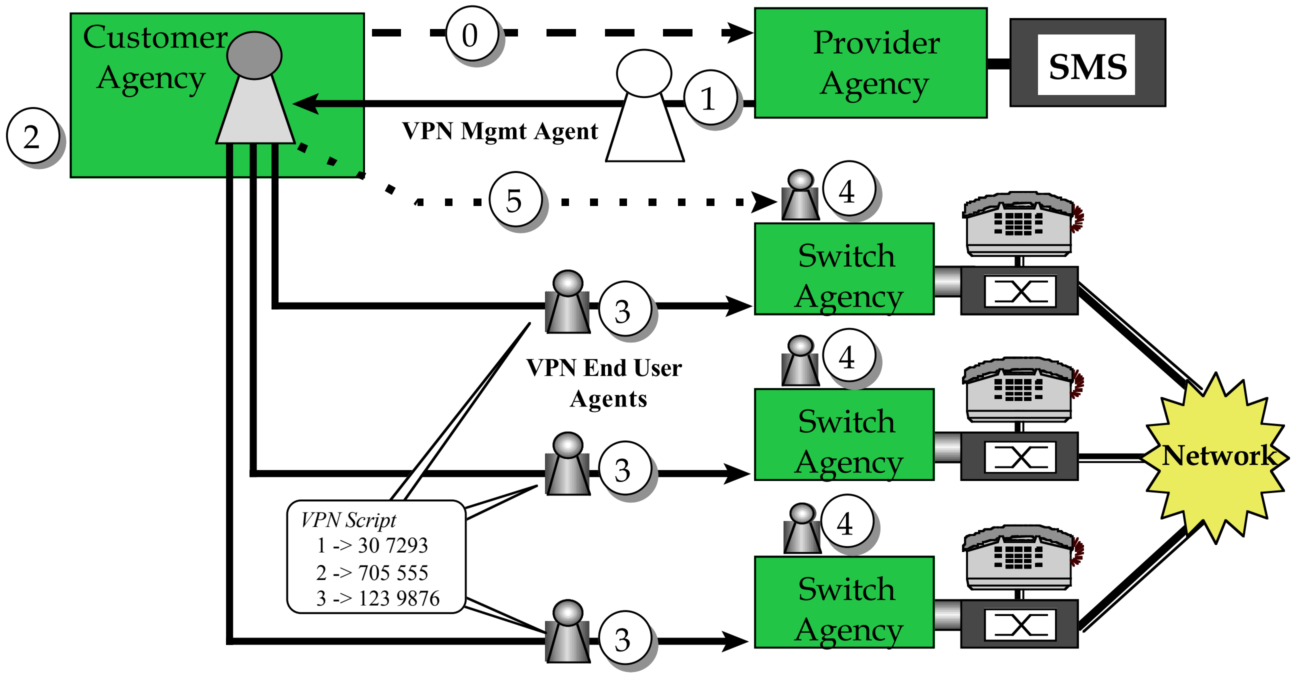
The pre-configuration phase at the customer agency (step 2) comprises the specification of the end users who shall be allowed to use the CF agent. Relevant user information contains the name and phone number as well as the address of the end user's agencies.

Additionally, several restrictions can be specified for each user: By activating the call screening mechanism of the agent, the area to which the phone number may be forwarded can be limited to the scope of a single town or country. Besides, the entire life time of the agent can be set. Note that further configuration aspects can easily be added.

In the final configuration phase that is performed by the end user(s) (step 4), the customization of the call forwarding logic is performed. The routing can be time-dependent or event-dependent, i.e. for example dependent on the calling party's number. The figure shows a simple example of the forwarding logic script: from 0900am to 1200am calls are forwarded to the number 282 (time-dependent routing); if the participant with the number 239 calls, the call is forwarded to the number 200 (event-dependent routing). Note that there are no restrictions regarding the complexity of the possible service logic. All states and events that are associated to detection points of the switch's basic call state model can be used by the IN service agents.

Note that the progress of this scenario is similar to the scenario overview described in the previous slide. In contrast to the general approach, the configured service agents do not move back to the provider for security purposes before executing their task on the switch agency. This intermediate step has been skipped since this first version of the scenario has been developed just to validate the idea and feasibility of an agent-based IN. However, this step is required in a „real world application“.

MA-based Virtual Private Network Service



VPN Script
1 -> 30 7293
2 -> 705 555
3 -> 123 9876

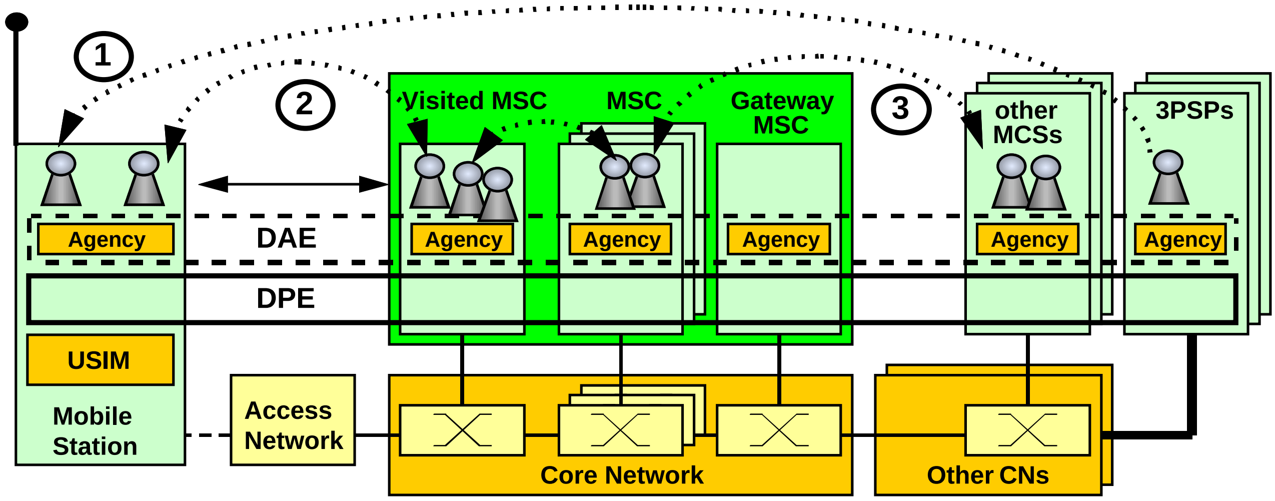
The IN service virtual private network (VPN) provides private network capabilities by using public network resources. The customer's lines, connected to different network switches, constitute the VPN. This slide shows how even this sophisticated service can be realized by mobile agent technology.

In order to establish the VPN, the customer retrieves a VPN management agent from a provider (step 0/1). The configuration of this management agent (step 2) comprises the definition of a private numbering plan, the specification of the locations and addresses of the desired VPN participants, and if desired, additional settings like e.g. special charging or call forwarding mechanisms. After the configuration, the management agent creates several VPN end user agents according to the specified participants (i.e. end users), supplies these agents with the required configuration data, and sends them to the switches that are associated to the participants' end user devices (step 3). The end user agents connect themselves to the hosted trigger agents (cf. Figure 8) and in this way automatically establish the virtual private network (step 4).

Apart from initiating the establishment of the VPN, the management agent maintains control over the network during its entire life time, allowing the customer to dynamically modify any configuration settings, or even to increase or reduce the number of participants by creating new end user agents or terminating existing ones. Changes of the numbering plan or any other information that is maintained by the end user agents can be modified by the management agent via RPC interactions (step 5).

Mobile Agents in Mobile Systems

Application areas:
1. End user Application, e.g. stock information from 3rd Party SPs
2. VHE: Adaptation of MS capabilities to visited network capabilities
3. Within Mobile System: MA-based Mobility Management



Based on the fact, that IN and TMN, considered today as the basis for the evolution of mobile communication system, will be affected by both distributed object technology and mobile agent technology, it seems to be logical to study these impacts also on mobile communication systems.

Basically three main areas have to be addressed in such an analysis:

1. Usage of agent technology for value added service subscription and provision to the end user, such as dynamic downloading of application software between the end user’s mobile terminal and the 3rd party service provider. In this case the mobile communications system is just the transport means for mobile agents.

2. Usage of agent technology for mobile communication service provision to the end user, such as dynamic downloading of appropriate/customized mobile communications service logic into arbitrary mobile stations. This allows for flexible extension of service software and increasing intelligence in the mobile terminal required for the Virtual Home Environment (VHE) implementation.

3. Usage of agent technology within the mobile communication system for flexible control and management solutions, such as mobile subscriber profiles. For example appropriate control logic can be flexible distributed across access network, i.e. follow the roaming user.

For further reading:

• D. Chess et al.: "Itinerant Agents for Mobile Computing", IEEE Personal Communications Magazine, October 1995

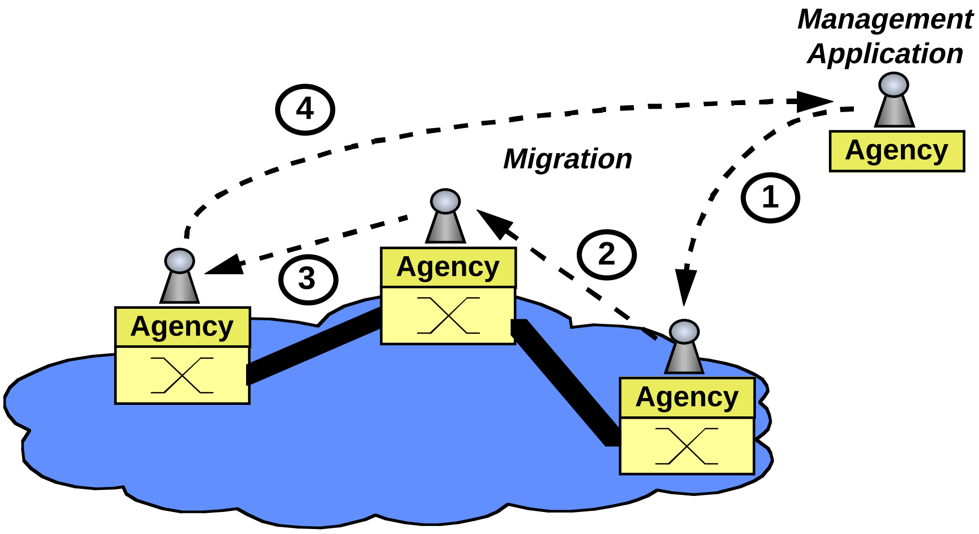
• R. Ramjee et al.: "The Use of Network-Based Migrating User Agents for Personal Communication Services", IEEE Personal Communications Magazine, December 1995

• T. Magedanz, L. Hagen, M. Breugst: ”Impacts of Mobile Agent Technology on Mobile Communication System Evolution”, pp.56-69, IEEE Personal Communications Magazine, Vol. 5, No. 4, August 1998

Agent-Based Telecom Management

Application areas in Service and Network Management

- Management by Delegation



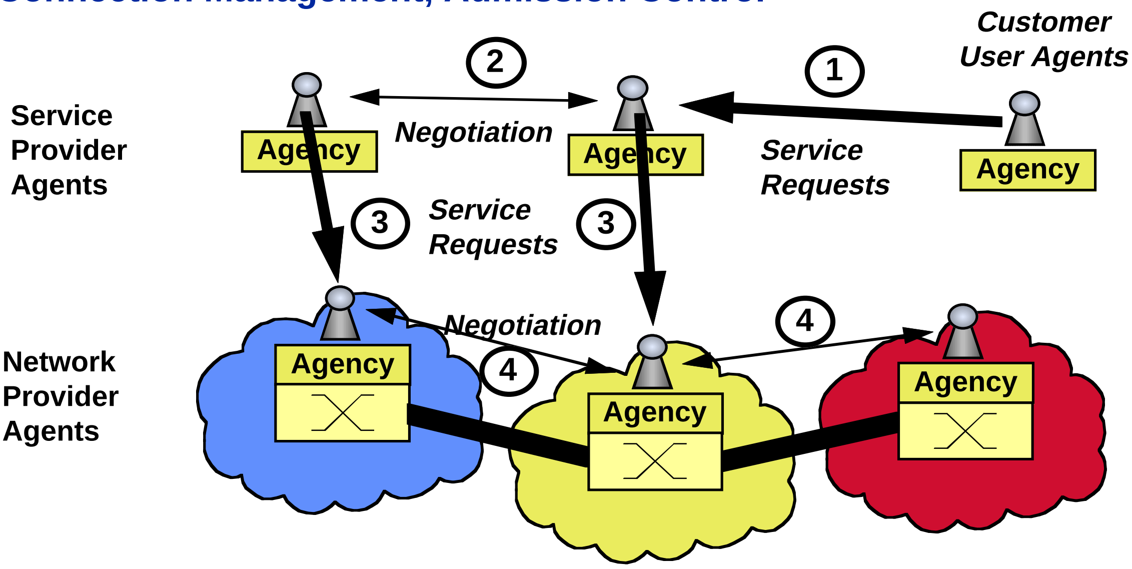
The distribution of management tasks in (network) management represents a major research issue since many years. This field of research, also referred to as "Management by Delegation (MBD)", adopts a decentralized management paradigm that takes advantage of the increased computational power in network nodes (i.e. management agents) and decreases pressure on centralized network management systems and network bandwidth. MBD enables both temporal distribution (i.e. distribution over time) and spatial distribution (i.e. distribution over different network nodes). The basic idea is to increase the management autonomy of management agents by running socalled "elastic procceses" on the "elastic" agents, which could absorb dynamically (i.e. by delegation) new functions. Besides the distributio of pure management applications the distribution of connection management logic emerged the last years. Today this approach provides the basis for active network research (see later in this talk).

In addition to the usage of code mobility, intelligent agent research concentrates on the negotiation between fixed agents controlling the connectivity and bandwidth of network nodes. This means that these agents could offer capacities in an open market model. One FIPA application is looking at a VPN implementation through negotiating agents.

Agent-Based Telecom Management

Application areas in Service and Network Management

- Connection Management, Admission Control



The distribution of management tasks in (network) management represents a major research issue since many years. This field of research, also referred to as "Management by Delegation (MBD)", adopts a decentralized management paradigm that takes advantage of the increased computational power in network nodes (i.e. management agents) and decreases pressure on centralized network management systems and network bandwidth. MBD enables both temporal distribution (i.e. distribution over time) and spatial distribution (i.e. distribution over different network nodes). The basic idea is to increase the management autonomy of management agents by running socalled "elastic procceses" on the "elastic" agents, which could absorb dynamically (i.e. by delegation) new functions. Besides the distributio of pure management applications the distribution of connection management logic emerged the last years. Today this approach provides the basis for active network research (see later in this talk).

In addition to the usage of code mobility, intelligent agent research concentrates on the negotiation between fixed agents controlling the connectivity and bandwidth of network nodes. This means that these agents could offer capacities in an open market model. One FIPA application is looking at a VPN implementation through negotiating agents.

The Future Vision: Active Networks

- **Term “Active Network (AN)” originates from Internet community, but same ideas exist in the telecom world**
- **Idea: Let the network take an active part in the service processing**
- **Key concepts: Programmable Switches & Mobile Code Systems**
- **Two approaches for AN exist:**
- **Discrete Approach - Programmable Switches (short term)**
  - **separation of service deployment and service processing**
  - **outband service deployment**
- **Integrated Approach - Capsules (long term)**
  - **every message is a programm, i.e. it contains a program fragment**
  - **inband service deployment and subsequent processing**
- **Hybrid Approach is possible**
  - **Send a capsule inband, subsequently fetch program outband**

Today flexibility is a key design issue for emerging network architectures in order to adapt instantly to changing customer service demands. In this context an emerging vision is to allow the network itself to take an active part in service provision, evolving from a dump transport network towards a large distributed processing environment. The basic idea of such "Active Networks" is the movement of service code, which has been traditionally placed outside the transport network, directly to the network’s switching nodes. Furthermore, this movement of service code should be possible in a highly dynamic manner. This allows the automated, flexible, and customised provision of services in a highly distributed way thereby enabling better service performance and optimised control and management of transport capabilities.

Taking into account the current research related to the developments toward open, active, and programmable networks within both telecommunications and internet communities, it becomes obvious that two enabling technologies are key in both worlds: programmable switches, which provide flexibility in the design of connectivity control applications, and mobile code systems / mobile agent platforms, which enable the dynamic downloading and movement of service code to specific network nodes.

Whereas current active network research concentrates mostly on multimedia information processing within internet environments, such as packet filtering, format conversion, packet distribution (i.e. multicasting), etc. in order to optimize the usage of the transmission networks, value added services are not yet considered in this context. However, this issue is becoming important in the context of PSTN / internet integration , where for example enhanced service control capabilities may have to be provided to internet-based telecom services, e.g. internet telephony.

Useful References:

• D.L. Tennenhouse, et.al.: "A Survey of Active Network Research", IEEE Communications Magazine, pp. 80-85, Vol. 35, No. 1, January 1997

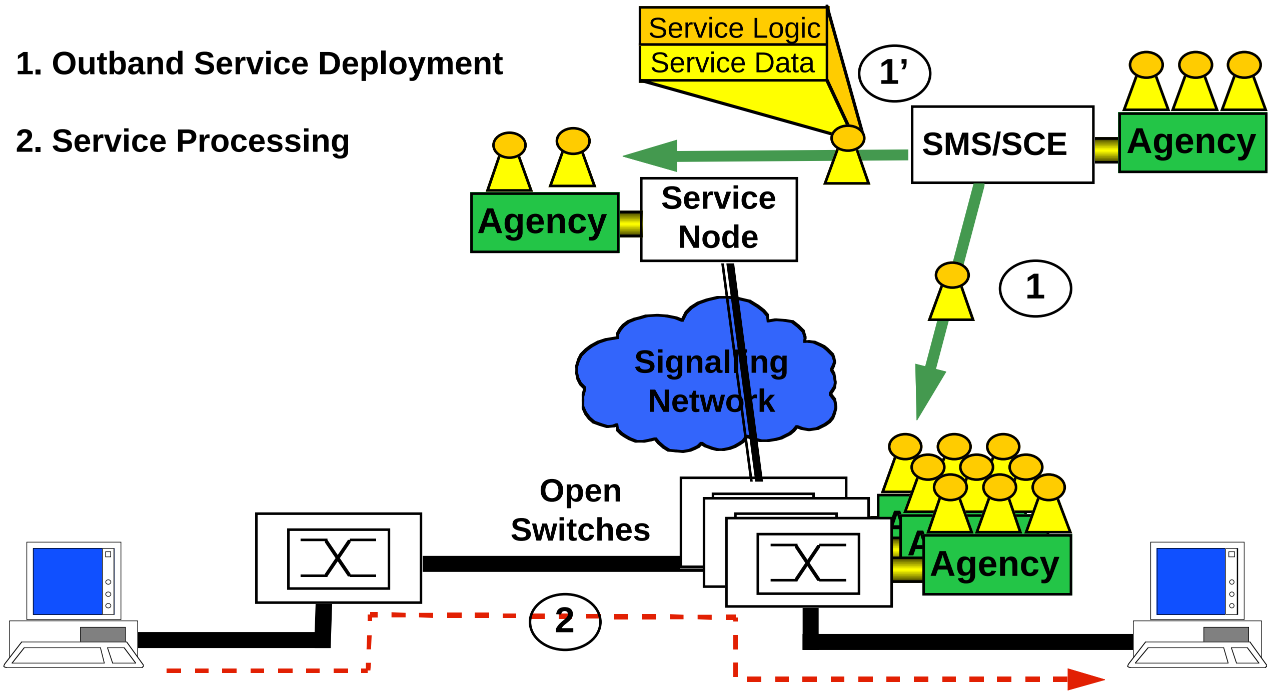
• Active Networks home page of MIT Lab for Computer Science, "http://www.tns.lcs.mit.edu/activeware/"

• Summafour Open Programmable Switching: http://www.summa4.com/products/openswitch.htm

• IEEE P1520 - Proposed IEEE Standard for Application Programming Interfaces for Networks: http://www.iss.nus.sg/IEEEPIN/

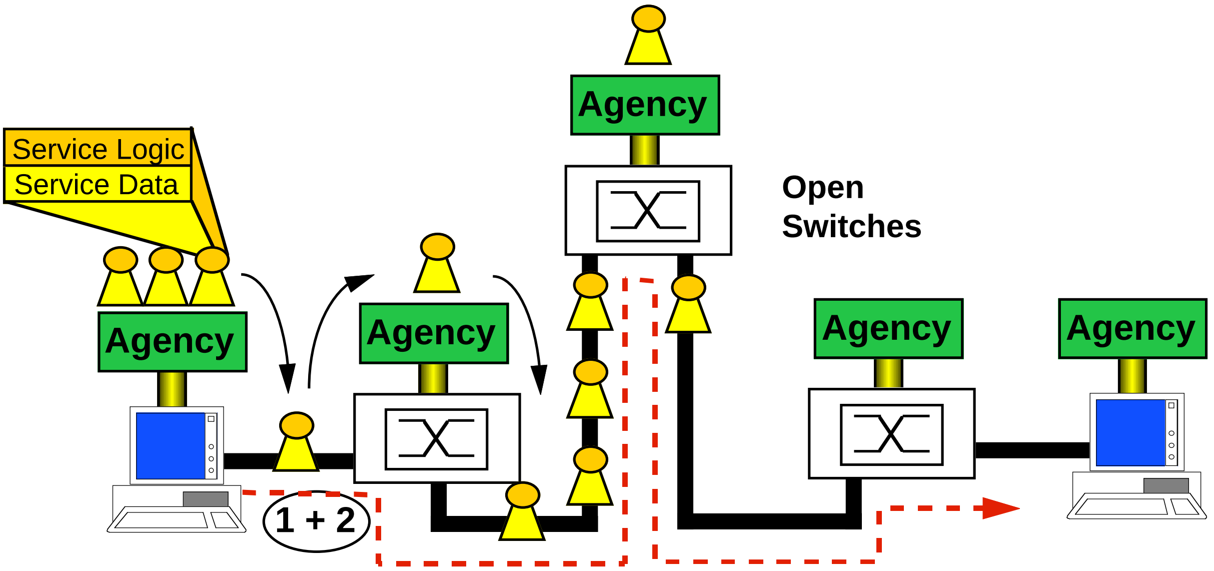
Active Networks: Programmable Switches

1. Outband Service Deployment
2. Service Processing

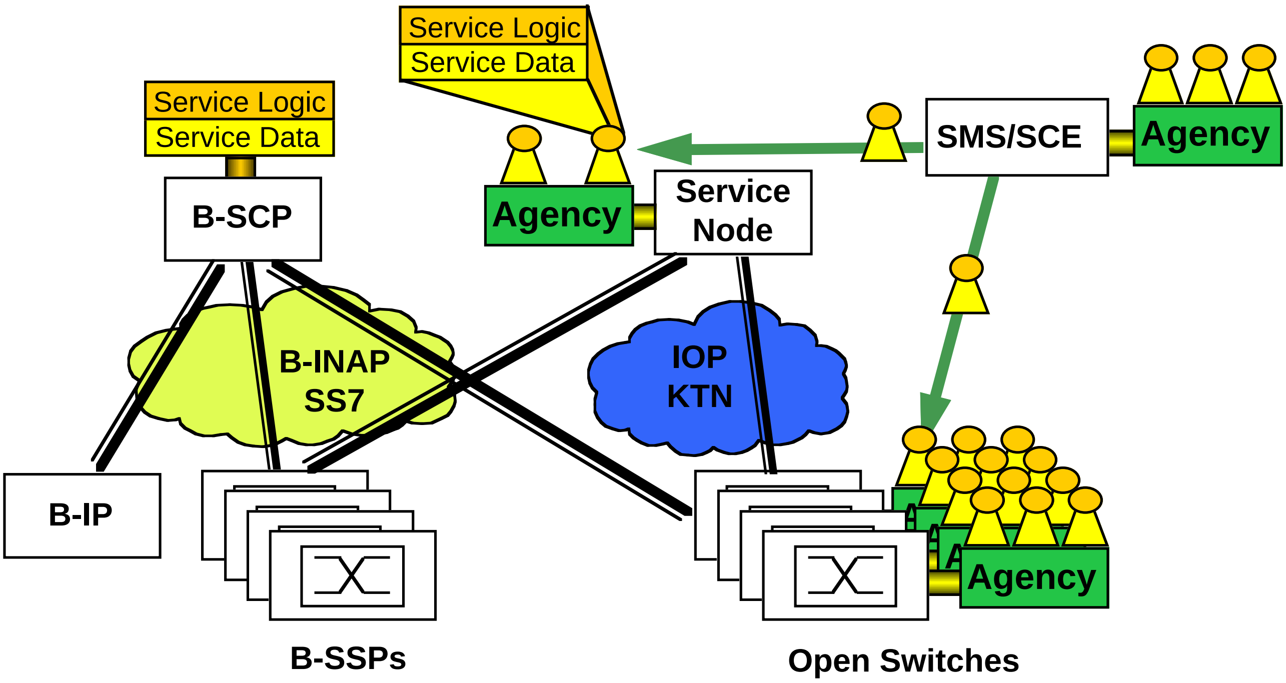


Active Networks: Capsules Approach

Integrated Inband Service Deployment (1) and Service Processing (2)



Active Intelligent Network - MARINE

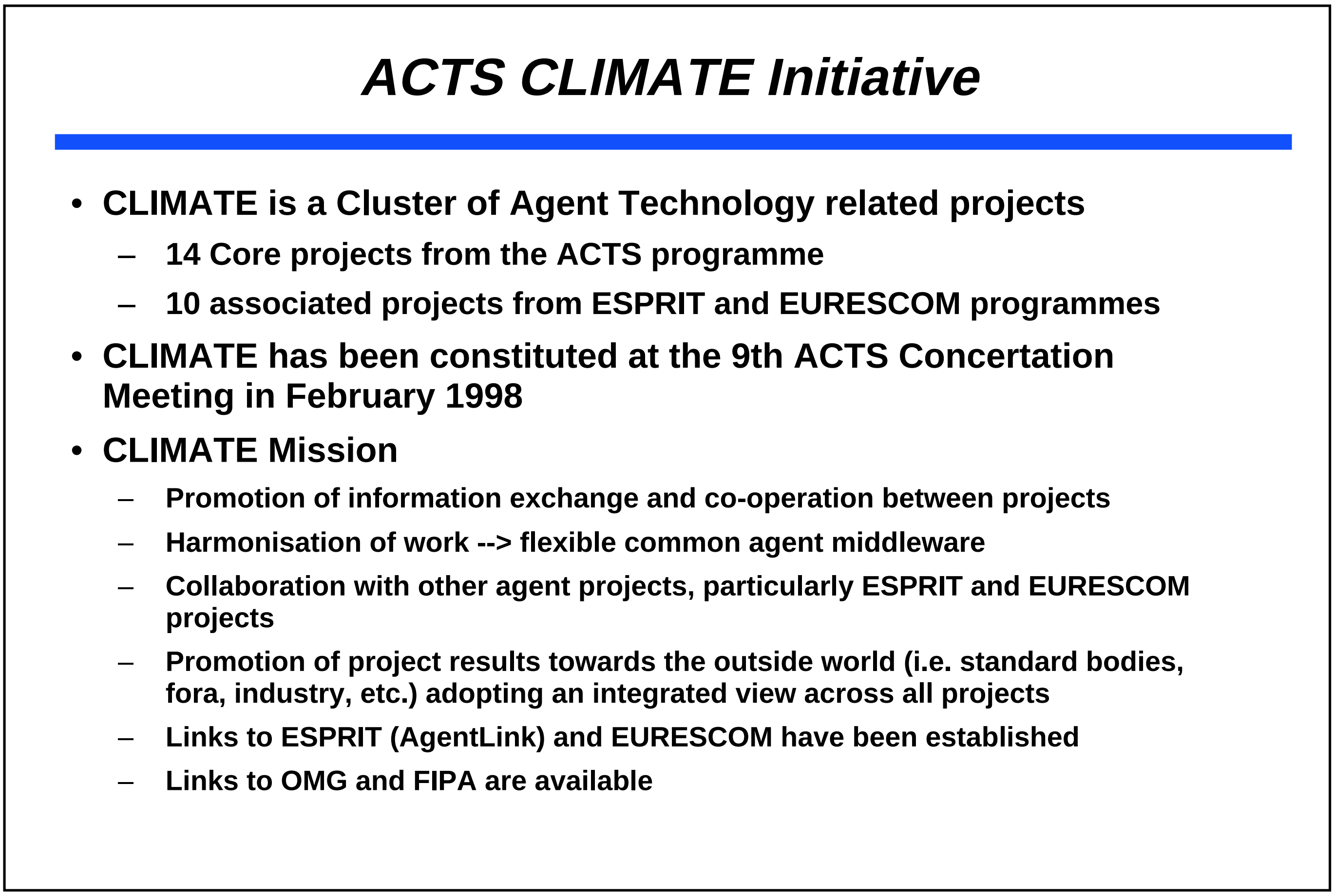


The European research programme on Advanced Communication Technologies and Services (ACTS) has launched in March 1998 a cluster of 14 projects dedicated to explore the usage of agent technologies in the area of telecommunications. As part of this cluster, the project MARINE ("Mobile Agent Environment in Intelligent Networks") is realising some of the presented ideas for implementing Broadband IN services. Other projects focus on the usage of agent technology in the context of mobile communications, service and network management, and electronic commerce.

References:

Information about the Cluster for Intelligent Mobile Agents for Telecommunications Environments (CLIMATE): http://www.fokus.gmd.de/cc/ima/climate/

Web site of the ACTS MARINE project: “http://www.italtel.it/drsc/marine/marine.htm”



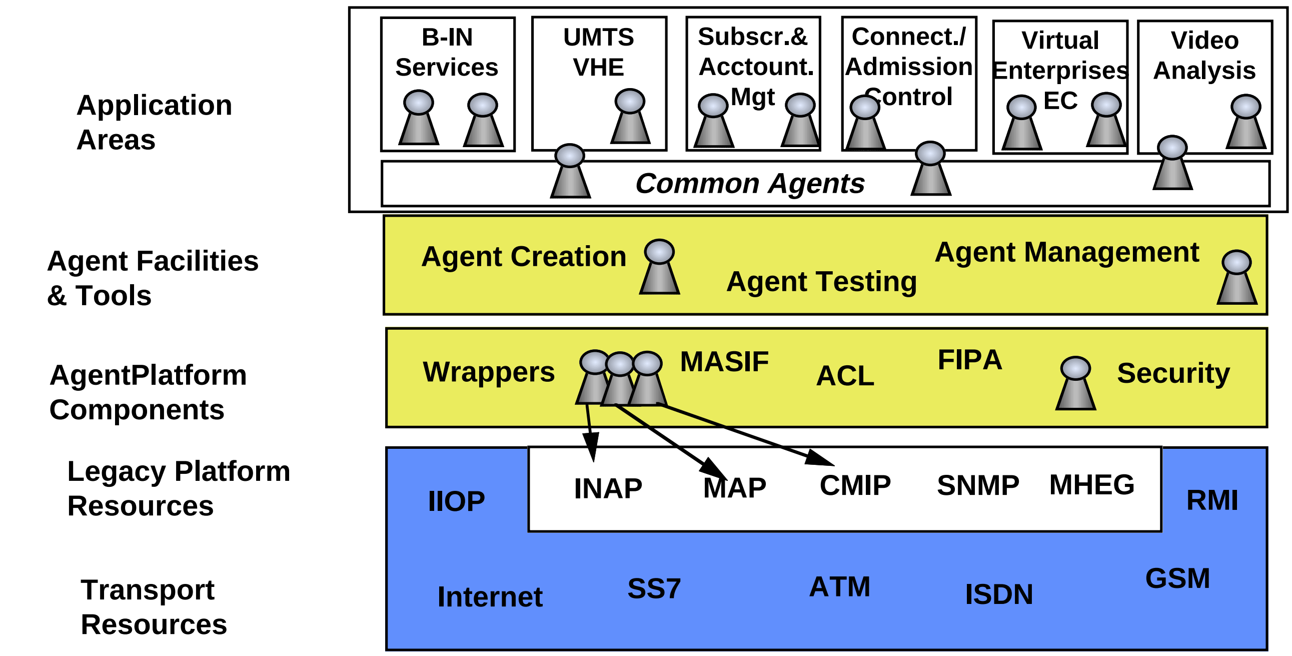
ACTS CLIMATE Initiative

- **CLIMATE is a Cluster of Agent Technology related projects**
  - **14 Core projects from the ACTS programme**
  - **10 associated projects from ESPRIT and EURESCOM programmes**
- **CLIMATE has been constituted at the 9th ACTS Concertation Meeting in February 1998**
- **CLIMATE Mission**
  - **Promotion of information exchange and co-operation between projects**
  - **Harmonisation of work --> flexible common agent middleware**
  - **Collaboration with other agent projects, particularly ESPRIT and EURESCOM projects**
  - **Promotion of project results towards the outside world (i.e. standard bodies, fora, industry, etc.) adopting an integrated view across all projects**
  - **Links to ESPRIT (AgentLink) and EURESCOM have been established**
  - **Links to OMG and FIPA are available**

CLIMATE's mission is to optimise the information exchange and to promote co-operation between these projects in order to enable the harmonisation of work, which ideally will result in a flexible common agent middleware, which could be used for different application domains. An important aspect of CLIMATE is the collaboration with other agent projects particularly in ESPRIT and EURESCOM, which are considered as associated CLIMATE projects. However, also other external projects may become associated members.

CLIMATE promotes the ACTS agent project activities and results towards the outside world (i.e. standard bodies, fora, industry, etc.) taking advantage of adopting an integrated view across all projects. It is envisaged that CLIMATE is taking an active part in contributing to relevant agent standards (e.g. OMG, FIPA) and telecommunication standards (e.g. IN, TMN, UMTS standardisation).

CLIMATE Populates an Agent Framework



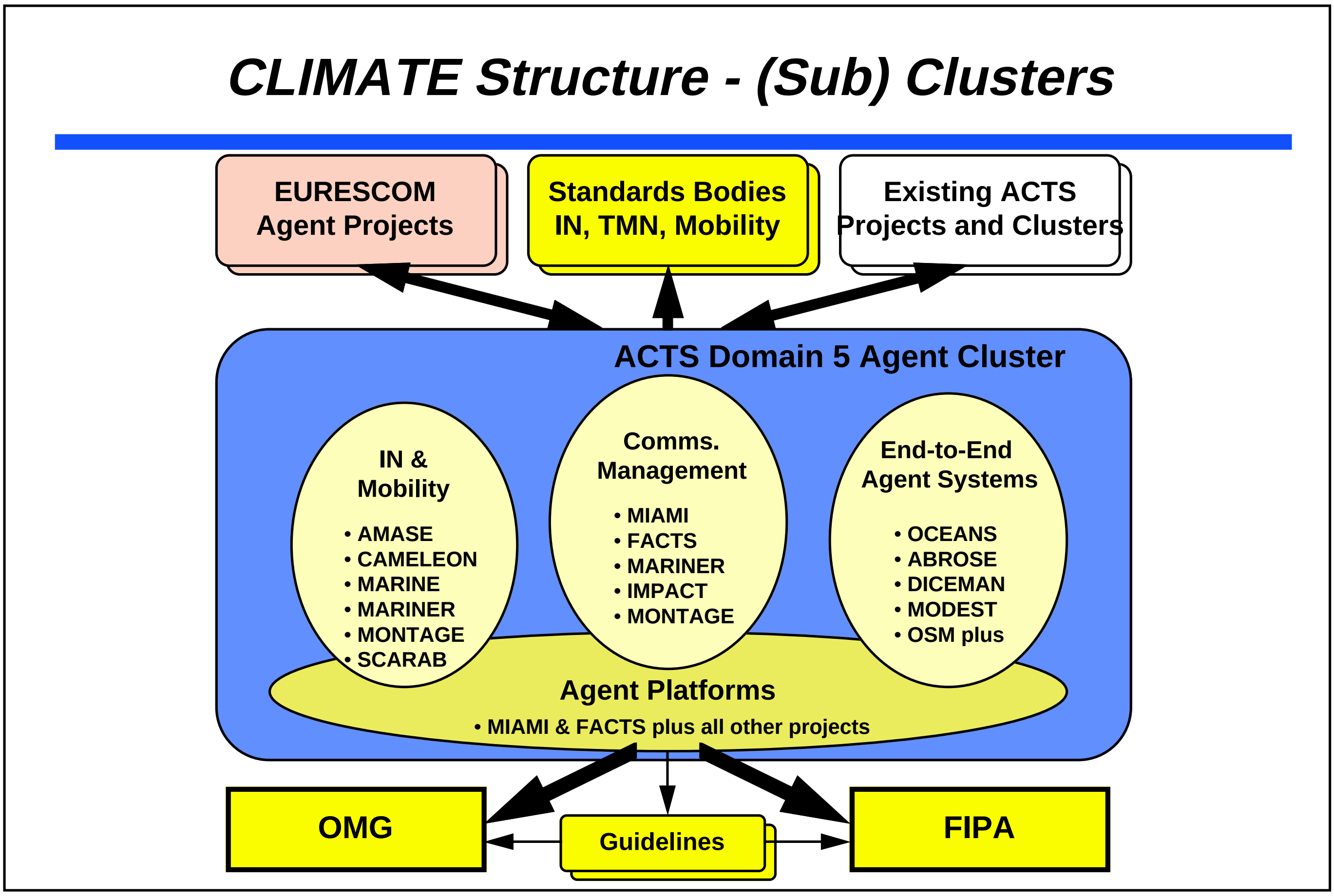
CLIMATE projects are contributing to an overall Agent Framework which is depicted above. This looks quite similar to OMG’s Object Management Architecture

Starting buttom up there are a variety of transport resources, which have to be controlled and managed in the telecommunications environment, spanning fixed and mobile smallband and broadband networks, as well as the circuit switched and packet data networks, such as the internet. These networks are controlled and managed through dedicated (legacy) protocols and interfaces, which have to be taken into account when developing advanced telecommuncation solutions. Furthermore, this includes also the access to databases and other services.

The access to these protocols (and thus the networks) will be performed via appropriate Wrappers or gateways to be provided in an agent environment, i.e. in an agent platform. Since there are different agent standards with different focal points, different platforms exist (i.e. mobile agent platforms versus intelligent agents). Within CLIMATE there is the idea to develop a Mobile Intelligent Agent (MIA) platform. Therefore all the projects, although focusing typically on different platforms, contribute through their developments to different components of such a generic agent platform. In fact an agent platform provides services through static service agents.

Additional (horizontal) facilities and tools, such as agent creation environments, testing environments, or agent service management systems enhance the capabilities of the platform itself. Although different projects focus on different tools and facilities it is believed that there is some common understanding in the end in regard to what is needed to be developed for supporting agent based application development. For example the relationship between DOT based service design and MAT based service design has to be studied.

On the top we have the application domains, which can be compared to vertical facilities. Specific applications will be implemmented by dedicated agent types, which require at the platform level specific wrappers in order to control or manage specific networks. However, there may be common agents for different application domains, which could be considered to be part of the agent facility level, such monitoring or subscription agents, user agents, terminal agents, etc.. The deveopment of an Agent Catalogue is a basic target for CLIMATE.



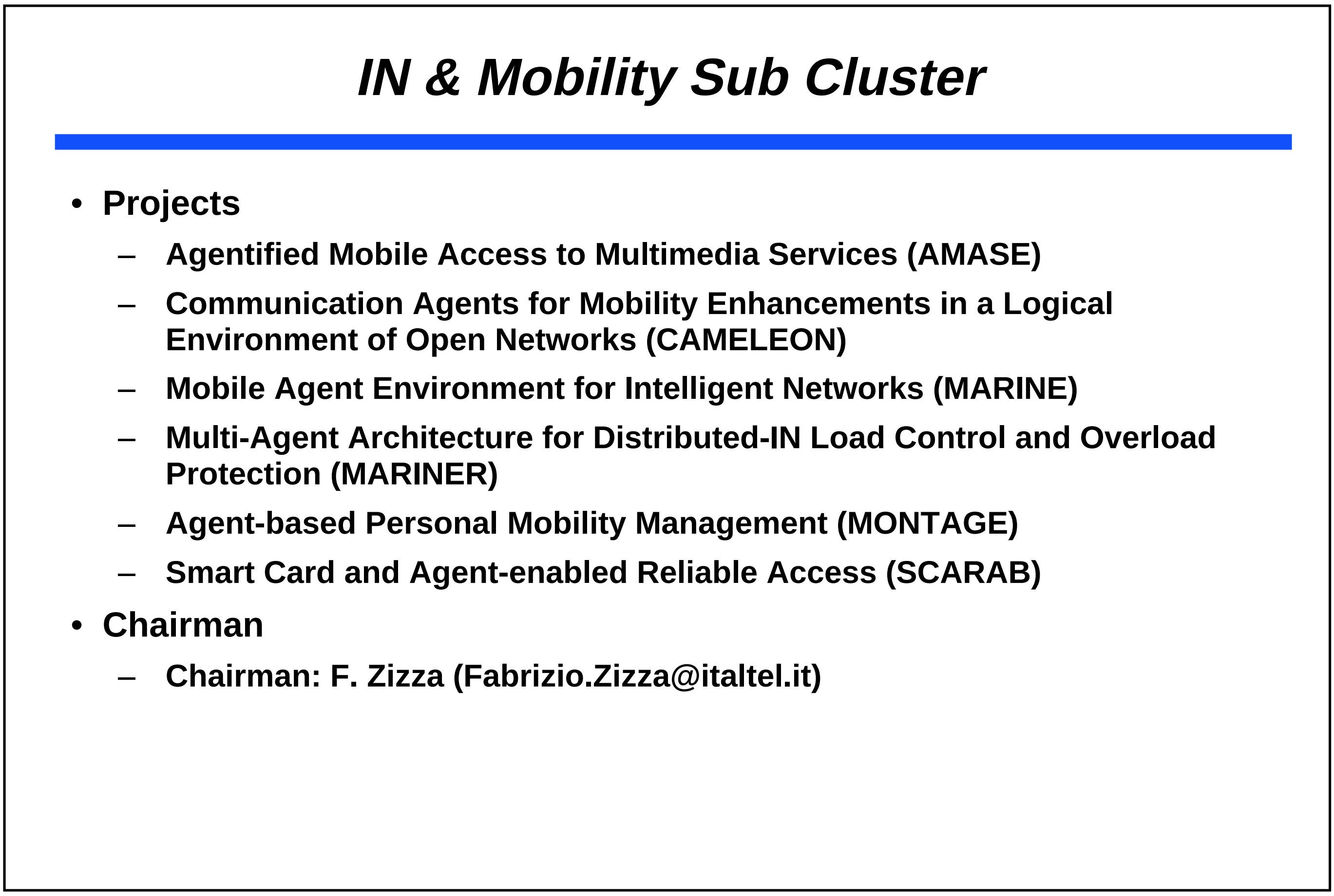
CLIMATE is structured into so-called sub-clusters in accord to three major application domains addressed by the core CLIMATE projects. However, it should be noticed, that a project may be active in multiple sub-clusters, since some projects are addressing several. These are:

The mission of the IN & Mobility Sub-cluster is to explore the application of Intelligent Mobile Agent Technologies in the context of IN-based and mobile telecommunication environments. Focal points of research include extension of existing IN and mobile architectures, including mobile devices and smart cards by introduction of mobile agent platforms, in order to provide ubiquiteously new applications, comprising Video an Demand, UMTS Virtual Home Environment, Customer Profile Management, mobility support, financial services, and dynamic provider selection.

The Communications Management Sub-cluster is looking at the impact of mobile and intelligent agent technologies on the design of service and network management services. Besides the development of agent-based management platforms, the management areas addressed by the projects forming this subcluster include IN/SS7 load control, connection admission control for ATM networks, accounting and charging services for fixed and mobile environments, and service reservation.

The End-to-End Agent Systems Subcluster is concerned with agent software designs which are concentrated in "end-systems" rather than within a network. The ACTS projects involved span a variety of application domains but recognise common DAI issues which can be illuminated optimally in this inter-project forum. To this end, the sub-cluster seeks to analyse selected design topics in relation to agent negotiation, knowledge representation, and human-to-agent interaction. Considerations, will where possible, include relevant standards with a view to comment and input from member projects and/or the sub-cluster as a whole.

The Agent Platform Sub-cluster is providing a forum of information exchange on agent platform issues, comprising experiences in the usage of available platform products, extension of basic platform capabilities as well as application related extensions.



IN & Mobility Sub Cluster

- **Projects**
  - **Agentified Mobile Access to Multimedia Services (AMASE)**
  - **Communication Agents for Mobility Enhancements in a Logical Environment of Open Networks (CAMELEON)**
  - **Mobile Agent Environment for Intelligent Networks (MARINE)**
  - **Multi-Agent Architecture for Distributed-IN Load Control and Overload Protection (MARINER)**
  - **Agent-based Personal Mobility Management (MONTAGE)**
  - **Smart Card and Agent-enabled Reliable Access (SCARAB)**
- **Chairman**
  - **Chairman: F. Zizza (Fabrizio.Zizza@italtel.it)**

Agentified Mobile Access to Multimedia Services (AMASE),

Contact: R. Pascotto (R.Pascotto@berkom.de), WWW: http://b5www.berkom.de/amase/

Keywords: user mobility, Wireless Access to Multimedia Services, Scalable Agent Platform, Small and mobile devices, WAP.

Communication Agents for Mobility Enhancements in a Logical Environment of Open Networks (CAMELEON),

Contact: Dr. F. Ramme (eedfra@eed.ericsson.se), WWW: http://www.comnets.rwth-aachen.de/~cameleon/

Keywords: Mobile and Intelligent Agent Technology, Ubiquitous Telecommunication Services, Service Roaming, Virtual Home Environment, VHE, UMTS, IMT-2000

Mobile Agent Environment for Intelligent Networks (MARINE),

Contact: Fabrizio Zizza (Fabrizio.Zizza@italtel.it), WWW: http://www.italtel.it/drsc/marine/marine.htm

Keywords: IN, user mobility, service logic, DPE, MAT, MA.

Multi-Agent Architecture for Distributed-IN Load Control and Overload Protection (MARINER),

Contact: Brendan Jennings (brendan.jennings@teltec.dcu.ie), WWW: http://www.teltec.dcu.ie/mariner

Keywords: Agents, Multi-Agent Systems, FIPA, OMG MASIF, Intelligent Networks, CORBA, Load Control, Overload Protection, Real-time.

Agent-based Personal Mobility Management (MONTAGE),

Contact: Mrs. Didoe Prevedourou (dpre@intranet.gr), WWW: http://montage.ccrle.nec.de

Keywords: Ubiquitous Service Provisioning, Service Selection, Personal Mobility, Accounting and Charging, Hypermedia Information Browsing

Smart Card and Agent-enabled Reliable Access (SCARAB),

Contact: Michel Van Ackere (ackerem@btmaa.bel.alcatel.be), WWW: http://www.scarab.montrouge.tt.slb.com:65530/

Keywords: smart card, agent, architecture, service, service access, mobility, user interface.

Communications Management Sub Cluster

- **Projects**
  - **FIPA Agent Communication Technologies and Services (FACTS)**
  - **Multi Agent-based ATM Management (IMPACT)**
  - **Multi-Agent Architecture for Distributed-IN Load Control and Overload Protection (MARINER)**
  - **Mobile Intelligent Agents for Managing the Information Infrastructure (MIAMI)**
  - **Agent-based Personal Mobility Management (MONTAGE)**
- **Chairman**
  - **Dr. S. Covaci (Covaci@fokus.gmd.de)**

FIPA Agent Communication Technologies and Services (FACTS)

Contact: Alan Steventon (alan.steventon@bt-sys.bt.co.uk), WWW: http://www.labs.bt.com/profsoc/facts/

Keywords: Agent interoperability, FIPA,Intellgent Agents, electronic commerce, service reservation, audio-visual broadcasting and entertainment.

Multi Agent-based ATM Management (IMPACT)

Contact: John Bigham (J.Bigham@qmw.ac.uk), WWW: http://www.acts-impact.org/

Keywords: intelligent agents, agent architecture, Connection Admission Control (CAC), control strategies, accounting/ charging, Agent Control Language (ACL), real applications, network simulation, FIPA model validation.

Multi-Agent Architecture for Distributed-IN Load Control and Overload Protection (MARINER)

Contact: Brendan Jennings (brendan.jennings@teltec.dcu.ie), WWW: http://www.teltec.dcu.ie/mariner

Keywords: Agents, Multi-Agent Systems, FIPA, OMG MASIF, Intelligent Networks, CORBA, Load Control, Overload Protection, Real-time.

Mobile Intelligent Agents for Managing the Information Infrastructure (MIAMI)

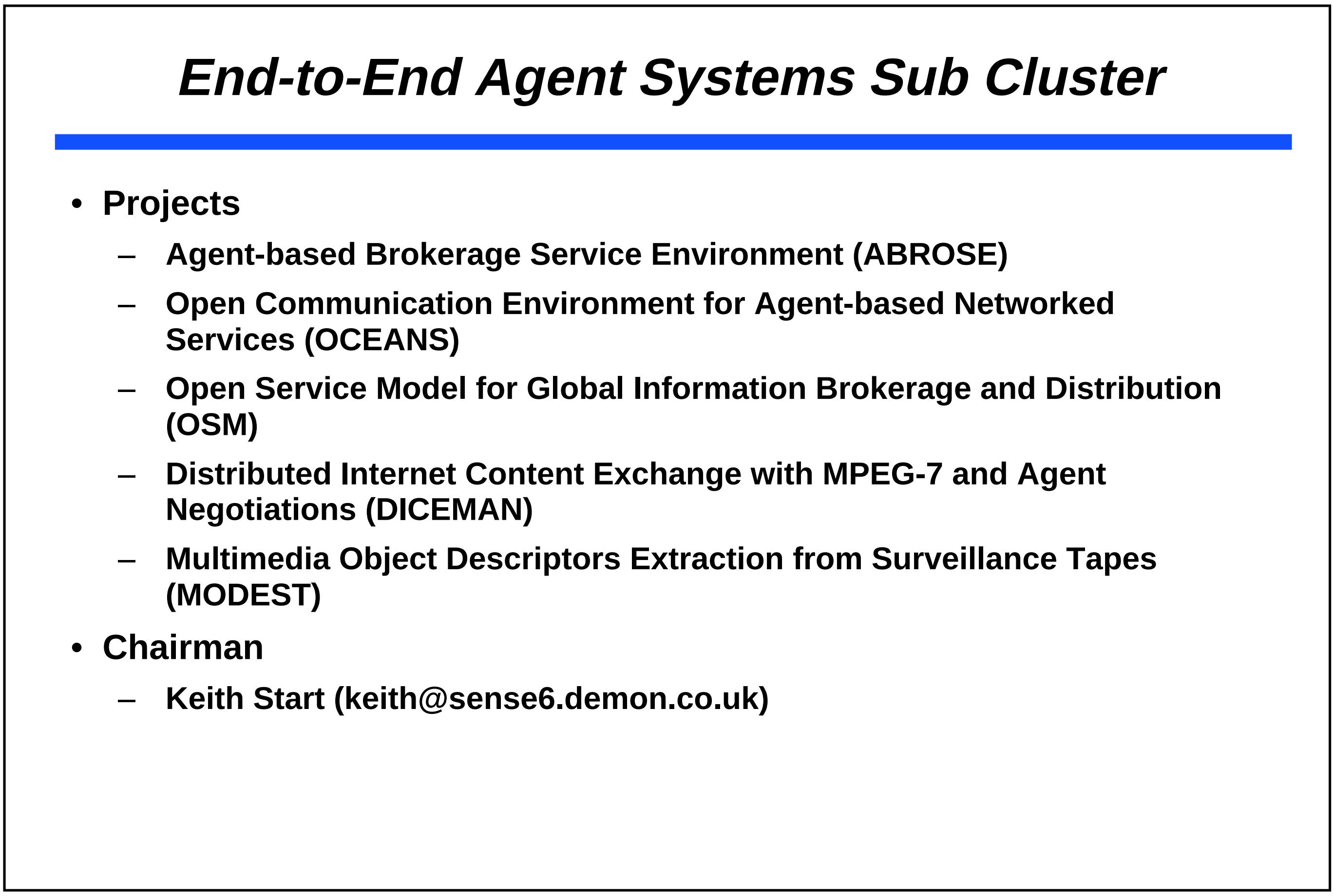
Contact: Dr. Stefan Covaci (covaci@fokus.gmd.de), WWW: http://www.fokus.gmd.de/ima/miami/

Keywords: mobile agents, network and service management, remote fault management, flexible routing strategies.

Agent-based Personal Mobility Management (MONTAGE)

Contact: Mrs. Didoe Prevedourou (dpre@intranet.gr), WWW: http://montage.ccrle.nec.de

Keywords: Ubiquitous Service Provisioning, Service Selection, Personal Mobility, Accounting and Charging, Hypermedia Information Browsing



End-to-End Agent Systems Sub Cluster

- **Projects**
  - **Agent-based Brokerage Service Environment (ABROSE)**
  - **Open Communication Environment for Agent-based Networked Services (OCEANS)**
  - **Open Service Model for Global Information Brokerage and Distribution (OSM)**
  - **Distributed Internet Content Exchange with MPEG-7 and Agent Negotiations (DICEMAN)**
  - **Multimedia Object Descriptors Extraction from Surveillance Tapes (MODEST)**
- **Chairman**
  - **Keith Start (keith@sense6.demon.co.uk)**

Agent-based Brokerage Service Environment (ABROSE)

Contact: E. Athanassiou (athanassiou@berkom.de), WWW: http://b5www.berkom.de/ABROSE/

Keywords: Information Brokerage, Electronic Commerce, Intelligent Agents, Multi Agent System, Transaction and mediation Agents.

Open Communication Environment for Agent-based Networked Services (OCEANS)

Contact: Stefano Antoniazzi (Stefano.Antoniazzi@italtel.it), WWW: http://www.sintef.no/ACTS-OCEANS

Keywords: agents, set-top-box, Internet, multimedia, protocols, user interface.

Open Service Model for Global Information Brokerage and Distribution (OSM)

Contact: Stephen McConnell (mcconnell@osm.net), WWW: http://www.osm.net

Keywords: OSM, ELECTRONIC COMMERCE, ECOMMERCE, BUSINESS OBJECTS, NEGOTIATION, MEDIATION, CORBA, ECDTF, AGENT, OMG.

Distributed Internet Content Exchange with MPEG-7 and Agent Negotiations (DICEMAN)

Contact: Liam Ward (Liam.Ward@Teltec.DCU.IE), WWW: http://www.teltec.dcu.ie/diceman

Keywords: MPEG-7, FIPA, Internet, audio-visual content, multimedia database, agents.

Multimedia Object Descriptors Extraction from Surveillance Tapes (MODEST)

Contact: B. Macq (macq@tele.ucl.ac.be) or P. Piscaglia (piscaglia@tele.ucl.ac.be), WWW: http://www.tele.ucl.ac.be/MODEST

Keywords: segmentation/tracking, description, intelligent agents, multi-agent system, MPEG, FIPA.

CLIMATE Contact Points

- **CLIMATE Web site:**

  http://www.fokus.gmd.de/cc/ima/climate/

- **For further information contact:**

  – **Dr. Thomas Magedanz (CLIMATE Chairman)**
  IKV++ GmbH, Kurfürstendamm 173-174, D-10707 Berlin, Germany
  Email: magedanz@ikv.de

Contact Points for the different Sub Cluster are:

1. IN & Mobility - Chairman: Fabrizio Zizza (Fabrizio.Zizza@italtel.it)
2. Communication & Management - Chairman: Dr. Stefan Covaci (covaci@fokus.gmd.de)
3. End-to-End Agent Systems - Chairman: Keith Start (keith@sense6.demon.co.uk)
4. Agent Platforms- Chairman: Dr. Stefan Covaci (Covaci@fokus.gmd.de)

Summary

- **Mobile agent technology offers new opportunities for telecommnications in respect to flexible service provision**
- **Integrated platform approach combines client/server paradigm and mobile agent technology**
- **Emerging CORBA-based mobile agent platforms are appearing for telecommunication applications**
- **MA-Based IN environment is based on Agencies within IN elements, enabling dynamic and better distribution of service agents**
- **Intelligence on demand to the switch --> Active Networks**
- **Many research projects within ACTS and EURESCOM look at this**

MA technology is gaining momentum in telecommunications. CORBA-based mobile agent platforms are under development. MA Platforms could be integrated in IN elements in order to provide services on demand.

For further reading:

• D. Chess et al.: „Itinerant Agents for Mobile Computing“, IEEE Personal Communications Magazine, October 1995

• T. Magedanz, K. Rothermel, S. Krause: "Intelligent Agents: An Emerging Technology for Next Generation Telecommunications?", in: Proceedings of IEEE INFOCOM ´96, pp. 464-472, IEEE Catalog No. 96CB35887, ISBN: 0-8186-7292-7, IEEE Press, 1996

• T. Magedanz, R. Popescu-Zeletin: "Towards Intelligence on Demand - On the Impacts of Intelligent Agents on IN", Proceedings of 4th International Conference on Intelligent Networks (ICIN), pp. 30-35,Bordeaux, France, December 2-5, 1996

• M. Breugst, T. Magedanz: ”On the Usage of Standard Mobile Agent Platforms in Telecommunication Environments”, pp. 275-286, in: Lecture Notes of Computer Sciences 1430, Intelligence in Services and Networks: Technologies for Ubiquiteous Telecom Services, S. Trigila et al. (Eds.), ISBN: 3-540-64598-5, Springer Verlag 1998

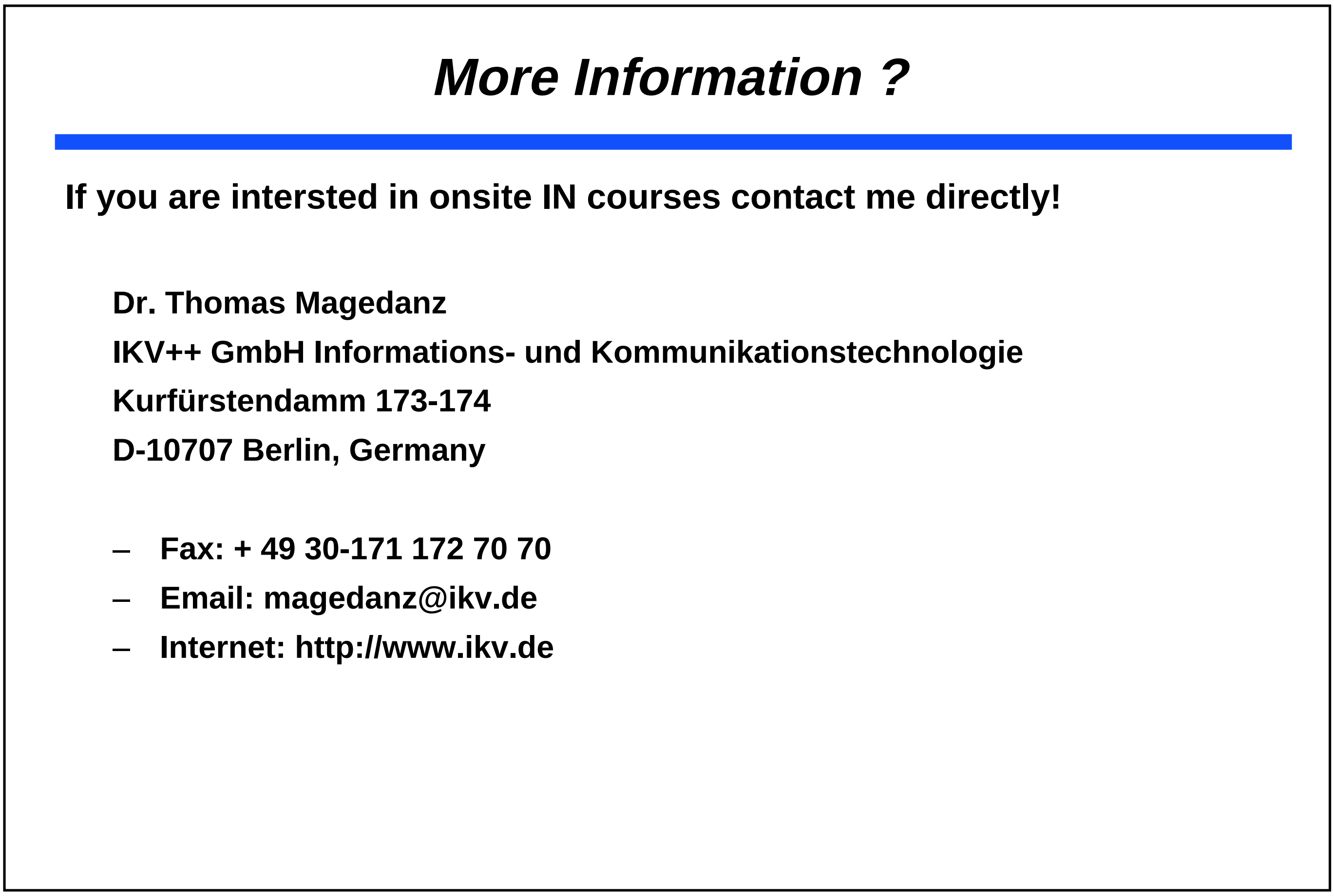
• T. Magedanz, M. Breugst: ”Mobile Agents - Enabling Technology for Active Intelligent Networks”, IEEE Network Magazine, pp. 53-60, Vol. 12, No. 3, Special Issue on Active and Programmable Networks, May/ June 1998

• T. Magedanz, L. Hagen, M. Breugst: ”Impacts of Mobile Agent Technology on Mobile Communication System Evolution”, pp.56-69, IEEE Personal Communications Magazine, Vol. 5, No. 4, August 1998

Putting the pieces together

The IN for the next millenium:

- **Besides mobile services, data service will become dominant**
- **Fixed / mobile / internet convergence (--> unified messaging)**
- **Decentralised service intelligence (distributed objects)**
- **On demand service downloading (mobile code)**
- **Open programmable switches --> flexible basic call models**
- **Internet (IP) will become key bearer transport network**
- **Service Nodes (incl. SRF) are key elements (allow for flexible extension, i.e. gateways)**



More Information ?

If you are intersted in onsite IN courses contact me directly!

Dr. Thomas Magedanz
IKV++ GmbH Informations- und Kommunikationstechnologie
Kurfürstendamm 173-174
D-10707 Berlin, Germany

- **Fax: + 49 30-171 172 70 70**
- **Email: magedanz@ikv.de**
- **Internet: http://www.ikv.de**