International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

Y.2901

(12/2006)

SERIES Y: GLOBAL INFORMATION INFRASTRUCTURE, INTERNET PROTOCOL ASPECTS AND NEXT-GENERATION NETWORKS

Next Generation Networks

The carrier grade open environment reference model

ITU-T Recommendation Y.2901

ITU-T Y-SERIES RECOMMENDATIONS

GLOBAL INFORMATION INFRASTRUCTURE, INTERNET PROTOCOL ASPECTS AND NEXT-GENERATION NETWORKS

GLOBAL INFORMATION INFRASTRUCTURE	
General	Y.100–Y.199
Services, applications and middleware	Y.200–Y.299
Network aspects	Y.300–Y.399
Interfaces and protocols	Y.400–Y.499
Numbering, addressing and naming	Y.500–Y.599
Operation, administration and maintenance	Y.600–Y.699
Security	Y.700–Y.799
Performances	Y.800–Y.899
INTERNET PROTOCOL ASPECTS	
General	Y.1000–Y.1099
Services and applications	Y.1100–Y.1199
Architecture, access, network capabilities and resource management	Y.1200–Y.1299
Transport	Y.1300–Y.1399
Interworking	Y.1400–Y.1499
Quality of service and network performance	Y.1500–Y.1599
Signalling	Y.1600–Y.1699
Operation, administration and maintenance	Y.1700–Y.1799
Charging	Y.1800–Y.1899
NEXT GENERATION NETWORKS	
Frameworks and functional architecture models	Y.2000–Y.2099
Quality of Service and performance	Y.2100–Y.2199
Service aspects: Service capabilities and service architecture	Y.2200–Y.2249
Service aspects: Interoperability of services and networks in NGN	Y.2250–Y.2299
Numbering, naming and addressing	Y.2300–Y.2399
Network management	Y.2400–Y.2499
Network control architectures and protocols	Y.2500–Y.2599
Security	Y.2700–Y.2799
Generalized mobility	Y.2800–Y.2899

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

ITU-T Recommendation Y.2901

The carrier grade open environment reference model

Summary

ITU-T Recommendation Y.2901 presents the carrier grade open environment reference model.

Source

ITU-T Recommendation Y.2901 was approved on 14 December 2006 by ITU-T Study Group 13 (2005-2008) under the ITU-T Recommendation A.8 procedure.

FOREWORD

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

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

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

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

NOTE

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

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure e.g. interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

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

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database at http://www.itu.int/ITU-T/ipr/.

CONTENTS

ITU-T Recommendation Y.2901

The carrier grade open environment reference model

1 Scope

This Recommendation presents the carrier grade open environment (CGOE) reference model that is used to categorize CGOE components. These CGOE components are intended for use in commercial off the shelf (COTS) components.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T Y.2012] ITU-T Recommendation Y.2012 (2006), *Functional requirements and architecture of the NGN release 1*.

3 Definitions

This Recommendation defines the following terms:

3.1 application: An application is a piece of software answering a set of user's requirements using telecommunication network services via an IT system.

3.2 application programming interface (API): A boundary across which a software application uses facilities of programming languages to invoke software services.

NOTE – These facilities may include procedures or operations, shared data objects and resolution of identifiers.

3.3 building block: A logical unit, composed of components, characterized by its ability to deliver certain self-contained service functionality.

3.4 carrier grade: Colloquially, a "carrier grade" implementation of a solution, building block, or a COTS component exhibits particular qualities beyond regular information technology (IT) reliability, availability, serviceability, and manageability (RASM) features, enabling its mission-critical use in a service provider's offering.

NOTE – COTS component can be called "carrier grade" with respect to a particular building block if it meets all of the necessary and sufficient non-functional requirements of a COTS category for such a building block.

3.5 CGOE category: A unit of description of the CGOE reference model. It comprises one or more CGOE components.

NOTE – This method of abstraction keeps the size of the framework manageable and understandable. It avoids being too specific or leaning towards the needs of a certain building block. For example, the alarm management category consists of several components, e.g., alarm generation and alarm clearance.

3.6 CGOE component: A CGOE component is an abstract description of technical tasks, interfaces and properties.

3.7 CGOE reference model: A model that organizes the CGOE categories.

NOTE 1 – Each category is intended to be independent in the sense that it does not require the existence of the categories above it; however, to produce carrier grade functionality, functions may be needed from more than one category.

NOTE 2 – Multiple categories are logically grouped and referred to as the server hardware and the operating platform.

3.8 COTS component: A hardware or a software component instantiation of one or more CGOE components.

NOTE 1 – Existing or new components may instantiate CGOE components.

NOTE 2 – The following are examples of components: database system, operating system and management middleware.

3.9 component instance: A component instance is a specific representation of a component, which satisfies the needs of building a specific building block.

NOTE – Technology providers develop component instances. During the engineering process within the solution providers, instances are chosen according to the requirements, and are integrated to eventually stage the entire building block. Examples of component instances: Linux, management middleware for Q3-access.

3.10 control plane: The control plane performs the call/session control and connection control functions.

3.11 diameter: An IETF protocol that may be used to provide an authentication, authorization and accounting (AAA) framework for applications.

3.12 framework: A framework is an environment that provides a partial solution, usually automating a particularly tedious or difficult part of an application project.

3.13 functional requirements: The set of interfaces, capabilities and features developed with respect to a service architecture associated with a building block.

3.14 lifecycle management: The management of a component, including loading it into memory, allocating the system resources it needs and removing it when it is complete.

NOTE – The lifecycle management of a component also comprises the software management functions, i.e., first installation of the component, the management of upgrades and updates of new releases/versions of the component.

3.15 management plane: The management plane performs management functions for the transport plane, the control plane and the system as a whole. It may also provide coordination between all the planes.

3.16 middleware: The mediating entity between two information elements. Such an element can be, for example, an application, an infrastructure component or another mediating entity.

3.17 non-functional requirements: A list of features that a building block must provide in order to ensure certain behaviour within the service architecture.

NOTE – This list mostly represents requirements to allow for smooth operations and lifecycle management.

3.18 open component: A component can be called "open" when it is capable of being accepted, rejected, extended and replaced in a building block with minimal restriction and regulation, following commonly-accepted, publicly-available criteria and open standards for the interfaces.

3.19 open standards: Standards made available to the general public which are developed (or approved) and maintained via a collaborative and consensus-driven process. Other elements of "open standards" include, but are not limited to:

- Collaborative process – voluntary and market-driven development (or approval) following a transparent consensus-driven process that is reasonably open to all interested parties.
- Reasonably balanced – ensures that the process is not dominated by any one interest group.

- Due process – includes consideration of, and response to, comments by interested parties.
- Intellectual property rights (IPRs) – IPRs essential to implement the standard to be licensed to all applicants on a worldwide, non-discriminatory basis, either 1) for free and on other reasonable terms and conditions, or 2) on reasonable terms and conditions (which may include monetary compensation). Negotiations are left to the parties concerned and are performed outside the SDO.
- Quality and level of detail – sufficient to permit the development of a variety of competing implementations of interoperable products or services. Standardized interfaces are not hidden and are not controlled other than by the SDO promulgating the standard.
- Publicly available – easily available for implementation and use at a reasonable price. Publication of the text of a standard by others is permitted only with the prior approval of the SDO.
- On-going support – maintained and supported over a long period of time.

NOTE – "Open standards" facilitate interoperability and data exchange among different products or services and are intended for widespread adoption.

3.20 operating platform: An operating platform is an amalgam of many different infrastructure technologies that host application systems.

NOTE – The following are examples of key components of an operating platform: operating system, programming language, human interface representation, database server, security infrastructure and management infrastructure.

3.21 service plane: The service plane comprises:

a) service presentation functionality being presented to the end user;

b) service implementation aspects with which the end user interacts.

3.22 service provider: A company that offers end-to-end telecommunication services (fixed/mobile, voice/data) to customers.

3.23 solution provider: A company that engineers and produces building blocks and solutions and sells them to service providers.

3.24 technical task: The functional work performed by a component.

3.25 technology provider: A company that develops component instances, which solution providers integrate into building blocks.

3.26 transport plane: The transport plane provides bidirectional or unidirectional transfer of user information from one location to another. It can also provide transfer of control and network management information.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

API	Application Programming Interface
BGP	Border Gateway Protocol
CGOE	Carrier Grade Open Environment
CIM	Common Information Model
COPS	Common Open Policy Service
CORBA	Common Object Request Broker Architecture
COTS	Commercial Off-The-Shelf

CSR	Customer Service Representative
DNS	Domain Name System
EJB	Enterprise JavaBeans
Enum	Enumeration
HA	High Availability
HTTP	HyperText Transfer Protocol
HW	Hardware
IP	Internet Protocol
ISV	Independent Service Vendor
IT	Information Technology
J2EE	Java 2 Platform Enterprise Edition
JDBC	Java DataBase Connectivity
JSP	Java Server Page
JVM	Java Virtual Machine
LDAP	Lightweight Directory Access Protocol
MTTF	Mean Time To Failure
ODBC	Object Database Connectivity
OS	Operating System
PDA	Personal Digital Assistant
PIN	Personal Identification Number
RASM	Reliability, Availability, Serviceability and Manageability
RDD	Relational Database Descriptor
RMI	Remote Method Invocation
RTP	Real Time Protocol
SAF	Service Availability Forum
SCTP	Stream Control Transmission Protocol
SIGTRAN	Signalling Transport
SIP	Session Initiation Protocol
SLA	Service Level Agreement
SMS	Short Message Service
UML	Unified Modelling Language
WAP	Wireless Application Protocol
WBEM	Web-Based Enterprise Management
WSDM	Web Services Distributed Management
XML	eXtensible Markup Language

5 The CGOE ecosystem

The CGOE open solution ecosystem extends to include the entire technology stack, ranging from hardware to applications. In this CGOE ecosystem three types of providers operate, viz:

- service providers who are responsible for delivering services to the end user subscribers;
- solution providers who are responsible for delivering solution building blocks to the service providers for the composition of services; and
- technology providers who are responsible for delivering functional components to the solution providers for the construction of solution building blocks.

The business environment for these providers is changing as follows:

- COTS technology is being adopted and deployed by enterprise customers and service providers in order to reduce the total cost of ownership of existing communication services and to rapidly develop and deliver new IP-based applications.
- The evolution of open industry standards allows solution providers to develop "mix and match" and "plug and play" multi-vendor COTS components that may be used by service providers to produce networks.
- The end-user subscriber is increasingly involved in provisioning and administering entitled services from the service providers.

While industry in general has recognized the advantages offered by COTS technology, service providers in particular will transition to COTS technology in their Next Generation Network (NGN) environment in stages. This transition will depend heavily on open industry standards and the availability of a rich COTS ecosystem. For the rich ecosystem to be useful to service providers it must also be of value to carriers and it can only be of value if it increases the number of COTS components. For COTS components to be useful for carriers, they must be carrier grade.

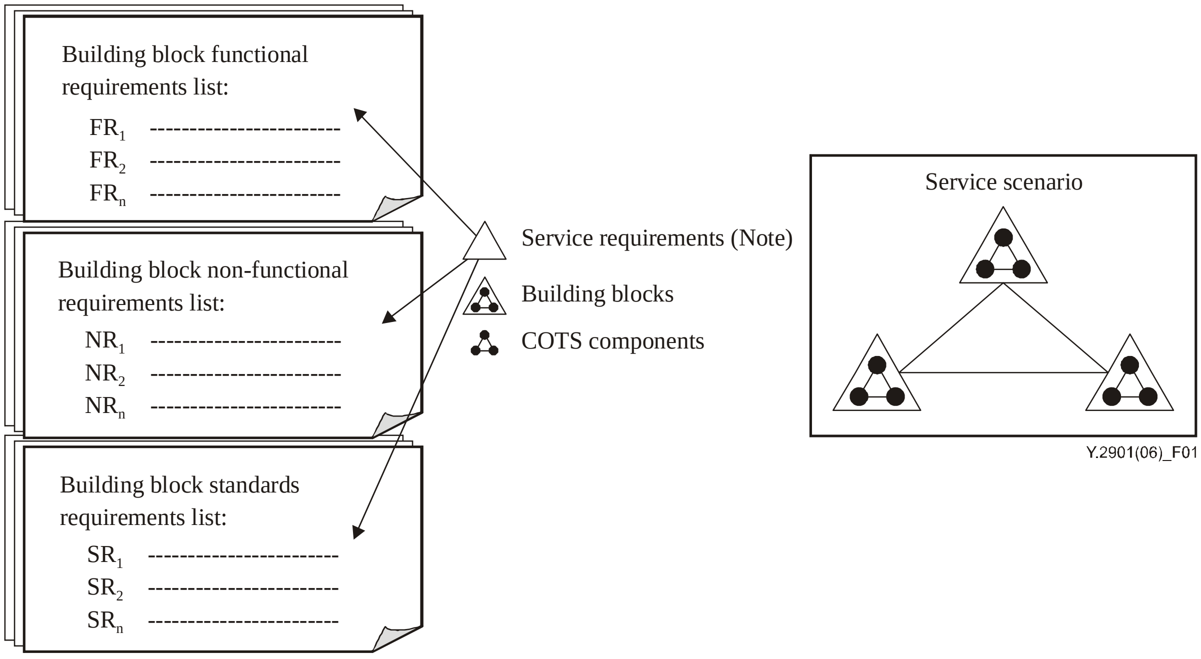
6 The COTS ecosystem

Service providers typically support a service development and runtime environment. The COTS ecosystem considered by this Recommendation is restricted to the runtime environment. This Recommendation describes how both CGOE and COTS components relate to the runtime environment. In particular, CGOE addresses the following areas of the runtime environment for the basic application model:

- networking, operating and HW platforms, storage;
- management interfaces and infrastructure;
- security and carrier grade functions; and
- transaction, session, and event management support functions.

7 CGOE general concepts

In the CGOE environment, NGN telecommunication services are enabled by one or more building blocks. Separate lists for functional requirements, non-functional requirements and standards requirements can be completed for each such building block associated with a service. Each completed set of lists represents the set of service requirements specific to that building block (see Figure 1).

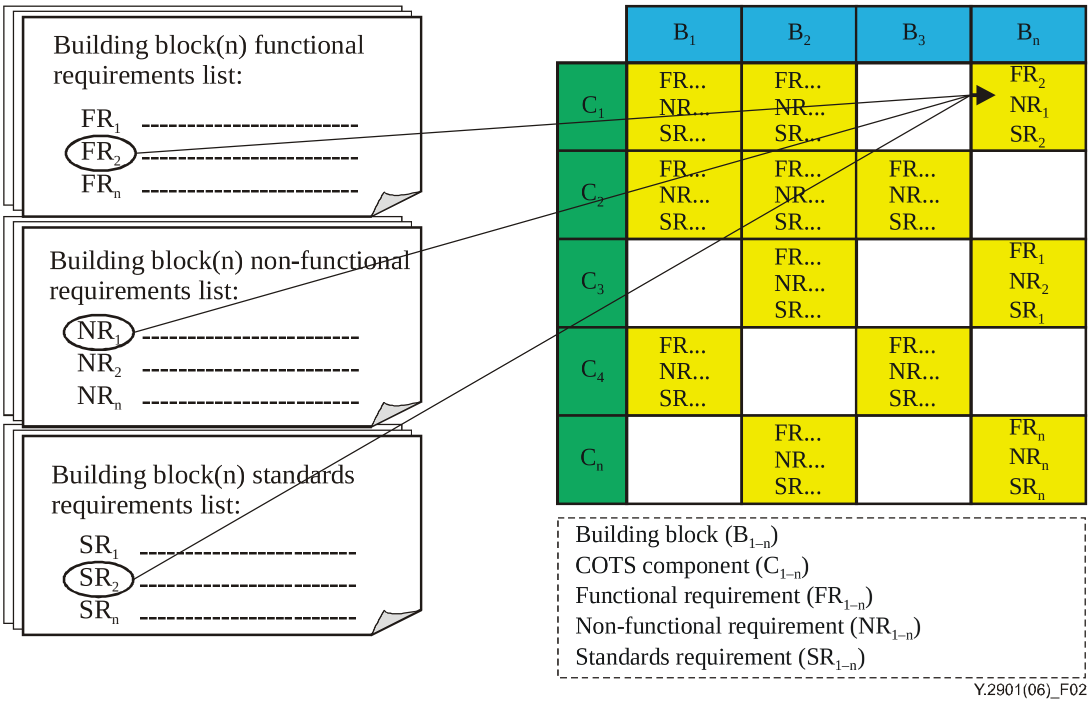

Building block functional requirements list:
FR1 ------------------------
FR2 ------------------------
FRn ------------------------
Building block non-functional requirements list:
NR1 ------------------------
NR2 ------------------------
NRn ------------------------
Building block standards requirements list:
SR1 ------------------------
SR2 ------------------------
SRn ------------------------
Service requirements (Note)
Building blocks
COTS components
Service scenario
Y.2901(06)_F01

NOTE – Includes functional, non-functional and standards requirements.

Figure 1 – Building block requirement lists

The mapping matrix for requirements of NGN service building blocks is as illustrated in Figure 2. Each column in the matrix represents a specific NGN service building block (B_n) and each row in the matrix represents a specific CGOE component (C_n). Each cell in the matrix represents a specific functional requirement (FR), a specific non-functional requirement (NR) and a specific standards requirement (SR) for a service building block that the identified CGOE component is expected to satisfy. The total number of service building blocks establishes the number of columns in the mapping matrix.

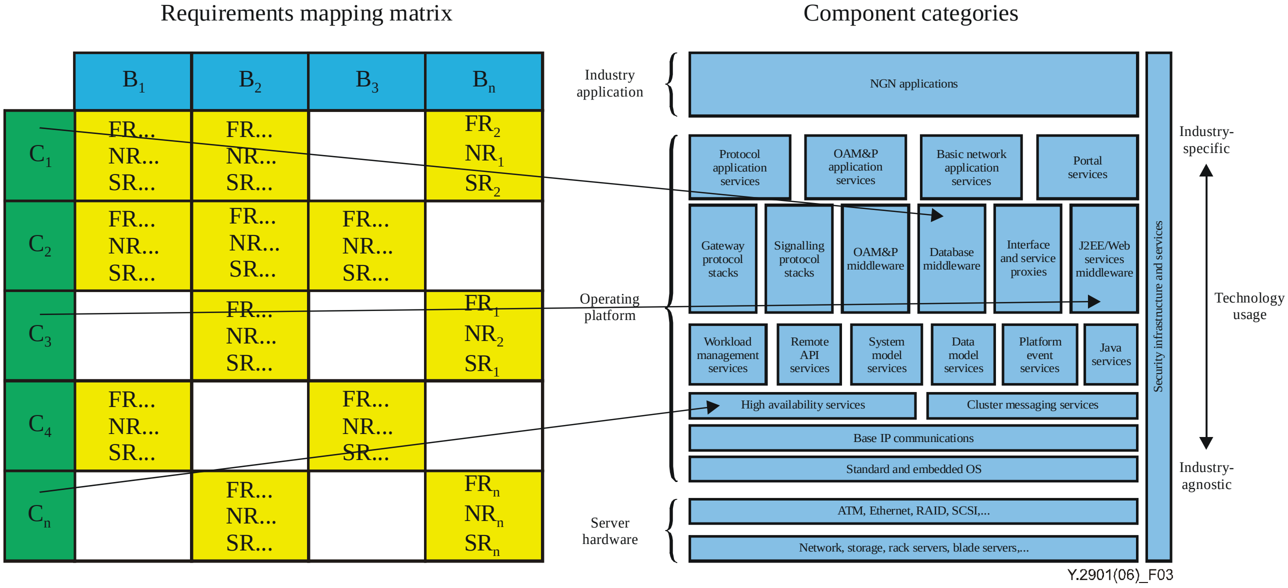
Note that when non-zero cells are present in a row, cells in that row may be identical. Should cells be identical, it implies that this particular CGOE component may be used to support more than one service building block.


Building block(n) functional requirements list:
FR_1 ------------------------
FR_2 ------------------------
FR_n ------------------------
Building block(n) non-functional requirements list:
NR_1 ------------------------
NR_2 ------------------------
NR_n ------------------------
Building block(n) standards requirements list:
SR_1 ------------------------
SR_2 ------------------------
SR_n ------------------------
Building block ($B_{1–n}$)
COTS component ($C_{1–n}$)
Functional requirement ($FR_{1–n}$)
Non-functional requirement ($NR_{1–n}$)
Standards requirement ($SR_{1–n}$)
Y.2901(06)_F02

	B_1	B_2	B_3	B_n
C_1	FR... NR... SR...	FR... NR... SR...		FR_2 NR_1 SR_2
C_2	FR... NR... SR...	FR... NR... SR...	FR... NR... SR...	
C_3		FR... NR... SR...		FR_1 NR_2 SR_1
C_4	FR... NR... SR...		FR... NR... SR...	
C_n		FR... NR... SR...		FR_n NR_n SR_n

Figure 2 – Requirements mapping matrix

CGOE components can be collected and organized into groupings called CGOE component categories for convenience of understanding and for identifying similarities between choices (see Figure 3). CGOE component categories were selected recognizing that support functions typically require aggregation of a number of independent CGOE components.


Requirements mapping matrix
Component categories
Industry application
Operating platform
Server hardware
NGN applications
Protocol application services
OAM&P application services
Basic network application services
Portal services
Gateway protocol stacks
Signalling protocol stacks
OAM&P middleware
Database middleware
Interface and service proxies
J2EE/Web services middleware
Workload management services
Remote API services
System model services
Data model services
Platform event services
Java services
High availability services
Cluster messaging services
Base IP communications
Standard and embedded OS
ATM, Ethernet, RAID, SCSI,...
Network, storage, rack servers, blade servers,...
Security infrastructure and services
Industry-specific
Technology usage
Industry-agnostic
Y.2901(06)_F03

	B_1	B_2	B_3	B_n
C_1	FR... NR... SR...	FR... NR... SR...		FR_2 NR_1 SR_2
C_2	FR... NR... SR...	FR... NR... SR...	FR... NR... SR...	
C_3		FR... NR... SR...		FR_1 NR_2 SR_1
C_4	FR... NR... SR...		FR... NR... SR...	
C_n		FR... NR... SR...		FR_n NR_n SR_n

Figure 3 – Relationship of CGOE components to CGOE categories

Each CGOE component in this open carrier-grade operating environment is critical to the operation and support of the NGN services. The CGOE component categories do not necessarily map to any autonomous vendor's COTS components. Rather, vendors' COTS components may be formed from one or more CGOE components and, if they contain CGOE components from different CGOE categories, may map to one or more CGOE categories according to their functionality. Additionally, technology providers who develop COTS components can use the set of CGOE components as a resource from which to select one or more CGOE components to implement into a specific COTS component.

The need to categorize the superset of all the CGOE components provides the basis of the CGOE reference model. The set of CGOE component categories identified for a specific building block provides the basis of a specific instantiation of the CGOE reference model. These categories are defined in the next clauses.

8 CGOE reference model

8.1 The NGN architecture

The CGOE reference model is aligned with the NGN architecture described in [ITU-T Y.2012]. This NGN architecture is shown pictorially as Figure 4.

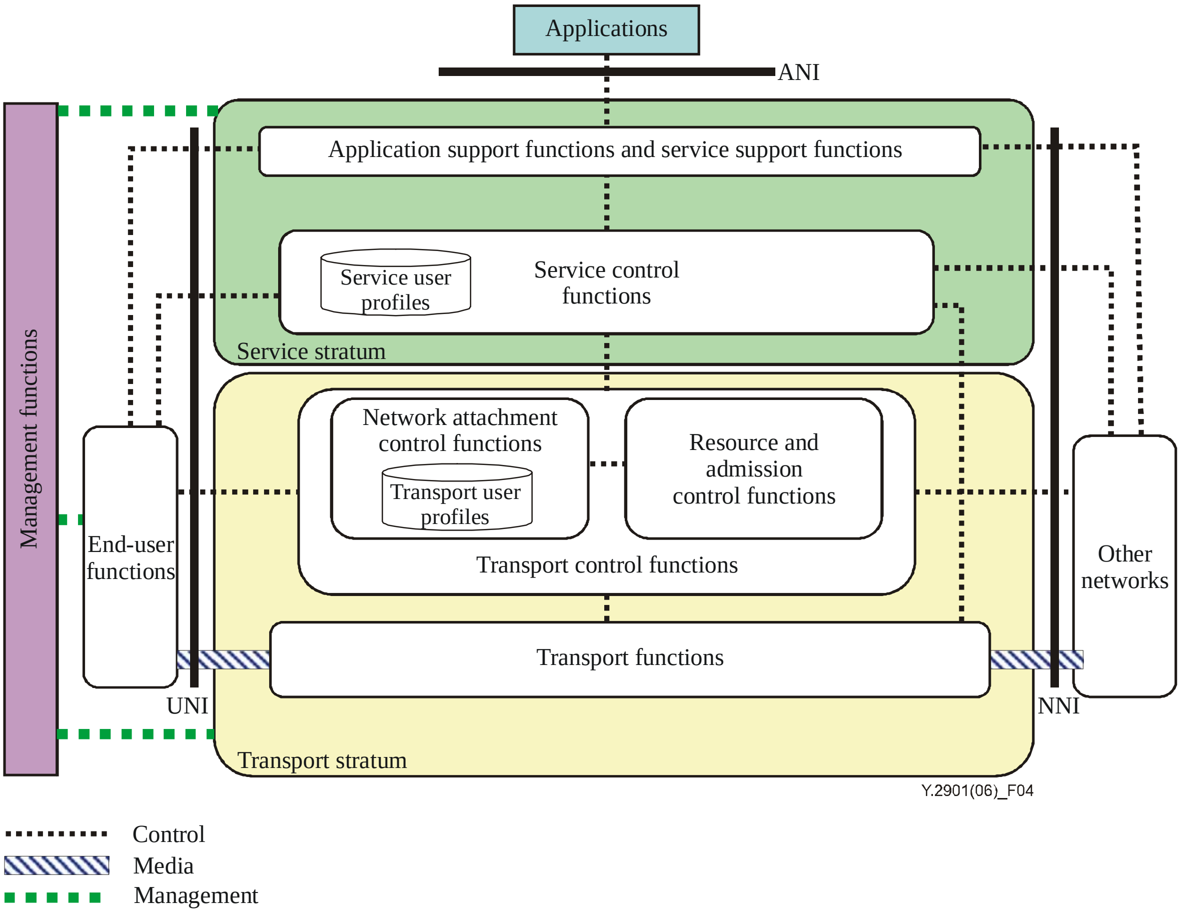


Figure 4 – Next Generation Networks – Architectural overview

The categories defined to organize the CGOE component set establish the CGOE reference model. Each category represents a different degree of standardization and is intended to be independent. To produce carrier grade functionality, functions may be needed from more than one category. Multiple categories are logically grouped and referred to as the server hardware and the operating platform. Detailed descriptions of the CGOE model and the various CGOE categories are presented in clause 9.

The CGOE reference model and the associated CGOE component categories are shown in Figure 5.

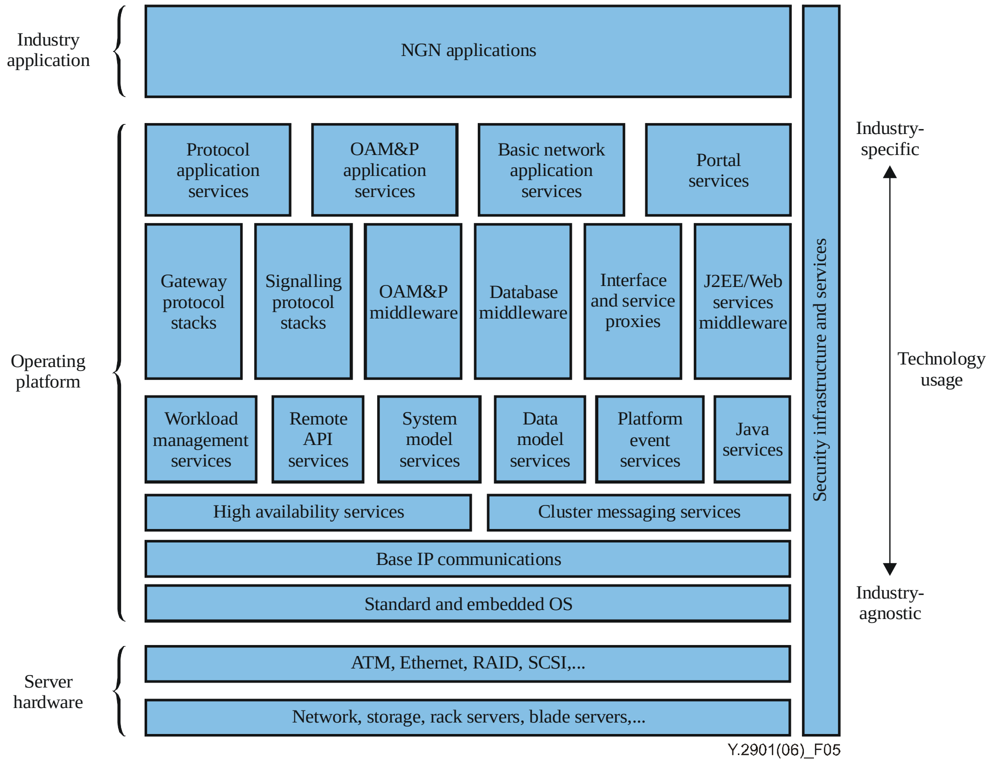

Industry application
NGN applications
Protocol application services
OAM&P application services
Basic network application services
Portal services
Gateway protocol stacks
Signalling protocol stacks
OAM&P middleware
Database middleware
Interface and service proxies
J2EE/Web services middleware
Workload management services
Remote API services
System model services
Data model services
Platform event services
Java services
High availability services
Cluster messaging services
Base IP communications
Standard and embedded OS
Operating platform
ATM, Ethernet, RAID, SCSI,...
Network, storage, rack servers, blade servers,...
Server hardware
Security infrastructure and services
Industry-specific
Technology usage
Industry-agnostic
Y.2901(06)_F05

Figure 5 – Carrier grade open environment reference model

The following clause describes each CGOE category and its role within the model.

9 Understanding the CGOE reference model

As stated earlier, the CGOE reference model identifies CGOE categories.

9.1 Industry application

This is the top of the CGOE reference model where solution-specific application logic is located. This Recommendation does not attempt to prescribe what applications to write and, of course, the possibilities are infinite. However, this CGOE reference model provides a framework whereby new COTS components can be developed to support new applications as the network evolves.

Network applications depicted in Figure 4 include control plane, management plane and service plane solutions. This model can be used for other strata as well, e.g., transport plane, access plane. Only an overview of some types of applications likely to be written can be provided.

Control plane applications

The control plane is used to establish connectivity between subscribers and network services. Typical control plane applications or solutions are products such as call processing servers, radio network access servers, etc. "Five to six nines" service availability and high scalability are strict requirements for control plane solutions.

Management plane applications

Carrier grade networks need to be highly manageable. They need to provide high levels of security to the user traffic. They also need to help minimize the cost of ownership. Network management applications such as network development, optimization, monitoring, reporting and administration are the prime applications targeted at the management plane. Service availability requirements are in the class of "five nines". Scalability requirements in the management plane are relatively modest when compared to the control plane requirements.

Service plane applications

Extra services on top of the connectivity network are created by the service plane applications. The number of applications requiring the services of an application server is on a steadily and rapidly increasing path. Solutions in current networks include but are not limited to: short message service (SMS) centre, WAP gateway, content delivery servers, serving mobile location centre, presence server, download server, web server access point and portals.

Independent service vendor (ISV) solutions run on the service plane. Service availability and scalability requirements vary a great deal in this plane. That depends on the solution area, business case and service level agreements (SLAs) currently in place, etc.

9.2 Operating platform

The operating platform may include databases, protocols, management, web services proxies, basic operating system with processor scheduling, I/O subsystems and interfaces to other hardware components that are used to enable the application and to extend the server hardware.

Abstraction of services is included for easier access and use by application developers. "Application services" also add value to the basic functions by combining and chaining the services with logic suitable for the application developers.

Services reside in this configuration for high availability and internal messaging capabilities, as well as components that enable scalability and manageability using carrier-grade enhanced OS (or standard OS for less restrictive service levels).

9.2.1 Protocol application services

This CGOE category provides added value for the application developers by hiding the complexity of multiple protocol stacks and by adding the abstraction of the protocol service access. Its functionality includes, but is not limited to:

- accessing the protocols for telecommunication purposes;
- combining primitives from several stacks to aid application development;
- providing an abstracted interface to protocols to aid application development; and
- enabling differentiation according to the specific NGN layer and application layer.

9.2.2 OAM&P application services

This CGOE category provides added value for the application developers by abstracting the OAM&P middleware to provide easy development of OAM&P applications.

It abstracts the services of OAM&P middleware across different OAM&P middleware implementations and links separate solutions with specific sets of application programming interfaces (APIs) based on the supported standard management models (e.g., CIM, WBEM, WSDM).

9.2.3 Basic networking application services

This CGOE category provides added value for the application developers by abstracting the platform to aid in the development of networking applications and by providing a set of generic networking services such as:

- accessing the database for networking purposes;
- supporting the state-machine based development paradigm;
- defining the network addressing models (e.g., ITU-T Rec. E.164/ENUM); and
- defining the network analysis models (e.g., address and route analysis).

These services abstract the specifics of the implementation of any facility put in place to serve a vendor-specific network application and allow a degree of plug and play ability to support the NGN network model. The goal here is to be able to switch with a fair degree of success and ease from one vendor's product to another vendor's product.

9.2.4 Portal services

This CGOE category provides a consistent, efficient and intuitive means to provide web-based end user interfaces for different end user equipments such as phones or PDAs.

Some outlined uses are:

- developing applets;
- developing portlets; and
- customizing web pages.

Portal services provide a rich set of functions and interfaces that will be available on all the devices that will access the services. For example, the services developers will only have to write the services once and portal services will ensure that the proper interface is presented to the user depending on the device that is being used to access the services.

Not only will the portal services provide a consistent access across all the services, they will also allow collaboration between users, providing a habitat or a virtual workplace where the features are utilized to their fullest potentiality and while providing a rewarding user experience.

CGOE encourages the interfacing of the portal services to other subscriber- and operator-facing services for the creation of user interfaces that provide better usability, improved intuitiveness and aesthetics that will ultimately result in tremendous savings of time and resources.

It is desirable that the development environment provides the ability to migrate existing applications to a portal environment that can be provided by CGOE adopters. This allows for ease and speed of the creation of new services and the extensions of existing services to a wider range of clients and devices.

9.2.5 Gateway protocol stacks

This CGOE category includes gateway protocols that are the payload counterpart of the signalling protocols, i.e., they are used for transferring user data. These protocols are such that they are naturally used in several network elements in a network infrastructure.

These include, for example:

- SCTP; and
- RTP.

These protocols are IP-based but non-IP based protocols can also be introduced if there are suitable interfaces in the hardware configuration. Protocols other than those listed above can also be used, but only protocol stacks that fulfil a specific application need should be chosen for use in the CGOE.

9.2.6 Signalling protocol stacks

This CGOE category includes the signalling protocol stacks that contain those carrier grade enhanced protocols that are needed for controlling services on the network. The protocols included in CGOE are such that they are used in several network elements in a network configuration.

These include, for example:

- SIGTRAN;
- SIP;
- Diameter;
- COPS;
- Megaco/ITU-T Rec. H.248; and
- BGP.

These protocols are IP-based but non-IP based protocols such as signalling system No. 7 can also be introduced if there are suitable interfaces in the hardware configuration. Protocols other than those listed above can also be used, but only protocol stacks that fulfil a specific application need should be chosen for use in the CGOE.

9.2.7 OAM&P middleware

This CGOE category of operation, administration, maintenance and provisioning (OAM&P) middleware provides the means to reach the objects defined by the system model via the system model services. OAM&P middleware features provide easy and efficient ways to create operable applications on the network elements. One of the prime objectives for an operator/service provided in this area is to make sure that all services are highly operational, i.e., service assurance[1] is achieved.

The OAM&P middleware is the external view of the system model and services. It externalizes the usage, performance and execution of the services.

For example, it provides (in many instances in collaboration with the portal services) the tools to create user interfaces as well as management and provisioning interfaces for activities such as:

- installation;
- configuration;
- update;
- monitoring;
- lifecycle management; and
- billing, accounting and charging.

The above activities occur in reference to the platform resources associated with service instances defined through the system model services category that is addressed in clause 9.2.15.

[1] Service assurance is another important context. It consists of the monitoring of the services. This means that the health of the service is monitored (errors, alerts, etc.), usage tests (use measurements, resource utilization, etc.) are run when customer service representatives (CSRs) report unusual customer exceptions in areas such as service usage (e.g., personal identification numbers (PINs) are not working/valid), and emergencies are handled when CSRs are not available (after normal working hours).

These tools include, for example:

- Aids for configuration management – Includes hardware and software configuration management, parameter management and bulk configuration aids.
- Aids for performance management – Includes tools for efficient monitoring and troubleshooting of network element performance from an operator's perspective.
- Aids for statistics collection – Entails information collection for monitoring of the network element performance from an operator's perspective. The information collected can be used to support the operation of the network element, analyse its operability and performance, and pinpoint possible problems and bottlenecks.
- The OAM&P middleware features are essential when driving the total cost of ownership down and when maintaining the serviceability of a system.

9.2.8 Database middleware

This CGOE category of database middleware provides the CGOE carrier grade environment for storing the data defined by the data model services. Several data management providers and solutions may be utilized depending on the needs of the server being defined on the platform. The database middleware may have to support *de facto* standards, e.g., ODBC, JDBC.

Both in-memory and disk databases, together with directories, can coexist on a platform implementing CGOE components. There can be a convergence from in-memory databases to disk databases if disk databases meet the performance requirements of specified applications.

9.2.9 Interface and service proxies

This CGOE category provides the core of the distributed programming model by brokering and mediating the APIs between clients and servers.

It provides an access point for application servers and applications by:

- distributing new sessions to appropriate registered application/application servers;
- managing session affinity between applications and services servers; and
- responding to managed shutdowns of application and services capability servers.

This CGOE category participates in the system start-up. It also defines the client/server responsibilities with respect to high availability, security and service discovery.

9.2.10 J2EE/web services middleware

This CGOE category refers to J2EE/web services middleware. J2EE (Java 2 Platform Enterprise Edition) provides CGOE with a simplified application development model and enjoys, if needed, the benefits of a thin client tiered environment. It also has built-in capabilities to interface with CORBA.

It includes, for example:

- W3C SOAP (simple object access protocol);
- UDDI (universal description, discovery and integration);
- OASIS SOA (service oriented architecture); and
- ISOC/IETF HTTP.

CGOE will require robust application servers with containers that are streamlined to be able to provide the highest qualities of service with the proper constructs for availability and serviceability. While Java and web services will play a dominant role in this space, other dominant (existing) interfaces, if open, may be included within this CGOE category.

9.2.11 High availability services

This CGOE category, high availability services, enables CGOE to provide highly available operating environments. Taken in the context of carrier grade, these are environments whose availability ratio can be as high as between 99.999% and 99.9999% (five to six nines) for a specific measure of time (typically one year).

The current approach to high availability is that it needs to be built into the system as well as the individual applications. To meet the highest availability requirements, applications need to be specifically designed for high availability.

High availability services are based on the philosophy of making software services reliable on the hardware and software platforms under the most adverse conditions. Redundant hardware resources are thus used, services are divided to active and standby instances or load-sharing groups, and application software actively contributes to availability. High availability services also provide supervision and recovery functionality for service instances.

High availability services provide the following:

- Starter service – Starts the application processes within a node. The operating system will start the high availability services when the node boots and high availability services will start all other processes.
- Graceful shutdown service – HA-aware applications are notified upon operator requests to shut down the system and are allowed to wrap up and terminate themselves "as gracefully as possible" within a certain period of time defined either by a platform configuration parameter or by the operator. All HA-aware applications should be prepared to handle this situation. This graceful shutdown service is provided in addition to the OS shutdown notification facility for a better synchronization and lifecycle management of all the resources belonging to the cluster.
- Supervision – A fault detection mechanism. Two types of supervision are performed: application supervision and node supervision.
- Communication support – Handles IP address recovery units and check pointing.
- System state management service – Maintains permanent and transient states for the managed objects it is aware of. Permanent state is persisted in permanent storage and is thus preserved in the event of cluster restarts. In contrast, transient state is lost in case of a system restart.
- Failure detection, isolation, recovery and repair services – These services supervise the nodes, the communication networks and the application processes. They are able to detect failure in these resources. They isolate nodes or recovery units in case of failure and initiate recovery either by using reconfiguration (software switch-over) or restarting. Change of faulty hardware can be done while the system is running on spare nodes.

CGOE takes advantage of the specifications for high availability services defined by the Service Availability Forum (SAF), where applicable. The interfaces and components in CGOE may be required to be SAF compliant where appropriate.

9.2.12 Cluster messaging services

This CGOE category, called cluster messaging services, must be reliable as it is often depended upon by high availability services. In order to successfully provide any of the described services, communication between the different systems or nodes and software running on the said nodes must be done reliably and without delays. A strong messaging subsystem will ensure that messages are delivered accurately to the destination nodes. Many possibilities exist for the implementation of a messaging subsystem, e.g., CORBA messaging, RMI, standard TCP/IP messaging, etc.

It can thus be said that a reliable, accurate and high performance messaging subsystem is at the heart of the high availability services.

9.2.13 Workload management services

This CGOE category, called workload management services in the CGOE reference model, is essential for carrier grade operations for the handling of overload situations and for balancing the load of a processing cluster. Thus, workload management services are divided into two groups of functions:

- Overload control functions that ensure the system does not lose its service availability even under an overload situation; and
- Load balancing functions that take care of IP level, transport level, and application level load balancing.

9.2.14 Remote API services

This CGOE category, called remote API services, provides CGOE with the means for distributed objects communication and execution. Remote API services provide location transparency, independence of interface definition from implementation, networking protocols, computing platform and programming language independence.

9.2.15 System model services

This CGOE category, called the system model services for CGOE, provides ways to access and use the managed objects of the system defined by the system model.

The system model services support activities such as:

- installation;
- configuration;
- updating;
- monitoring; and
- lifecycle management.

The above activities occur in reference to the platform resources associated with service instances defined through the CGOE system model.

This portion of CGOE provides the means for easier applications management (i.e., non-disruptive installation and update). It provides a scalable management model and allows for integration of existing management solutions. The OAM&P artifacts can administrate the system model objects contained in the system model portion.

9.2.16 Data model services

This CGOE category, called data model services, provides ways to access and use data schemas required by the system. The aim is to provide an abstraction level that allows the interoperation and substitution of different data management solutions on the platform.

This includes, for example:

- system model data schemas;
- provisioning data schemas;
- configuration data schemas;
- system API definitions; and
- functions to update, create, and delete the schemas.

The schemas can be defined with XML, RDD, LDAP and other suitable tools.

9.2.17 Platform event services

This CGOE category, called platform event services, provides services for managing events and logs.

9.2.18 Java services

This CGOE category is used to refer to basic Java runtime services that may be used when full J2EE web services would not be needed.

It includes, for example:

- runtimes and APIs for running applets and applications written with Java; and
- essential compiler, tools for writing, deploying and running applets and applications in the Java programming language.

The CGOE advocates the use of a common Java virtual machine for both J2EE and Java stand-alone applications and services. Java is depicted in Figure 5 to show that, in some server deployments, a full J2EE server may not be required but in fact may be installed to provide the common JVM for stand-alone use.

9.2.19 Operating system

The operating system is the software foundation upon which the rest of CGOE operating platform is built. As such, it must provide the solid basis that is extended by components of the other CGOE components. The criterion for selecting the operating system for CGOE has much to do with the "openness" of the operating system. Such selection, as a result, may cause some stringent requirements to be placed on that specific OS for supporting specific open standards such as POSIX [b-ISO/IEC 9945-1].

In addition to openness, CGOE requires the use of any necessary carrier grade OS capabilities.

9.2.20 Base IP communications

This CGOE category refers to the base IP communications protocols used for CGOE and which are typically expected to reside within the operating system. For the CGOE, significant examples of these protocols are: TCP, UDP, IPv4, IPv6, IPSec, PPP, ARP, FTP, DNS, NFS, RTP, RTCP, SSL. Additional protocols that may be included within this category are HTTP, SMTP, and SNMP, their use depending upon the OS chosen for the CGOE.

9.3 Server hardware

CGOE category that may be used in several industries with minimal modification and includes physical hardware and the drivers needed to use that hardware. While CGOE is hardware-agnostic, some requirements need to be met by both hardware and drivers to enable carrier grade operation.

9.3.1 Drivers

Drivers are essential to CGOE for enabling support of plug and play carrier grade hardware and software components. The vendors of hardware components are expected to make drivers available for their components.

9.3.2 Hardware

CGOE covers a variety of hardware platforms, given the capabilities to meet the necessary carrier grade requirements exist on the selected platform. It must be recognized that some hardware capabilities can be modified or waived in situations where a solution has service level agreements that can tolerate their absence.

10 Security considerations

The CGOE reference model makes provisions for security. However, the specification of specific security requirements is outside the scope of this Recommendation.

Appendix I

CGOE principles

(This appendix does not form an integral part of this Recommendation)

The CGOE principles are presented in Table I.1. The purpose of CGOE principles is to provide information that may be valuable to the user of this Recommendation. They provide additional information about topics covered within this Recommendation, but are not essential for an understanding of this Recommendation.

For example, the principle of open standard interfaces requires the use of industry standard interfaces for externally exposed functionality. By aligning with this principle, the product intends to implement open standard interfaces as a feature of the product.

Table I.1 – CGOE principles

Number	CGOE principle/tenet	Criteria
1	Open component interfaces	External open standards interfaces used or provided by a component.
2	Modular	Implementation of functionality is componentized to a single purpose.
3	Scalable	Ability to provide appropriate service in small and large configurations, low and high loads, local and distributed configurations.
4	Reliable	Component is designed and implemented to have very low mean time to failure (MTTF). Component behaviour is predictable, repeatable and measurable.
5	Available	Ability to detect and recover from hardware, software or operator faults before the service being delivered is affected.
6	Manageable	Ability to monitor and control a function to ensure the appropriate operation of a system with respect to configuration, state, diagnostics and performance.
7	Secure	Ability to authenticate presence, access and output of component.
8	Maintainable	Ability to test, validate and upgrade component with no impact to system operations.

Bibliography

[b-ISO/IEC 9945-1] ISO/IEC 9945-1:2003, *Information Technology – Portable Operating System Interface (POSIX) – Part 1: Base definitions*.

SERIES OF ITU-T RECOMMENDATIONS

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

Printed in Switzerland
Geneva, 2007