



University of East London Institutional Repository: <http://roar.uel.ac.uk>

This paper is made available online in accordance with publisher policies. Please scroll down to view the document itself. Please refer to the repository record for this item and our policy information available from the repository home page for further information.

Author(s): Lago, Patricia; Falcarin, Paolo; Andreetto, Alessandra; Licciardi, Carlo Alberto

Article title: Next Generation Networks: the service offering standpoint

Year of publication: 2001

Citation: Lago, P. et al. (2001) "Next Generation Networks: the service offering standpoint" In: Proceedings della 7a IEEE-ITU International Conference on Intelligence in Networks (ICIN'2001), Bordeaux, France, Oct. 2001, pp. 256-261.

Next Generation Networks: the service offering standpoint

Paolo Falcarin, Patricia Lago

{ [paolo.falcarin](mailto:paolo.falcarin@polito.it), [patricia.lago](mailto:patricia.lago@polito.it) }@polito.it
Politecnico di Torino
Dipartimento di Automatica e Informatica
Corso Duca degli Abruzzi, 24
I-10129 Torino, Italy

Alessandra Andreetto, Carlo Alberto Licciardi*

{ [alessandra.andreetto](mailto:alessandra.andreetto@tillab.com), [carlo.licciardi](mailto:carlo.licciardi@tillab.com) }@TILab.com
Telecom Italia Lab
Network Intelligence Dept.
Via R. Romoli, 274
I-10148 Torino, Italy

1. Introduction

Next Generation Networks (NGN, [1]) was the buzzword last year. Different people imagine different things when they think of NGN. How can this concept be defined?

NGN is the application of Internet, IP and IT solutions to Telecom Services, including (but not only) the integration and sometimes the substitution of circuit switching with packet switching either for trunking or for access.

Surprisingly, usually people think of a mere porting of Telecom protocols over an IP network (read H.323 protocol). Why should we reuse existing telecom solutions over an IP transport? The usual answer is to reduce cost of ownership. Is that really true? Is that a reason enough?

Indeed, the introduction of NGN opens a huge opportunity for incumbent telecom operators: enabling the renewal in the service offering (meaning cash!)

In this perspective, NGNs should enable the provisioning of new classes of services:

- ❑ any-to-any ubiquitous communication services including unified messaging
- ❑ customer-centered and highly personalized services
- ❑ mixed voice/data services
- ❑ e-call services (web initiated call set-up)
- ❑ integrated voice and data VPN
- ❑ advanced network call center
- ❑ audio video conferencing

A common feature for all these service classes is seamless service access (users can use their own services, no matter where they are, which terminal they are using, which access network they are attached to).

In this paper, we will present the objectives and results of the Eurescom Project P1109 [1]. The overall goal is to support this view, in evaluating solutions for NGNs from a service-offering standpoint and understanding the wider effects of introducing NGNs in terms of the inter-operability and functionality of NGN products.

Basically, the project aims at answering the following questions: Why NGNs? Are NGNs programmable? Do NGNs solve the problems of IN? Which NGN solutions? Are NGN interoperable? Are NGNs cost effective? Are NGNs service developed? Is Application Server just a new SCP?

In particular, by selecting a significant pool of products approaching the market in the area of Media Gateway/Media Gateway Controllers, Call Agents, SoftSwitches, Application Servers, and Application Creation Environments, the project has developed service scenarios through lab trials on top of the selected products. Based on this evaluation the project will also study and define migration scenarios from existing network infrastructure towards establishing NGNs and services.

The involvement of selected vendors as Technology Partners has been a key issue to evaluate NGN products by means of prototyping new services. Some interoperability tests have been carried out, in particular to understand how different Network Elements and applications interconnect with each other including with legacy systems and services.

In particular, this paper focuses on service programmability of NGN solutions, i.e. to verify the *attitude* of NGN Service Platforms to provide innovative services over heterogeneous networks in *developer friendly* way. Section 2 describes the overall reference architecture, sections 3 defines the key technologies, which can be used for service development. Section 4 describes a service example (advanced VPN), which has been implemented to investigate the identified technologies.

2. Architectural issues and objectives

In order to better identify the scope of NGN platforms we have defined a high level architectural view, which is depicted in Figure 1.

The aims of this architecture are to identify the main elements, which characterize NGN platforms in terms of functionalities and interfaces, and to define a common terminology to identify different entities.

* Contact address for correspondence.

Application Creation Environment supports the life-cycle of a service or an application, which can consist of a series of phases, namely:

- service analysis and conception,
- application creation,
- acceptance testing,
- application deployment,
- application provisioning and operations,
- application removal.

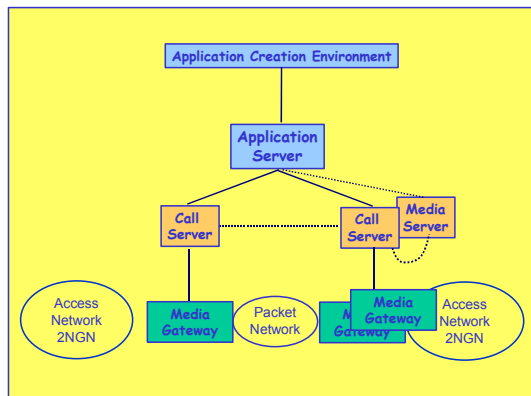


Figure 1. Reference Architecture

In an NGN context, the concept of **Application Server** should be an evolution of Web-based application servers, able to execute services controlling Call Servers and Next-generation Special Resources. Application Servers should make up a sort of IT platform broadening the role of IN-SCF to cover new network/service scenarios. The main functionality that an AS should support are: Service Logic Execution Environment (SLEE), service life-cycle management, support for developing services/policies by means of APIs or scripting languages (e.g. CPL, VoiceXML), system and service management, registration mechanisms support (including SIP registrar or H.323 registration request).

Call Server mainly provides call control functionality (call routing, call signaling process – SIP [8][10], H.323, SS7, Megaco/H.248 – including signaling gateway capability, third party call setup, static/dynamic trigger activation/deactivation, static/dynamic event subscription activation/deactivation) according to a given call model. It must provide also an interface (i.e. standard protocol or open API) towards Application Servers to enable service and policy control. It can also include QoS control for the media flow¹. Call Agents, SoftSwitches, Media Gateway Controllers are some examples of Call Servers. Multiple Call Servers might co-operate in order to handle a single call.

Media Server encompasses features for Media Resources control (such as IVR, Text to Speech and Speech Recognition devices, messaging server). Usually they

are accessible by Application by means of VoiceXML interfaces.

Media Gateway functions provide conversion between circuit-switched resources (line, trunks) and the packet network (IP, ATM). They can also support mechanisms for QoS enforcement.

Access Network to NGN in Figure 1 represents the different ways to access the services offered by an NGN platform (i.e. PSTN, ISDN, xDSL, ...) while packet network usually stands for IP/ATM backbone.

NGN products will be analyzed based on available documentation. These products will be classified according to their functionality in order to better define requirements for the architecture. RFI (Request for Information) will be issued to vendors in order to identify the best development environment for service creation.

3. Analysis of Technologies

In an open customization environment for Internet Telephony, various emerging approaches can be used to define new services. Some examples are XHTML, CPL, VoiceXML, J2EE, SIP Servlet and SIP CGI.

XTML: the Extensible Telephony Markup Language [15] is an XML-based service description language and associated service execution framework. It has been designed to provide a unified approach for the delivery of enhanced telecommunication services. XHTML clearly defines and separates descriptive aspects of services (that are relatively static and common to all network-based personal communication services), from the more dynamic aspects that correspond to specific technologies, protocols, or application domains. In particular, the language natively supports the integration with standard technologies like CORBA, EJB and DCOM by offering constructs to embed external invocations.

XTML provides core description of basic service logic, to process incoming network events or to redirect them to linked services. Strong points of XHTML are: the reduced time-to-market, portability and integration of multi-vendor solutions. On the other hand, XHTML cannot be considered as an alternative to VoiceXML or CPL: e.g. XHTML cannot support IVR-like functionality. At last, XHTML is still a proprietary technology, even though it has been proposed as standard to W3C.

CPL: Call Processing Language [16] is a simple, static language to describe how Internet Telephony call invitations should be processed, hence permitting the creation of simple call control service logics. In particular, CPL is tailored for SIP (even if interaction with H.323 is also provided) and it is lightweight, efficient and easy to implement and to parse (being XML-based). On the other hand, it is quite difficult to integrate CPL services with other software components; it does not provide PSTN user interaction features (e.g. play-announcement, play-announcement-collect-digits) and it is hardly extensible. Comparing XHTML with CPL, the

¹ Enforced to other servers – media gateways or routers

first is more oriented to service development while the second better adapts to service customization.

VoiceXML: VoiceXML [19] is an XML-based language for creating voice-based user interfaces, in particular for telephony. It uses speech recognition and touchtone (DTMF keypad) for input, and pre-recorded audio and text-to-speech synthesis (TTS) for output. VoiceXML aims at giving developers full control over the *spoken dialog* between user and application: the application prompts the user, which in turn responds. VoiceXML has features to control audio output, audio input, presentation logics, event handling, and basic telephony connections. In short, it implements voice-browsing among a fixed number of options: the service logics is stored in a document server (e.g. web server) and it is fetched by the VoiceXML server and interpreted in order to create a TTS output, depending on user input. This consists of an input text string generated by the VoiceXML after a process of speech recognition made on vocal input coming from the user's phone.

EJB/J2EE: EJB (Enterprise Java Beans [17]) and its supporting platform J2EE™ (Java 2 Enterprise Edition [18]) are a widespread approach for web-application development. They can be used to create new services based on Java technology and J2EE API. This is made of different parts like: Java Messaging Service (JMS) for messaging; Java Authentication and Authorization Service (JAAS) to integrate user access control into services; JAXP (Java API for XML Processing) to enable applications to parse and transform XML documents, etc.

Other powerful features integrated in J2EE are servlet and JSP (Java Server Pages) that represent respectively the server and client side for Java-based web applications. These two features can be used to directly implement service logics, or as a front end to a server side implementation composed of EJB containers executing a group of Java Beans.

A drawback of the EJB architectural model (being in Java) is that byte code interpretation reduces performance, and proprietary solutions for RMI/IIOP and other features (like load balancing and traffic monitoring), affect its portability.

Concerning SIP support, application servers (like J2EE) do not conform yet to any standard. Nonetheless, current efforts include the JAIN proposal (for SIP API support) and emerging technologies like SIP Servlets and SIP CGIs (for the underlying architectural model).

SIP Servlet: SIP Servlets [13] provide a high-level extension API for SIP servers, hence enabling SIP applications to be deployed and managed according to the common servlet model. In other words, through SIP servlets, SIP servers' functionality can be extended with application-specific logics. An important difference with generic HTTP servlets is that SIP servlets execute on SIP servers, too, hence representing a natural extension of SIP networks. Servlets can be implemented in Java only.

SIP CGI: Similar to SIP servlets, also SIP CGIs [14] implement application-specific logics extending SIP servers functionality, and naturally integrate Web applications into services. Differently, SIP CGIs support multiple languages, but being executed in a separate process, each invocation requires to start a new process, with a consequent difficulty in state and information sharing. At last, CGIs in general require a more complicated implementation.

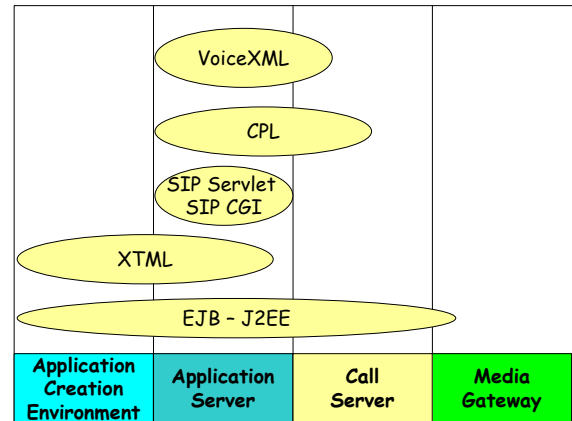


Figure 2. Technologies vs. Reference Architecture

In summary, Figure 2 shows how each analyzed technology can be applied to our Reference Architecture.

We can observe that all technologies cover the Application Server level. In details, EJB/J2EE yields to the broader applicability: being a middleware supplying a generic execution environment, it covers all programmable architectural levels (from Application Creation Environment to Call Server) and can further involve the Media Gateway level by encapsulating gateway technologies into Java ones. Other technologies represent languages on various abstraction levels, according to the type of service development they support, the type of users, and their skill.

For instance, while XTML is oriented toward expert developers in charge of NGN programming, CPL is more suitable for smart users in charge of customizing call routing policies.

Further, CPL supports call control programmability at the Call Server level. On the contrary, VoiceXML mainly provides voice- and text-based user interaction at the Application Server level, even though it also covers the Call Server level (or better the Media Server level – see Figure 1) by integrating media conversion functionality.

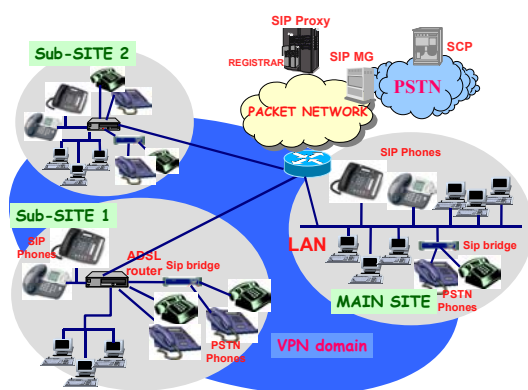
At last, SIP Servlets and SIP CGIs implement SIP components on the Application Server level.

4. Service Scenario

This section describes a service scenario challenging NGN programmability: an advanced VPN (i.e. based on a converged network like POTS/IP) for SOHO. Further,

we involve personalization features (e.g. virtual presence [2]) including closed user groups management, role management, incoming and outgoing call screening, etc. Advanced features based on the integration of voice and data communication capabilities could also be considered like invitations to application sessions (e.g. joint editing), directory services (e.g. personal address books, click-to-dial) and push services.

Once we provide that, our customers would like to place and receive also off-net calls (to/from PSTN).



Entities within an Enterprise environment are suitable to be easily modeled as a group hierarchy (i.e. a tree structure) composed of groups (as root), subgroups (as intermediate nodes) and roles and members (as leaves).

Also, the service scenario evidences the need for a personalized user profile to maintain customizable policies based on both user- and group-based preferences.

4.1.1. User Perspective

In particular, the service scenario supposed that the administrator defined Enterprise information, as summarized by the following table:

In the table, columns represent respectively: employees (users) involved in the scenario, groups within the Enterprise and in which users play a role, their group role, and the policies defined for each user.

John is a VPN-user using as terminals his PC, a SIP phone and a mobile phone. His profile allows him to:

Susan is John's secretary, and is a VPN-user using as terminals a PC and a fixed telephone. Her profile is configured as follows:

Mark is a VPN-user using as terminals a PC and a SIP phone. His profile permits him to place calls within the VPN only. It means that he can reach any VPN-user (even if the last one is outside) but outgoing call screening feature (next section) doesn't enable him to dial public PSTN numbers.

At last all VPN-users (i.e. John, Susan and Mark) are entitled to customize their own policies for incoming call screening. On the contrary, Paul is only entitled to re-

ceive incoming calls upon valid registration, and he cannot change his profile information.

After having described roles and users/groups policies, the following considers an execution scenario.

Assume that John wants to communicate with Susan: if John uses a PC, he has to call Susan directly by sending an invitation to *uid: Susan@ManagementStaff*.

Unfortunately, Susan is on holidays this week and John urgently needs some secretarial work. In order to contact a colleague of Susan with the same skills, John has to communicate with a Secretary (role) belonging to the Management Staff (group). To find out the address of such person, he can browse the enterprise Web site:

- if John uses a phone, he has to dial a personal number, which identifies a particular role within a group in the enterprise (e.g. *xxxzzzz* where *xxx* represents a call towards a personal number, and *zzzz* is the personal number itself);
- if John uses a client application on his PC, he has to call any Secretary that belongs to the same group of Susan. In this case he has to send an invitation to *Secretary@ManagementStaff*.

Afterwards, John can communicate with another secretary (not represented within the table, for simplicity).

To support the execution scenario just described, an NGN platform should support a set of modular service features that can be composed (e.g. by use of technologies presented in Section 3) to define NGN services. In particular, service features include the possibility to manipulate profile information for both groups, roles and users, and the correct interpretation of such information whenever a call is issued from/to any type of terminal. Further, for personalization support, service features must support the definition of policies associated to both individual users, and generic groups/roles. Selected service features to cover above requirements are presented in the following.

4.1.2. Service Features

This section describes the main features for the enhanced VPN service scenario. Policies are based on either users' role or their membership to a group.

➤ **Abbreviated Dialing**

It allows VPN-users to dial a short number to localize another VPN-user.

➤ **Incoming Call Screening**

The ability for VPN-users to define their policies in order to be reached according to events (i.e. busy) and conditions (time, caller identity, etc). In other words, this is the way to perform Call Filtering applied to calls received by VPN-users.

➤ **Outgoing call screening**

Call filtering applied to calls placed by VPN-users. For example, according to the table, Paul cannot perform outgoing calls.

➤ **On-Net Calls**

The called user is a member of the VPN: the "dialed number" could be a logical name/number. Different sub scenarios arise from different originating and terminating terminal type (IP or usual phones).

➤ **Off-Net Calls**

The called user is not a member of the VPN: "the dialed number" is a PSTN number prefixed with a specific code (e.g. "0"). For example, according to the table, Mark cannot place calls towards the PSNT.

➤ **Group Management**

The ability to:

- ☐ create and remove new groups
- ☐ add and remove roles to/from a group
- ☐ add and remove users to/from a group
- ☐ define group-level personalization policies (e.g. a policy establishing that users belonging to the Visitors group cannot communicate with users within the Management Staff group)

➤ **Role Management**

The ability to:

- ☐ create and remove new roles
- ☐ set which members cover a role

➤ **User Management**

The ability to associate a user/member to a role within a group, and for each user, to state personal policies.

4.1.3. Service Logic

This section describes an example of service logic applicable to the service scenario discussed through Section 4. Obviously, this is not an exhaustive example.

If John, registered on an IP terminal, calls Susan, the logic must verify which kind of terminal Mark is registered on, and then translates his logic number/name to either the actual IP address or PSTN number of Susan's terminal. In the latter case, the logic should also perform checks in order to verify:

- ☐ if John is authorized to call Susan (i.e., since they belong to the same group it's sensible to think that they are allowed to communicate);
- ☐ if John is authorized to perform external PSTN calls (different call classes could be defined: international calls, domestic call, urban calls, etc.).

The logic must also verify policies set by Susan to deal with her incoming calls (incoming call screening service feature – e.g. route all incoming calls to the IP phone).

A question follows spontaneous: how this service logic could be implemented?

Different technologies, amongst those analyzed in Section 3, are suitable for different aims. Data collected within the users profiles can be defined by using XML language.

Service logics could be described quite easily by using languages that belongs to the family of XML-based languages. Both XHTML and CPL can be used even though XHTML seems to be better due to its extensibility.

More complex service logics could be described by Java-based technologies like EJB/J2EE or even SIP-based technologies.

5. Conclusions

This paper highlights benefits for Network Operators in adoption of NGN with respect to existing solutions. It shows how NextGen Service Platforms can provide advanced services and the benefits, which come from using SIP protocol and programmable Application Server in terms of easy-to-use, flexibility and programmability.

The implementation of the service scenario described in section 4, on top of the identified architecture & technologies, has shown:

1. the architecture can be deployed: products which implement the network elements identified in fig. 1 are available on the market
2. XML-based languages are suitable to describe user profiles and personal service policies
3. XTML and CPL are suitable languages to describe services; the former is extensible and can be used by the service provider, the latter is more limited and suitable mainly for service customization
4. service portability over different platforms is enabled by the use EJB/J2EE technologies.

Looking at the bad side, interoperability among several solutions is still an issue: proliferation of optional features in the open standards (SIP, JAIN, ...) and vendor-specific implementations sometimes prevent solutions from being interoperable.

In conclusion, NGN service platforms ease the provisioning of advanced added value services, which span over heterogeneous networks (packet and circuit switched). NGN have to be seen by incumbent operators as revenue generators (new services) and not as cost saving.

Further work is needed to deeper analyze migration strategies from existing architecture (such as IN) towards NextGen Platforms. At first sight SIP seems to be a suitable protocol to pursue this objective.

Acknowledgements

The information presented in this paper is partially based on the work done in the EURESCOM project P1109 "Next Generation Networks: the service offering standpoint" [1]. The authors would like to thank all EURESCOM P1109 participants.

6. References

- [1] Eurescom Project P1109 *Next Generation Networks: The services offering standpoint*. On-line at <http://www.eurescom.de/secure/projects/P1100-series/P1109/P1109.htm>
- [2] P. Lago, C.A. Licciardi, G. Canal, A. Andreetto, *An architecture for IN-Internet hybrid services*. In Computer Networks Journal, Special Issue on Intelligent Networks and Internet Convergence, T. Magedanz, H. Rudin, I. Akyildiz (Eds.), Elsevier, Vol. 35(5), April 2001, pp. 537-549
- [3] M. Barry et al., *What an IN system should be*, EURESCOM P909 Deliverable 1, <http://www.eurescom.de/Public/Publications/Projectresults/P909-series/909d1.htm>, October 1999.
- [4] Schulzrinne, H., Rosenberg, J., *The IETF Internet Telephony Architecture and Protocols*, IEEE Network, May/June 1999.
- [5] Andreetto, A., Canal, G., Lago, P., Licciardi, C.A., *Ubiquitous communication over IN and Internet*, In Proceedings of the 6th ITU International Conference on Intelligence in Networks (ICIN '2000), Bordeaux, France, pages 35-40, January 2000.
- [6] Blavette, V., et al., *Architecture and service scenarios for Internet-IN convergence*, EURESCOM P909 Deliverable 2, <http://www.eurescom.de/public/projectresults/P900-series/909d2.htm>, March 2000.
- [7] Cuda, A., Canal, G., *From a proxy to "the" Next Gen Network Element*, Proceedings SIP2001 conference.
- [8] Rosenberg, J., *Architecting SIP Networks for ITSPs*, Proceedings SIP2001 conference.
- [9] Canal, G., Cuda, A., Lago, P., *IN-like Services Enabled by Open SIP System*, Proceedings SIP2000 conference
- [10] Canal, G., Cuda, A., *Why SIP will Pave the way towards NGN*, In Proceedings of the 7th ITU International Conference on Intelligence in Networks (ICIN '2001), Bordeaux, France, October 2001.
- [11] Handley, M., Schulzrinne, H., Schooler, E. and Rosenberg, J., *SIP: Session Initiation Protocol*, RFC 2543, March 1999.
- [12] Boetselaars, L., et al., *CD-ROM: Enabling technologies for IN - Internet integration*, EURESCOM P909 Deliverable 4, January 2001.
- [13] *The JAIN APLs: Integrated Network APIs for the Java Platform*. (White Paper), Jun. 2000. On-line at <http://java.sun.com/products/jain>
- [14] Lennox, J., et al. *Common Gateway Interface for SIP*, RFC 3050, Jan. 2001.
- [15] XTML specification. On-line at <http://www.pactolus.com/pcs-xtml.pdf>
- [16] Lennox, J., Schulzrinne, H., *CPL: A Language for User Control of Internet Telephony Services*, Internet Draft, Nov. 2000. On-line at <http://www.ietf.org>
- [17] *Enterprise JavaBeans™ Specification*, Version 2.0. On-line at <http://java.sun.com/products/ejb>
- [18] *Java™ 2 Platform, Enterprise Edition Specification Version 1.3*. On-line at <http://java.sun.com/j2ee/docs.html>
- [19] *VoiceXML specification*. On-line at <http://www.voicexml.org>