http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 7 (5): 715-727, 2008
ISSN 1812-5638

A Novel Practical Service Delivery Platform for Next Generation Networks

A.H. Darvishan, F. Sarabchi and H. Yegane
Iran Telecommunication Research Center, Tehran, Iran

Abstract: In this research, model for service layer creation for providing various multimedia services in next generation networks is introduced. In this model, open standards and commercial software frame works have been utilized so that model implantation is completely practical. Also, a new architecture based on TISPAN model explains participation of 3rd party operators in creation and implementation of services on an integrated telecom network.

Key words: Service model, NGN, SDP, API, JAIN, parlay, web service

INTRODUCTION

Nowadays telecommunication industry is bearing important changes so that new challenges and opportunities for network operators can be created. Network operators are facing new concepts of architecture and methods in order to provide services and these methods happen to be independent of network infrastructure. In today's it and telecom world, architecture and network structure is dependent on service and is formed based on service requirements (Eurescom Project P1109, 2001).

Unavoidable SDP concept which has been originated from service convergence in telecom, intended to enable rapid development and deployment of new converged multimedia services, from basic POTS phone services to complex audio/video conferencing for multiplayer games. SDP with the help of a standardized frame work creates an environment for implementation and management of different services (Silva *et al*., 2006). SDPs typically provide a service control environment, a service creation environment and execution environment and abstractions for media control, presence/location, integration and other low-level communications capabilities. In SDP, layers, blocks, interfaces, software and hardware resources and how they can be accessed and orientations are defined.

On the other hand, a legacy telecom network consists of different access networks and each network has its own interface, standards, control and service layer. In some of these networks there is no separation between layers. These unorganized interworking and network discontinuity has caused many problems for operators and network users. From operators' point of view, scalability will be difficult and network expansion will be costly. From users' point of view, network services will only be available to those who are the users of the network and there is no possibility of utilizing other network services. Also in these legacy networks there is very little possibility for 3rd party participation in providing services for existing users. In fact, the network creators are service providers too.

The idea that service and control layers have similar logical and interoperability behavior in all networks (even though in reality they use different protocols and interfaces), has brought about a new idea in telecom world. The idea based on service, is a concept which limits technology at user access point and converges different networks in service and control layers. Future telecom world is moving towards network convergence and creation of an integrated infrastructure, independent of user access, with the intention of providing many services. Consequently, the most important effect of this convergence will be reduction in economical pressure created from different service infrastructure. It will also bring about employment opportunities for telecom operators, especially for operators with limited fund, who can operate as a 3rd party operator without incurring costly of infrastructure creation.

In this research, we propose a detailed SDP architecture for implementing within next generation networks to provide IT and telecom services in a convergence form. The most popularity of the proposed model is the modern technologies used in its architecture.

SDP CONCEPTS AND STANDARD MODELS

An SDP is a next generation services platform for telecommunications service providers supporting emerging standards such as IP Multimedia Subsystem (IMS) and Session Initiation Protocol (SIP) while at the

Corresponding Author: Amir Hasan Darvishan, Iran Telecommunication Research Center, P.O. Box 14155-3961, Tehran, Iran
Tel: +98 21 8497751 Fax: +98 21 8497777

same time inter-working with the legacy world as well. The SDP utilizes common off-the-shelf technologies provided by traditional hardware suppliers such as IBM, Intel, HP, or Sun while, meeting telecom-grade requirement such as 99.999% availability of service and a well-defined disaster recovery solution. Ideally, SDP is defined as stated below:

- Rapid deployment, provisioning, execution, management and billing of value added services
- Rapid deployment, provisioning, execution, management and billing of value added services
- Uniform and standardized access to different network capabilities and services as well as different content sources for use by application developers.

SDP can provide secure and open access for third party operators for the sake of providing new services utilizing existing network infrastructure and capabilities.

By analyzing different SDPs which have been presented by telecom organization or practical server vendors, a general concept for SDP based on Fig. 1 is understood.

Service execution block creates an environment for execution and development of a wide range of voice and data applications with the help of distributed service logic in different servers. This element also has the ability of services management, fault management, SLA management, load balancing, charging and billing registration. Network abstraction block provides standard interface for supporting service presentation in networks such as call management, session control, protocol transfer and user location and status. These interfaces are created with the intention of SDP element independence from network and the technology within it. Service Exposure block provides voice and data service presentation capabilities for third party operators. Utilizing this block, an operator can develop its network using standard and secure interfaces. On the other hand, other third party operators with no knowledge of telecom or understanding of the network can create their required services and present it in the network with the help of open standards. Content delivery block is used for management and service information or data transfer – like video on demand.

For proposing a novel service model, it is obvious that one should use standard concepts that are globally accepted by all telecom organizations. The reason for this decision is because all telecom manufacturers only support standard frameworks.

TISPAN model: According to ITU-T definition, next generation networks will have the ability to provide various telecom services to users and also provide unlimited user access to various service providers. In y.2011 recommendation, the network structure is separated to two layers called Transport stratum and service stratum (Fig. 2). These two layers can be developed independent of each other. Transport layer consists of 3 layers called Access layer, Aggregation layer and core layer and these provide data Transportation between network nodes. Transport in next generation networked is based on IP network which specify QoS and security features. IP-connectivity is provided to NGN user equipment by the transport layer, under the control of the network attachment subsystem (NASS) and the resource and admission control

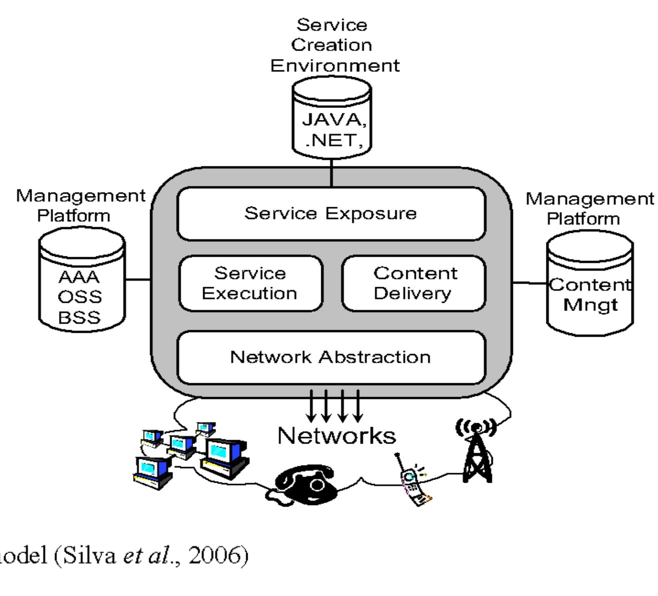

Fig. 1: SDP architecture model (Silva *et al*., 2006)

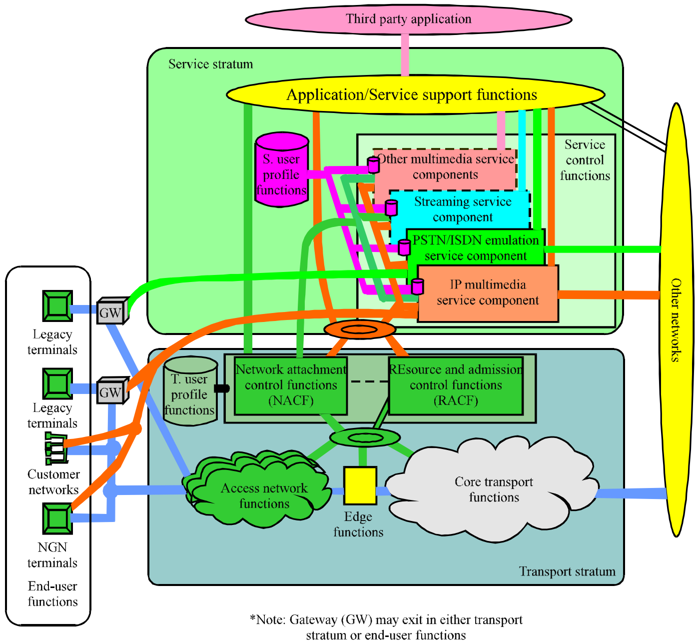

Fig. 2: TISPAN NGN layered architecture (ITU, 2004)

subsystem (RACS). These subsystems hide the transport technology used in fixed access and core networks below the IP layer.

The NGN supports business agreements between the access network operator and the network operator providing NGN IMS services (NGN IMS operator). The NGN IMS shall be able to offer services to users that are attached to access networks owned by another operator. The service offering may be restricted by the capabilities of the access network and the business agreement between the access network operator and the NGN IMS operator. Service layer is independent of Transport layer and provides the possibility to present integrated services to various users. The functionality of this layer consists of session and non-session related services, registration, media resources control, validation and authentication at service level and. The service stratum provides a variety of functions as communication services such as user authentication and registration, discovery of communication partners and negotiation of session conditions as well as supplementary services to support and enhance those services.

The NGN IMS supports the provision of SIP-based multimedia services to NGN terminals. Functional entities of an IMS may be used by an operator in support of transit network scenarios. The routing may be performed, depending on the entity performing the routing and depending on the traffic case, on signalling information, configuration data and/or data base lookup.

The NGN IMS, also known as Core IMS is a subset of the 3GPP IMS defined in 3GPP TS 23.002 which is restricted to the session control functionalities. Application Servers (AS) and transport/media related functions such as the Multimedia Resource Function Processor function (MRFP) and the IP Multimedia Gateway Functions (IM-MGW) are considered to be outside the core IMS.

NGN IMS: NGN IMS (Poikselka *et al*., 2006) model with presenting all services found in fixed and mobile network on a unified infrastructure, defines a standard architecture for next generation networks. The change of the existing networks to all IP network on which all services and media (Audio, image, music and …) are offered to users utilizing an integrated structure, is the foundation of IMS based networks (Grønbæk, 2006).

The main components of the IMS, as shown in Fig. 3, involved in SIP signalling, are the CSCFs. The CSCF SIP servers perform a number of functions such as multimedia session control and address translation function. In addition, the CSCF must manage service control, voice coder negotiation for audio communication and Authentication, Authorization and Accounting (AAA). The CSCF plays three roles:

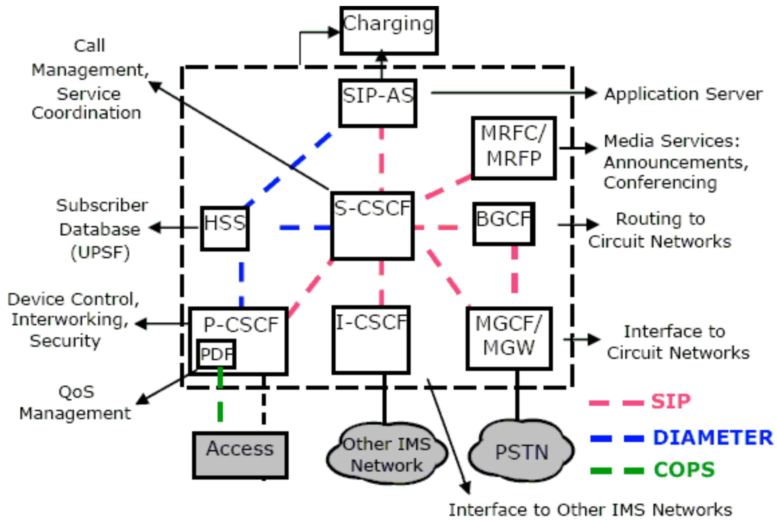

Fig. 3: IMS Functional Components (Grønbæk, 2006)

- The Proxy CSCF (P-CSCF) role
- The Interrogating CSCF (I-CSCF) role
- The Serving CSCF (S-CSCF) role

The S-CSCF is the key server with the main responsibility for the session management and service provision. The specification of the functional architecture of the IMS core component can be found by DES/TISPAN 02029 NGN-R1 (2006).

SERVICE CREATION TOOLS

New service Creation capability in service layer, independent of network infrastructure and lower layer technologies, is one important advantage and objective of next generation network. In order to able 3rd party operators to make use of these services, tools for creation and implementation of services have been devised. These tools can be presented in two main methods:

- Standard Application Programming Interfaces (APIs) that span diverse NGNs, allowing 3rd party application developers to produce new services. API tools define methods and classes with the use of common programming languages and network operators call these methods that are located in 3rd party operator server with the help of distributed software. In the past few years several industry efforts have emerged to develop such open APIs, including Parlay (Parlay Group, 2007), JAIN (Sun Microsystem, 2004,) and the Open Services Architecture (Parlay Group , 2007)
- Scripting languages are lightweight, highly customizable and typically interpreted languages, appropriate for rapid application development. These qualities and features make scripting languages applicable to the field of application programmability next to APIs. Scripting languages represent, in an XML-based file, a description of the service logic. Typically, scripts are created, edited and validated using regular editors or as a result of applying transformation techniques. For example the Service Creation Mark-up Language (SCML) (Bakker and Jain, 2002), VoiceXML (McGlashan *et al*., 2004) and Call-Control extensible Mark-up Language (CCXML) (Auburn, 2007) are scripting languages that connect existing components with a particular API, depending on the script file content

API tools

OSA/Parlay: The Open Service Access (OSA) /Parlay (Parlay Group, 2007) defines an architecture that enables the interworking between the IT applications and the telecommunications features through an open standardized interface. It is defined by the Parlay Group, an international consortium of Information Technology and Telecommunications companies.

Parlay specification provides 3rd party application developers with language-independent APIs that allow access to functions and capabilities. Hence, one can create service by defining the process of a service operation with API and execute that service by placing APIs executing codes on the 3rd party operator sever and calling these APIs from network operator server.

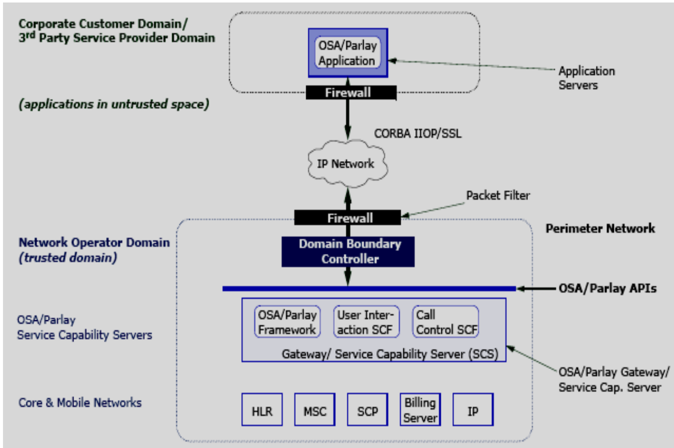

Fig. 4: Parlay Architecture (Parlay Group , 2007)

Figure 4 presents parlay architecture. The network functionality is described as Service Capability Features (SCFs) and applications could be deployed in a 3rd party domain. SCFs implement groups of Parlay/OSA APIs and provide access to the network capabilities. They are implemented by Service Capability Servers (SCSs) that are logical entities and interact with the network elements. These building blocks are placed on network operator server and a 3rd party operator with the help of distributed infrastructure like CORBA or WEB Service and with calling these APIs can create its own service and provide that service in the network.

SCSs as network resources, frees 3rd party operators from interaction with network infrastructure. In using this standard, any common programming language can be used for service creation and provision.

JAIN (Sun Microsystem, 2004a): The Java APIs for Integrated Networks is a set of API based on Java and is used for rapid product development and next generation network services. JAIN also provides a Java based infrastructure on which 3rd party operators and network operators can provide their services in an open environment. The JAIN is similar in many respects to Parlay nonetheless some of the differences are that:

- JAIN provides APIs not only at the functions layer but also at the protocols layer
- In JAIN all the APIs are specified in Java
- JAIN clearly defines a Service Logic Execution Environment (SLEE)
- JAIN defines a Service Creation Environment (SCE)

JAIN provides service portability, independence from network, open and secure deployment of telephone, data and wireless network resources. JAIN supports a set of API at different network levels including JAIN SIP, JAIN IN... call control and JCC/CAT (Sun Microsystem, 2001b). These API are utilized for creation, handling and terminating of sessions with 3rd party JAIN (Fig. 5).

In JAIN there is also an environment for execution and management of services called JSLEE. This part acts as a software platform for network operator server. In this environment, service building blocks with the name of Service Building Block (SBB) are implemented and 3rd party operators with the help of JAIN-SPA can have access to these blocks. Note that SBB is the same as SCS is parlay.

Script tools

CCXML/Voice XML: In general voiceXML is used as a technology allows a user to interact with a web server through voice-recognition technology, which exploits Media Server capabilities. One dynamic factor in providing voice XML is its responsive capability to IVR users. VoiceXML framework is based on HTTP protocol and is related with Internet and telephone network. In this system, the user is confronted to some resources like pre-recorded audio, text-to-speech synthesis (TTS), Automatic Speech Recognition (ASR) and DTMF. VoiceXML is a high-level abstraction language and this means that developers with little training can use it. VoiceXML makes it easy to rapidly create new applications and protect developers from low level programming issues.

VoiceXML provides very little in terms of call control, besides CCXML by providing advanced telephony

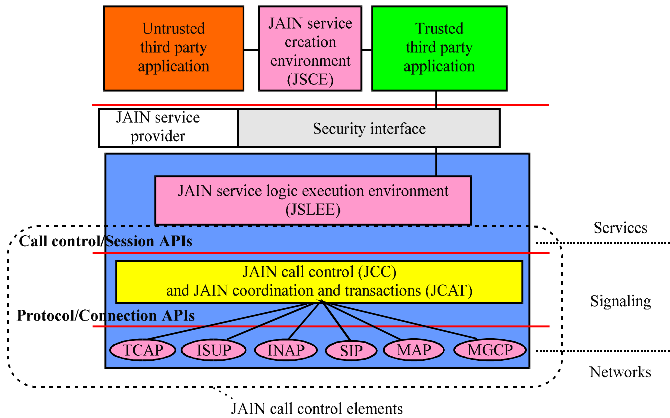

Fig. 5: JAIN Architecture (Sun Microsystem, 2004a)

operation like conference control, click-to-dial and is a supplement for voice XML. CCXML has been designed to complement and integrate with a VoiceXML system. CCXML is used to overcome the weak point of voice XML in handling call control.

SCML: The service creation mark-up language was designed by JAIN engineering group with the intention of service creation in next generation networks. Even though current engineering group endeavour is concentrated on call control capabilities, the objective of SCML designers is based on providing services like mobility, multiparty, multimedia and etc.

SCML server is based on JCC programmable interface and is therefore independent from network protocol and signalling layer. This language hides network technology from service creator and provides basic events managements so that call control can be implemented. In short, once SCML server receives a service request via JCC, it performs service logic and sends the appropriate message to network control layer (with the help of network protocol) via JCC.

DISTRIBUTED SOFTWARE CREATION TOOLS

The objective of presenting distributed processing technologies is to provide means and standards required for production and utilization of components for distributed software creation. With these in hand, utilization of services in local or remote distributed software will be provided and in consequence 3rd party operators can interact with network users based on service and independent of network infrastructure. Below three relevant technologies are considered:

RMI Technology: RMI (Sun Microsystem, 2001a) provides Java programmer with mechanisms so that they can execute different object methods on a remote virtual machine (JVM). Software objects can completely be passed or returned like a parameter. This feature means that a Java programmer with the help of RMI, can transfer new codes in the network and execute them in remote virtual machine in a dynamic way. This unique feature has allowed this technology to be used in distributed system creation. In a distributed environment, RMI clients can have access to new version of Java services and there is no need to distribute programs to clients.

CORBA technology: This technology provides the *means* to call remote software object methods and transfer various simple data type or data structures consisting of simple data type. CORBA (OMG, 2004) services are defined via an interface written with IDL. Different IDL for various programming languages have been devised and unlike RMI, this technology is not limited to a specific language and with it many distributed software systems in distributed environments can be related. CORBA technology because of its easy usage, flexibility, portability (in different languages) and speed in real time applications has become popular among programmers for linking different software platforms for a system development.

COM technology: After introducing different technologies by Microsoft like OLE (Object Linking+Embedding) for linking various MS-office documents and OLE2 for linking all windows supported documents, COM technology was introduced by Microsoft. At first, this technology with respect to distributed components did not offer a substantial capability. But with presenting of Win-95 and Win-NT and necessity of distributed components and the relation between these components, Distributed COM (DCOM) and its improved version with the name of COM+ was introduced. Both DCOM and CORBA work based on client/server model for their communication between

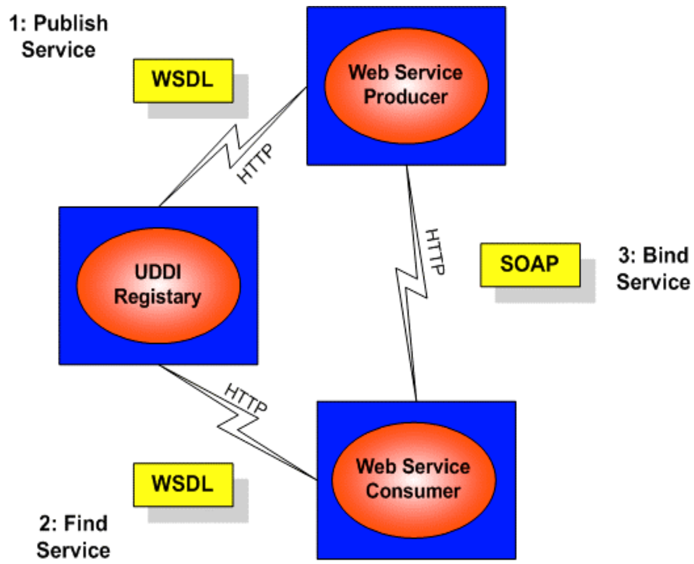

Fig. 6: Web service mechanism (W3, 2002)

components and therefore have similar behaviour. DCOM compared with other technologies have less popularity and is mostly used in .NET platforms.

Web service technology: This mechanism is a method for using programs and distributed services in internet infrastructure. This technology with the help of common web based standards and protocols, like HTTP and XML, provide the capability to call services located on various servers. This similarity is the cause of this technology increasing in current software architecture. Currently this technology in many SDP architectures is used as a key interface for connection with 3rd party operator servers. The way this mechanism behaves is shown in Fig. 6. In step 1, service provider with WSDL (Christensen *et al*., 2001) descriptive language based on XML- defined by W3 organization-registers its own services in UDDI database which is installed in many important web sites such as IBM and Microsoft. Service acceptor refers to these data bases and selects the required service, then obtains WSDL description service and with the help of its own software platform- like .NET- makes connection with the service provider and obtains the required service using SOAP protocol (Gudgin, 2007). SOAP protocol is based on XML and is transferred on HTTP protocol. It should be indicated that J2EE architecture supports this technology.

SOFTWARE PLATFORM

J2EE: J2EE (Sun Microsystem, 2005) is a platform-independent, Java-centric environment, building and deploying Web-based enterprise applications online. The J2EE platform consists of a set of services, APIs and protocols that provide the functionality for developing Web-based applications.

J2EE architecture introduces a standard for design, development deployment of EJB software in a distributed environment-JAVA RMI mechanism. This architecture provides to the programmer rapid application development, transaction management, security and database requirements. Therefore, software developers only concentrate on the logic of their programs. Currently many companies like IBM and Oracle offer commercial J2EE service providers:

- J2EE supports pure HTML, as well as Java applets or applications
- Enterprise JavaBeans (EJBs) provide another layer where the platform's logic is stored. An EJB server provides functions such as threading, concurrency, security and memory management. These services are transparent to the author
- The Java servlet API enhances consistency for developers without requiring a graphical user interface

Nowadays software developers create transactional distributed programs using Enterprise Java Beans (EJB) and using this architecture can increase speed, security and reliability to a greater extent. EJB is a Java API developed by Sun Microsystems that defines component architecture for multi-tier client/server systems. EJB systems allow developers to focus on the actual business architecture of the model, rather than worry about endless amounts of programming and coding needed to connect all the working parts. Developers just design (or purchase) the needed EJB components and arrange them on the server. Because EJB systems are written in Java, they are platform independent. Being object oriented, they can be implemented into existed systems with little or no recompiling and configuring.

As shown in Fig. 7, J2EE not only support EJBs, it also supports web applications too. Servers based on J2EE internally consist of web servers and therefore make connection with network using HTTP protocol. This point means that J2EE servers not only are used in Internet applications, but also support WEB services, which is an important point in providing services in next generation networks.

JSLEE: JSLEE (Sun Microsystem, 2004b) with high performance, flexibility and open standard provides an environment for service creation and presentation. JSLEE inherits some concepts from J2EE, yet does not replace or contradict the J2EE environment. JSLEE is a complementary platform addressing very specific and special requirements not covered by J2EE as of today. However, the integration between both platforms is explicitly foreseen by the standards. The invention of JSLEE was motivated by the telecommunication industry's trend towards component-based architectures realized in open, standardized platforms. JSLEE is standardized and hence helps reduce time to market and development costs. Being Java-based, the paradigm of "write once, run anywhere" is supported and allows portable standards-compliant applications. As shown in Fig. 8, the followings are some of the architectural features:

- Use of Java technology like JVM, JDBC and JMX
- Comprised of JAIN interfaces like JCC for service execution independent of network technology and JSPA for service creation by third party operators
- Ability to interact with J2EE and make use of WEB service mechanism
- Independent of network technology and its protocols

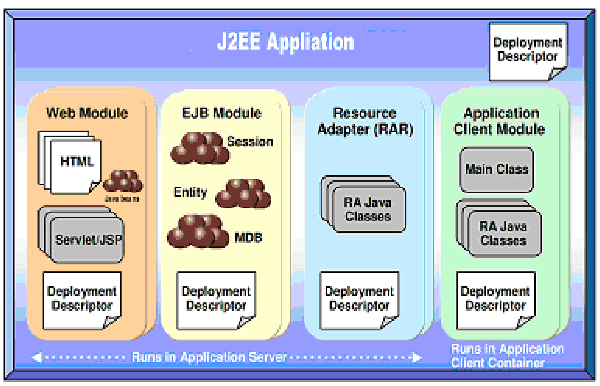

Fig. 7: J2EE Platform (Sun Microsystem, 2005)

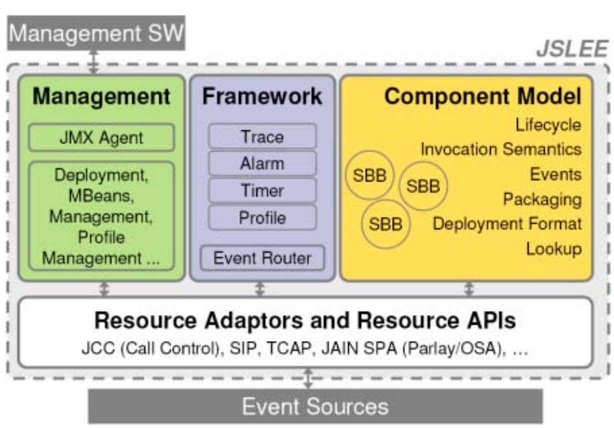

Fig. 8: Overview of JAIN SLEE's standardized architecture (Sun Microsystem, 2004b)

In this environment, service developers create service building blocks (SBB) in the form of classes of Java language and install them in JSLEE. JSLEE with the help of Event Router blocks and Resource Adaptor interfaces, route protocol events to these classes and in reverse pre-defined service logic in these classes are converted to network sessions. Besides, JSLEE introduces network abstraction by the means of resource adaptors.

OPERATOR SERVICE DELIVERY MODEL

In Fig. 9, the overall aspect of service model consisting of model elements and interface are presented. In this Fig. 9 mapping of elements and applications to building blocks of a SDP, is also shown.

In order to create new services in the network or/and services presentation by third party operators, application server supports standard parlay (Parlay Group, 2007) interfaces. Also service creation environment (SCE) which is usually in graphic form provides service creation capabilities using network resources. Application server generally cares for service session control (among user or users with media server). Media server provides media processing requirements in executing multimedia services. Therefore, both these elements fake part in service execution. Application server should also be capable of providing services for legacy networks like PSTN and be able to provide services for next generation networks too. Therefore, this element is required to support various network protocols. SIP is connecting protocols between elements and this protocol is used for service session creation. Application server is utilized media server functionalities and resources by using other protocols like VXML (McGlashan *et al*., 2004) and MSCL (Grønbæk, 2006). In previous, practical implementation of proposed model with the help of current telecom technologies will be explained.

Practical service model for integrated networks is shown in Fig. 10 operational nodes called application server, Media server and 3rd party are considered in the model.

Application server: Software platform for this node is JSLEE which provides an environment for service execution. This environment was introduced in next section. Protocol events, are routed to service classes by Event Routers based on defined profiles in this platform. Service classes are execution logic for a specific service which has been implemented by JAVA language. In JSLEE, service building blocks are installed as network resources- according to SCS parlay standards and utilized by service classes and third parties. These basic blocks, provide service interactions with networks via standard APIs.

Third parties have access to these resources using mechanisms like RMI or WEB service. A graphical environment for service creation is also provided in application server and its outputs are service classes in JAVA language. These classes are used by JSLEE classes for new service execution. In SCEs network resources are

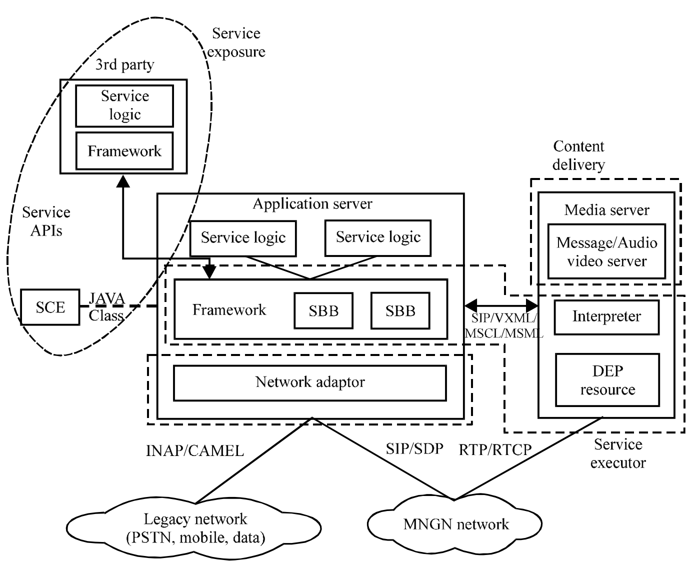

Fig. 9: Proposed Service model architecture

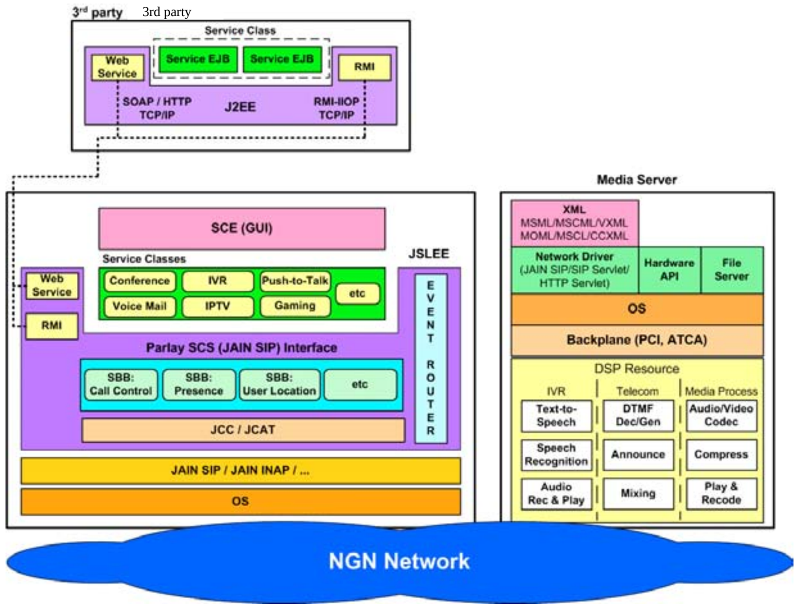

Fig. 10: The detail of proposed service model

used as basic blocks for describing a service. As it was indicated earlier, JAIN SIP, JAIN INAP and JCC provide independence from network protocol for service developers.

Media server: This element consists of two parts: Hardware and software. Hardware consists of processing cards based on DSP processor, along with device for IVR services, telecom features like DTMF detection/creation, mixing for video for video/voice conferences, announce, media processing like various video/voice coding and compression and recording of media.

These hardware resources are used by XML interpreter APIs for MSML and MSCML support. In this element, accessing network is performed with the help of SIP protocol and use of JAIN SIP or SIP servlet APIs. File server provides the requirements for media saving in the database.

3rd party: 3rd party operators with the help of J2EE platform can implement their services in the form of EJB and via RMI or WEB service mechanisms make connections with application server and call network resources APIs.

THIRD PARTY SERVICE MODEL

Nowadays, moving towards next generation networks is in its initial phase. Telecom operators are devising methods and state of developments of various layers for easy migration. Therefore a unified and an accurate reference can not be found which explains all layers of network. Ambiguity in service layer and new concepts is becoming more colourful because of lack of experience with these entities. In this research, as shown in Fig. 11, we propose comprehensive model for service creation and delivery in next generation networks.

In order to implement the mentioned model, the first step is to provide IP connection between servers. This connection can be created by the network operator with the creation of a private IP network which guarantees network security and quality of service. In case existing IP network is used, this connection should be set up utilizing common methods like VPN, L2/L3 MPLS VPN, VPLS and IP SEC. Note that the way IP connections between 3rd party severs and network are made, has no effect on overall model which is concentrated on service presentation. What is important is the existence of a secure IP connection between servers with a good quality of service.

As it is indicated, there are two methods for creating services based on API and script. Therefore this model should have compatibility with any 3rd party operator or any method. Those operators who use API tools need a CORBA, RMI or WEB service server so that they can call remote methods. As it is seen in Fig. 6, service logic, service execution mechanism consist of sessions in the network for service presentation, is located on 3rd party operator server and network operator by calling standard APIs on its own server and mapping service to API.

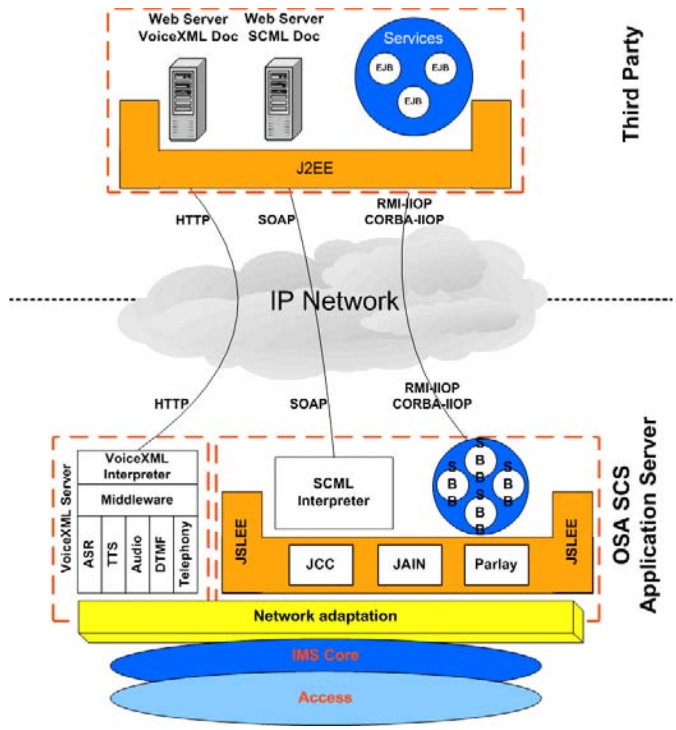

Fig. 11: Service model for 3rd party partnership

Network operator by calling standard APIs on its own server, maps services to APIs via using JSLEE profiles which causes the service logic to be performed on the 3rd party operator server. Results are then returned to network operator server with the help of network adoption block, which is the same as JAIN protocol API and results are then converted to network sessions. CORBA service provider provides above mentioned mechanism for parlay API and RMI service provider provides it for JAIN API. Web service is also used in connection with parlay X. Software platforms based on J2EE technology and with support for all three service provider, are the best option for the implementation of this mechanism.

Third party operators, who utilize scripts for providing services, have a simpler mechanism. These operators provide their services based on XML format and whenever the network user request for such a service, its XML document is provided to network operator. This mechanism is very similar to processes that are performed in Internet. SCML interpreter and voice XML/CCXML play the same role in network operator as Internet Explorer does for Internet. When specific service (with a specific URL) is requested by a user, SCML or voiceXML/CCXML server with the help of HTTP or SOAP protocol, receive the relevant XML document or message from web service provider and by interpreting that request, server the user.

In voice XML/CCXML technology, as shown in Fig. 12, 3rd party operators write the required service text with common editors and deploy it on their own web server and when the network operator makes a request for that, the relevant document is sent for the network operator and the existing voiceXML/CCXML interpreter on network server will interpret and with the help of basic service blocks like speech recognition, sound file recording, DTMF and etc. will provide the required service to the user. Network operator server will play voiceXML gateway for service creation and execution utilizing this mechanism. VoiceXML gateway is usually located on a separate OSA SCS application server.

In SCML technology, as shown in Fig. 13, there are 3 different ways for providing services. There are three possibilities here: 1) the SCML interpreter could reside on the 3rd party server itself, 2) the SCML interpreter could reside on an Application Server, or 3) the SCML interpreter could reside on both sides.

In case 1, the required service will be interpreted and executed on the 3rd party operator server and SCML interpreter makes remote calls to the JCC interface (e.g., using Java RMI). In the case 2 in which only SCML interpreter reside on network operator server, SCML document is received and executed by network operator with the help of HTTP protocol. In the case 3, web service

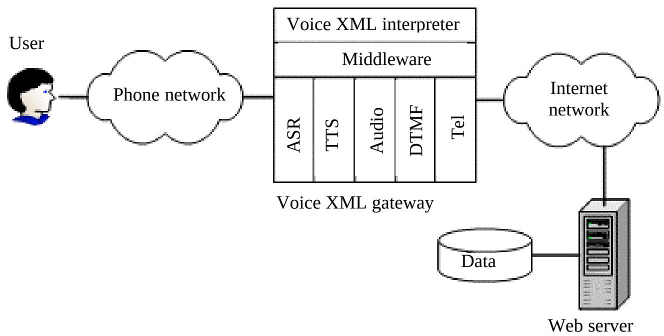

Fig. 12: VoiceXML system architecture

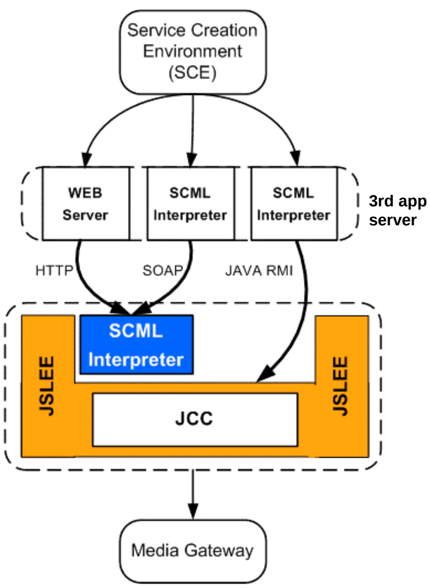

Fig. 13: SCML system architecture

will provide the means for service execution. In this case, 3rd party SCML interpreter send and XML message via SOAP protocol for network SCML interpreter and then XML document is received and executed by network SCML interpreter.

CONCLUSION

In this research, a model for the use of network developers was introduced, in which all objectives and requirements of next generation networks utilizing current commercial tools and technologies was considered. This model reduces network implementation time and makes design simple.

This model, by creating a service convergence layer in the network and presentation of various telecoms and IT services, feeds various users of access networks. The recommended model uses current software platforms like J2EE and open standards for network resource sharing. This policy eases the way for third party operators to participate and use network resources in order to provide services.

Also, a model for relating different 3rd party operators to network operators was put forward and needed requirements for making connections and service creation between these operators was revealed. From a scientific point of view this model is regarded as a flexible and easy model for implementation because it uses common mechanisms and platforms like HTTP protocols, SOAP and J2EE which is highly used by service provider these days. Other advantages of this model are its scalability and redundancy. In this model, 3rd party operators can use text or API technology to create and provide their own services without any limitations. In this model, there is no limitation to the number of 3rd party operators involved and the only prerequisite for the presence of 3rd party operators in this model, is the existence of a secure IP connection between the servers.

REFERENCES

Auburn, R.J., 2007. Voice browser call control: Ccxml version 1.0, w3c working draft. http://www.w3.org/TR/.

Bakker, J.L. and R. Jain, 2002. Next generation service creation using xml scripting languages. ICC, New York, USA. www.argreenhouse.com/papers/jlbakker/bakker-icc2002.pdf.

Christensen, E., F. Curbera, G. Meredith and S. Weerawarana, 2001. Web services description language (wsdl). http://www.w3.org/TR/wsdl.

DES/TISPAN 02029 NGN-R1, 2006. Telecommunications and internet converged services and protocols for advanced networks (tispan); ip multimedia subsystem (ims); functional architecture. http://webapp.etsi.org/WorkProgram/Report_WorkItem.asp?wki_id=21411.

Eurescom Project P1109, 2001. Next generation networks: The services offering standpoint. http://www.eurescom.de/secure/projects/P1100-series/P1109/P1109.htm.

Grønbæk, I., 2006. NGN, IMS and service control-collected information. Telenor, R and I Research Note No. 31/2006. www.telenor.com/ri/pub/not06/n_31_06.pdf.

Gudgin, M., M. Hadley, N. Mendelsohn, J. Moreau, H. Frystyk Nielsen, A. Karmarkar and Y. Lafon, 2007. Simple object access protocol (soap) 1.2 specification. http://www.w3.org/TR/soap12-part1/.

ITU, 2004. Y.GRM-NGN: General principles and general reference model for next generation network, Recommendation Y.2011. http://www.itu.int/ITU-T/studygroups/com13/ngn2004/index.html.

McGlashan, S. *et al*., 2004. Voice extensible markup language (Voicexml) Version 2.0, w3c recommendation. http://www.w3.org/TR/2002/WD-voicexml20-20020424.

OMG, 2004. Common object request broker architecture (corba) specification. http://www.omg.org/technology/documents/formal/corba_iiop.htm.

Parlay Group, 2007. Parlay/OSA API Specifications. http://www.parlay.org/en/specifications/apis.asp

Poikselka, M., A. Niemi, H. Khartabil and G. Mayer, 2006. The IMS: Ip Multimedia Concepts and Services in the Mobile Domain. 2nd Edn., John Wiley and Sons, New York, ISBN: 978-0-470-01906-1.

Silva, N.M. *et al*., 2006. Service oriented architectures for convergent service delivery platforms. P1652, EDIN 0532-1652.

Sun Microsystem, 2001a. Java remote method invocation (Java RMI). http://java.sun.com/javase/technologies/core/basic/rmi/index.jsp.

Sun Microsystem, 2001b. JSR 21: JAIN JCC Specification. http://jcp.org/en/jsr/detail?id=21.

Sun Microsystem, 2004a. JSLEE and the JAIN Initiative. http://java.sun.com/products/jain.

Sun Microsystem, 2004b. JSR 22: JAIN SLEE API Specification. http://jcp.org/en/jsr/detail?id=22.

Sun Microsystem, 2005. J2EE Java Platform, Enterprise Edition (Java EE). http://java.sun.com/javaee/.

W3, 2002. Web Services. http://www.w3.org/2002/ws/.