

Specificații Java pentru dezvoltarea aplicațiilor mobile folosind SIP și IMS

Iolanda ALECSANDRESCU, Dănuț BURDIA*

Abstract. Convergence is a key trend in the evolution of communications technologies. The telecommunication and data networks are converging on the basis of packet switching. These IP-based telecommunication networks are referred to as IP Multimedia Subsystem (IMS). The IMS is based on Session Initiation Protocol to provide voice services as well as data services to mobile devices in real time over third generation (3G) wide area cellular networks. Java technology has become widely available in mobile terminals and provides a standard platform for application development. Due to the availability of Java in a wide range of mobile terminals it is the most suitable technology platform to implement SIP-based applications for this kind of terminals. This paper focuses on how Java applications can benefit from IMS and SIP.

Keywords: Mobile Applications, IMS, SIP, 3G, Java,

Rezumat. Convergența este o tendință cheie în evoluția tehnologiilor de comunicații. Rețelele de telecomunicații și cele de date sunt convergente pe baza comutării de pachete. Aceste rețele de telecomunicații bazate pe IP sunt cunoscute ca subsisteme multimedia IP (IMS). IMS se bazează pe protocolul de inițializare a sesiunii (SIP) pentru a furniza servicii de voce, precum și servicii de date în timp real pentru dispozitive mobile în rețelele celulare de arie largă de generația a treia (3G). Tehnologia Java a devenit disponibilă pe scară largă în terminalele mobile și oferă o platformă standard pentru dezvoltarea de aplicații. Datorită disponibilității de utilizare a tehnologiei Java într-o gamă largă de dispozitive mobile, aceasta reprezintă platforma cea mai potrivită pentru implementarea aplicațiilor bazate pe SIP pentru aceste tipuri de terminale. Această lucrare prezintă modul în care aplicațiile Java beneficiază de avantajele subsistemului multimedia IP și a protocolului SIP.

Cuvinte cheie: Aplicații mobile, IMS, SIP, 3G, Java

Introducere

Subsistemul multimedia IP îmbină mobilitatea rețelelor celulare de arie largă cu funcțiile rețelelor IP. Cu IMS multe dintre aplicațiile de rețea, cum sunt email-ul, mesageria instant, Voce peste IP (VoIP) și video streaming pot fi disponibile și în rețelele wireless. IMS și-a propus să ofere toate serviciile pe care le oferă internetul și de asemenea să furnizeze noi servicii [1]. De asemenea, utilizatorii rețelelor vor putea folosi serviciile atât în rețelele lor de bază cât și în roaming. IMS permite

* Department of Telecommunications, Technical University "Gheorghe Asachi" of Iasi, Romania

utilizatorului să negocieze cu operatorul calitatea serviciului în timpul sesiunii cât și în timpul inițierii acesteia. Subsistemul multimedia IP va oferi operatorilor de rețele posibilitatea să controleze sesiunile de comunicație prin utilizarea protocolului SIP. Pe baza acestui protocol furnizorii IT cum sunt Skype, MSN, Google etc. oferă comunicații diversificate prin combinarea serviciilor de email, chat, apeluri voce și video.

Datorită independenței de platformă, aplicațiile mobile folosind Java domină piața, iar introducerea subsistemului multimedia IP face din Java cea mai potrivită platformă pentru implementarea aplicațiilor bazate pe SIP pentru terminalele mobile [2,3]. Pentru a vedea cum influențează IMS aplicațiile Java mobile, în această lucrare se va face o descriere a subsistemului multimedia IP și a protocolului SIP folosit de IMS. În ultima parte a lucrării se vor prezenta specificațiile oferite de Java pentru SIP și relațiile dintre aceste specificații.

Protocolul SIP

Protocolul de inițializare a sesiunii (SIP) este un protocol de semnalizare standard de la nivelul de aplicații ce este utilizat pentru a iniția, modifica, controla și încheia o sesiune între doi sau mai mulți agenți utilizatori. SIP permite utilizatorilor să se conecteze la rețea în diferite puncte de acces și să fie totuși găsiți de către alți utilizatori și aplicații, indiferent de localizare. Acest protocol a fost conceput pentru a fi independent de nivelul de transport și poate rula peste diverse protocoale de transport cum ar fi protocolul de control al transmisiunii (TCP), protocolul de datagrame (UDP) sau protocolul de control a transmiterii tip flux (SCTP – *Stream Control Transmission Protocol*). SIP este un protocol simplu ce are la bază modelul de tip cerere-răspuns folosind mesaje de tip text, având elemente de design similare cu HTTP-ul și alte standarde internet de la IETF (*Internet Engineering Task Force*). Aplicațiile protocolului SIP variază de la telefonie prin Internet la aplicații de control, incluzând e-commerce, conferințe multimedia, servicii de mesagerie instant. SIP a fost selectat pentru subsistemul IMS ca protocol de semnalizare [4].

SIP are o arhitectură client server în care clienții SIP (sau agenții utilizatori) sunt noduri finale [6]. Serverul proxy SIP păstrează locația finală (adresa IP) a clienților. Utilizatorii sunt identificați prin identificatorul uniform de resurse (URI). Prin înregistrarea cu serverele de înregistrare SIP, URI-ul utilizatorului primește limitele adreselor IP ale dispozitivelor pe care utilizatorul le are conectate la rețea. Astfel,

folosind un server proxy, utilizatorii înregistrați pot fi contactați prin propriile identificatoare URI, chiar dacă adresele lor de contact nu sunt cunoscute. Orice utilizator poate găsi astfel adresa de contact a altui utilizator folosind serverul proxy SIP și apoi poate stabili o sesiune media cu acesta.

Există două tipuri de mesaje SIP: cereri (trimise de client spre server) și răspunsuri (trimise de server spre client). Mesajele de răspuns conțin coduri de răspuns numerice, ce au la bază codurile de răspuns HTTP. Există două tipuri de răspuns și șase clase pentru coduri. Tipurile de răspuns sunt: *provizorii* (clasa 1xx) – răspunsuri provizorii folosite de server pentru a indica faptul că cererea este în curs de procesare, dar nu terminată și, respectiv, *finale* (clasele 2xx, 3xx, 4xx, 5xx, 6xx) – răspunsuri finale de terminare a tranzacției SIP. Pe scurt, semnificația codurilor este următoarea: 1xx- provizorii: apelare, căutare, așteptare etc, 2xx- succes: cererea a fost acceptată cu succes, 3xx- redirectare: acțiuni suplimentare ce trebuie luate pentru a completa cererea, 4xx- eroare client: cererea nu a fost completă din cauza unei erori a clientului, 5xx- eroare server: serverul nu a reușit să satisfacă cererea, 6xx- erori globale: cererea nu poate fi îndeplinită din diverse motive, cum ar fi ocupat, refuzat, nu poate fi acceptată de orice server. Structura unui apel SIP este prezentată în figura 1.

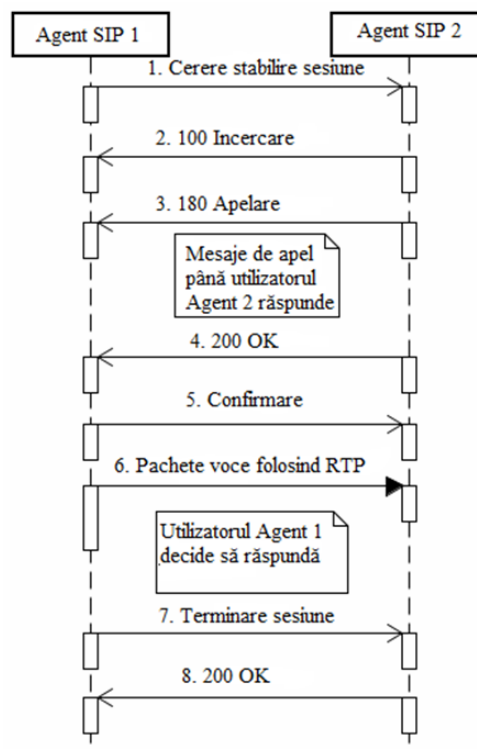


Fig. 1. Structura unui apel SIP

În figura 1 clientul Agent 1 trimite un mesaj INVITE (REQUEST) către server-ul proxy pentru a-și exprima intenția de începere a unei sesiuni VoIP (sau alt tip de sesiune, cum ar fi sesiune video) cu clientul Agent 2. Mesajul SIP include de asemenea un mesaj SDP (*Session Description Protocol*) de descriere a conținutului media al sesiunii, de exemplu parametri ca: tipul codecului audio ce trebuie folosit, portul RTP (*Real Time Transport Protocol*) folosit pentru transmiterea unui flux de date audio sau video. Server-ul proxy trimite un mesaj INVITE clientului Agent 2 incluzând parametrii SDP așa cum a fost trimis de clientul Agent 1. Clientul Agent 2 acceptă apelul și eventual trimite un mesaj SIP 200 OK către serverul proxy, care la rândul său trimite un mesaj SIP 200 OK clientului Agent 1 indicând succesul începerii sesiunii. Cele două noduri finale negociază parametrii relevanți, încep sesiunea VoIP și comunică între ele folosind pachete de voce pe porturile RTP. În cazul altor tipuri de sesiuni media, procedeul este același dar diferă doar parametrii SDP utilizați.

Există o gamă largă de aplicații ce folosesc SIP cum ar fi VoIP, videoconferințe, mesageria instant, prezența, înregistrare utilizator, push to talk over cellular (PoC), soluții de afaceri și jocuri în rețea. Prin SIP distribuitorii de servicii pot personaliza și furniza o serie de servicii care pot include conferință, prezență, mesagerie instant și media într-o singură sesiune. Astfel distribuitorii de servicii pot crea mai degrabă o aplicație flexibilă decât să susțină aplicații multiple rigide. Acest principiu reduce pentru furnizori costul proiectării și dezvoltării serviciilor și, în plus, utilizatorii se pot bucura de servicii îmbunătățite.

Arhitectura IMS

Subsistemul Multimedia IP (IMS - *IP Multimedia Subsystem*) este o platformă pentru furnizarea de servicii multimedia bazată pe protocoale de internet. Prin această tehnologie se permite utilizatorilor să acceseze, creeze și schimbe conținut digital folosind diverse metode de acces, de exemplu cu dispozitive wireless de generația a treia –3G (telefoane mobile, PDA, etc.) dar și prin acces obișnuit la internet prin rețele locale (LAN) sau conexiuni de bandă largă.

Principalul scop al platformei IMS este de a reuni domeniile comunicațiilor mobile și de internet pentru a simplifica managementul rețelelor și de a oferi o altă dimensiune, mai generală, pentru comunicațiile multimedia. Acest deziderat este realizat datorită separării funcțiilor de control de cele de transport. Aceasta înseamnă

că în cadrul tehnologiei IMS serviciile sunt separate de rețelele folosite pentru transportul acestor servicii.

Inițial, conceptul de subsistem multimedia IP a fost dezvoltat în cadrul proiectului în parteneriat 3GPP (*Third Generation Partnership Project*) [5] ca mijloc de a oferi operatorilor mobili posibilitatea de a furniza mai eficient servicii de date prin migrarea de la sistemul GSM la 3G. Ulterior, tehnologia IMS a fost adoptată de alte organizații de standardizare atât pentru rețele pe fir cât și pentru cele wireless.

Conceptul IMS poate fi privit ca o metodă prin care furnizorii de servicii și transport din telecomunicații își pot construi rapid și ușor infrastructura de bază, interoperabilă, pentru furnizarea serviciilor și aplicațiilor multimedia. Deoarece furnizarea acestor servicii și aplicații multimedia se bazează pe interfețe și protocoale standardizate, acestea permit furnizorilor să dezvolte și să integreze noi servicii ce vor fi livrate prin infrastructura IMS. Totodată, tehnologia IMS oferă un mecanism clar pentru gestionarea numărului mare de servicii pentru internet utilizând tehnologii ca HTTP și Java, permițând acestor servicii să fie utilizate și în comunicațiile mobile. În plus, IMS permite ca serviciile să poată fi transferate fără probleme între rețelele de acces atunci când utilizatorii comută între aceste rețele.

Arhitectura IMS este prezentată într-o formă simplificată în figura 2 și conține multiple entități funcționale dispuse în cadrul a trei nivele: nivelul de transport, nivelul de control și nivelul de aplicații. Fiecare entitate are interfețe deschise bine definite astfel că furnizorii pot crea ușor produse care să ofere diverse funcții sau să asigure interfața cu diferite părți ale sistemului. Totodată, furnizorii sunt liberi să creeze produse care combină multiple funcții din IMS sau pot alege să producă componente hardware sau software care adresează o anumită funcție din IMS.

Nivelul de transport cuprinde mai multe tipuri de rețele de acces bazate pe pachete (GPRS, UMTS, CDMA2000, rețele locale wireless – WLANs, ADSL). Rețeaua de telefonie publică tradițională (PSTN) este un bun exemplu de rețea cu comutare de circuite. Utilizatorii sunt conectați la infrastructura IMS fie prin nivelul de transport, fie direct printr-un terminal IMS (de exemplu, un dispozitiv wireless 3G) sau printr-un dispozitiv non-IMS care interfațează infrastructura IMS printr-un gateway.

Nivelul de control este partea din IMS care furnizează toate funcțiile de control al sesiunilor și apelurilor. Acest nivel cuprinde trei componente :

– Funcția de control a sesiunii de apel (CSCF – *Call Session Control Function*), este responsabilă cu înregistrarea și autentificarea utilizatorilor în rețea. Întrucât IMS utilizează pentru semnalizare protocolul SIP nu este necesară cunoașterea adresei IP destinație deoarece pentru a contacta destinatarul poate fi utilizat identificatorul de resurse SIP URI, definit în standardul RFC 3261 și care este asemănător unei adrese de email. Funcția de control a sesiunii reprezintă prima cale de apel pentru un utilizator când încearcă să se conecteze la rețea deoarece asocierile dintre SIP URI și IP sunt menținute de aceeași funcție.

– Serverul abonaților (HSS – *Home Subscriber Server*), este o bază de date centralizată care conține toate informațiile utilizatorilor, cum ar fi localizarea rețelei de bază, informații de securitate, informații despre profilul utilizatorilor, etc.

– Controlul media, acționează ca o interfață între serverul de aplicații din nivelul de aplicații și serverul media situat în nivelul de transport.

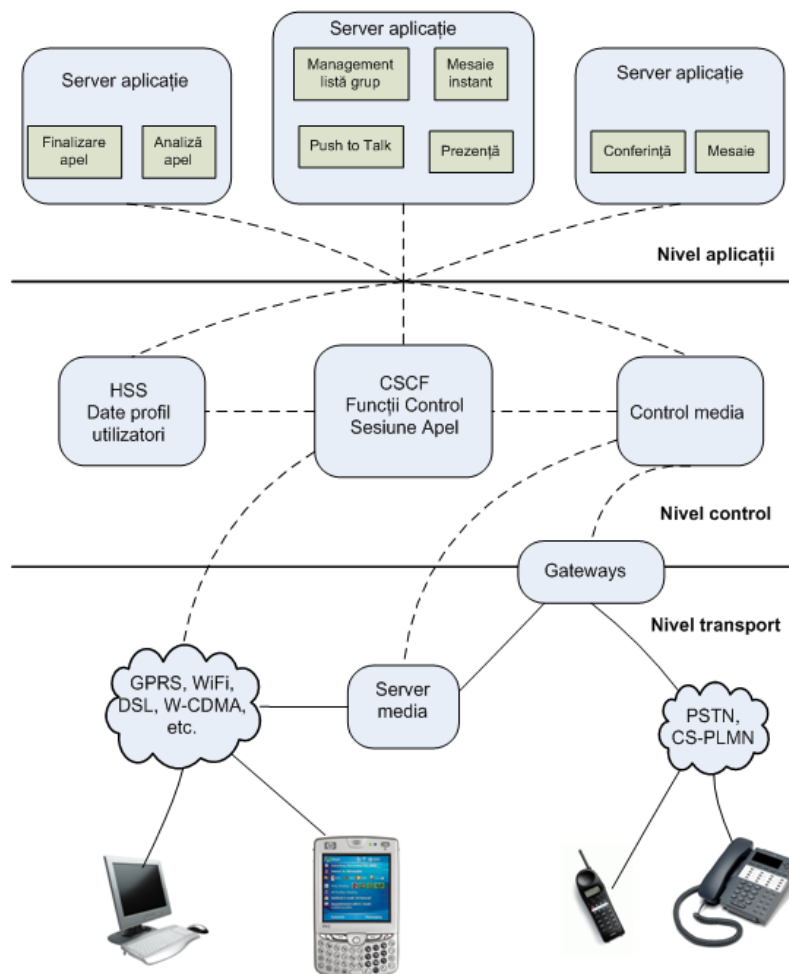


Fig. 2. Arhitectura simplificată a IMS

Nivelul de aplicații este responsabil cu furnizarea serviciilor de telefonie și date. În acest nivel se află serverele de aplicații iar furnizorii de rețea pot utiliza oricâte servere pentru a oferi aceste servicii. Toate serviciile sunt livrate către nivelul de control prin interfața IMS folosind protocoale standardizate, în special protocolul SIP.

Standarde Java implementate pentru SIP

În cadrul tehnologiei Java sunt definite trei platforme, numite Java 2 Standard Edition (J2SE), Java 2 Enterprise Edition (J2EE) and Java 2 Micro Edition (J2ME). Comunitatea Java a creat specificațiile pentru aceste trei platforme, sub egida Java Community Process, primul pas în dezvoltarea acestora fiind o cerere de tipul Java Specification Request (JSR). Specificațiile Java sunt referite adeseori sub forma numărului corespunzător JSR. Platformele Java împreună cu categoriile de dispozitive care le suportă sunt prezentate în figura 3.

Platforma J2SE (Standard Edition) definită pentru aplicații desktop, oferă suport pentru crearea de aplicații independente și appleturi. Această platformă include tehnologia JavaWeb ce permite lansarea și instalarea locală a programelor scrise în Java direct de pe web. Această platformă cuprinde două componente de bază: JRE (*Java Runtime Environment*) și JDK (*Java Development Kit*). JRE pune la dispoziție biblioteci, mașina virtuală Java și alte componente necesare pentru a executa appleturi și aplicații scrise în limbajul Java. În plus, ca parte a JRE există două tehnologii cheie: Java Plug-in, care permite applet-urilor să ruleze în browsere și Java Web Start, care execută aplicații independente într-o rețea. JRE stă la baza tehnologiilor pentru platforma J2EE. JRE nu conține instrumente și utilitare cum ar fi compilator și debugger pentru dezvoltarea appleturilor și a aplicațiilor. JDK este un superset al JRE, conține tot ceea ce este în JRE plus instrumente precum compilatoare și debugger necesare pentru dezvoltarea appleturilor și aplicațiilor.

Platforma J2EE (Enterprise Edition), pentru aplicații distribuite, se folosește pentru dezvoltarea de aplicații și servicii web bazate pe componente cum ar fi servlet-uri, pagini JSP (*Java Server Pages*).

Platforma J2ME (Mobile Edition) - pentru dezvoltarea de aplicații pentru dispozitive mobile. Arhitectura J2ME definește configurații, profile și pachete opționale. Astfel, pentru fiecare tip de sistem embedded se alege categoria și profilul împreună cu pachetele opționale necesare. Configurațiile conțin bibliotecile de bază

ale limbajului Java pentru o categorie de dispozitive. Deasupra fiecărei configurații este definit un profil. Profilele definesc biblioteci specifice dispozitivelor mobile pentru interfața utilizator, rețea și stocarea datelor. Fiecare profil are propriul mediu de execuție și este realizat pentru un număr de dispozitive mobile similare. Pachetele opționale sunt extinderi ale profilelor, fiecare adaugă funcționalitate la unul sau mai multe profile. Până în prezent s-au definit două configurații: CDC (*Connected Device Configuration*) - pentru dispozitive mobile mai puternice (PDA evoluat, dispozitive de rețea) și CLDC (*Connected Limited Device Configuration*) - pentru dispozitive cu restricții de resurse (telefoane mobile, unele PDA-uri, pagere).

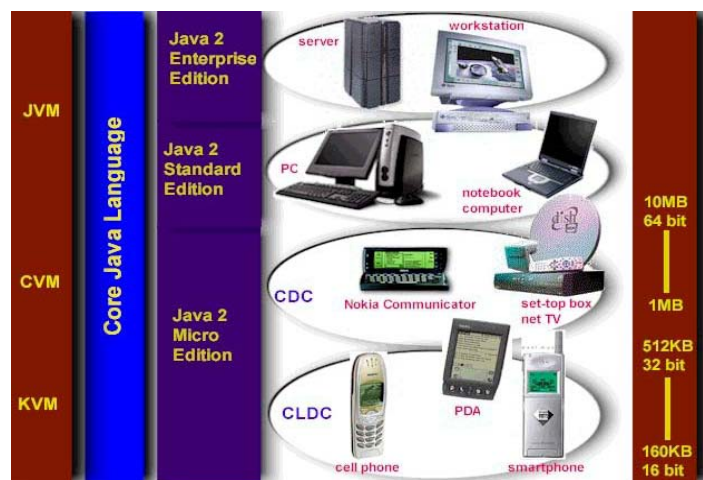


Fig. 3. Platformele Java și dispozitivele corespunzătoare

Având în vedere cele trei platforme Java există următoarele specificații JSR pentru SIP:

1) Specificația JAIN SIP

Specificația JAIN - Java pentru rețele integrate (*Java for Integrated Networks*) este definită de JSR32 [7]. Compania Sun Microsystems (preluată între timp de Oracle) este liderul specificației alături de alte companii (British Telecom, Cisco Systems, Ericsson, etc.). A fost prima specificație SIP standardizată de JCP ce poate fi utilizată pentru platforma J2SE și permite dezvoltarea de agent utilizator independent, proxy și aplicații servere redirecționale. JAIN SIP furnizează o interfață standardizată pentru protocolul SIP care poate fi utilizată în dezvoltarea aplicațiilor în domeniul comunicațiilor mobile.

JAIN a definit de asemenea interfața JCC (*Java Call Control*) pentru aplicații de control al apelului în telecomunicații. Interfața JCC poate fi mapată la SIP când aplicația SIP este construită peste implementarea JCC.

2) Specificația SIP Lite

Această specificație este definită de JSR 125 [8] și poate fi implementată atât pe platforma J2SE cât și J2ME și oferă un mediu de aplicații pentru dezvoltatorii care nu sunt experți în SIP. Compania Ubiquity este liderul specificației alături de alte companii (British Telecom, dynamicsoft, Hughes Software Systems, Nokia Corporation, etc.). Specificația SIP Lite a fost inițial dezvoltată pentru platforma J2SE dar, fiind suficient de mică, poate fi implementată și pe platforma J2ME. Motivația ce stă în spatele SIP Lite pentru platforma J2ME este furnizarea unui model obiect ce poate fi adecvat pentru dispozitivele de dimensiuni medii având mai multă putere de procesare și memorie decât telefoanele mobile, de exemplu PDA-uri și telefoanele SIP.

3) Specificația SIP Servlet

Această specificație este definită de JSR 116 [9], liderul specificației fiind compania dynamicsoft alături de 8x8, Ericsson, IBM, Nokia Networks, Nortel Networks, Siemens AG, Sun Microsystems și Ubiquity. Specificația SIP Servlet definește un mediu pentru execuția aplicațiilor SIP în rețea. Este implementată pe servere de aplicații care suportă SIP și, opțional, HTTP și J2EE. Servlet-urile SIP pot crea și accesa sesiuni, apeluri și tranzacționa date. Spre deosebire de servlet-urile HTTP, cele SIP pot interacționa cu SIP, aplicații și servere proxy.

4) Specificația SIP pentru J2ME

Această specificație este definită de JSR 180 [10], Nokia fiind liderul specificației alături de Cisco Systems, dynamicsoft, Motorola, Panasonic, Siemens AG, Sun Microsystems, Symbian, etc. Specificația definește o interfață pentru dispozitivele mobile cu resurse limitate. Acceptarea SIP ca protocol ales de arhitectura IMS din 3GPP subliniază importanța înțelegerii dispozitivelor mobile și a comunicației SIP. Specificația SIP pentru J2ME are la bază profilul CLDC al tehnologiei J2ME și, în general, va fi utilizată împreună cu profilul CDC. Dispozitivele client trebuie să suporte SIP pentru versiunea 5.0 a arhitecturii UMTS.

5) Specificația Java API pentru IMS (IMSAPI)

Această specificație este definită de JSR 281 [11], în grupul de experți ai specificației fiind China Mobile Communications, Ericsson AB, IBM, Infineon, LG Electronics, Lucent, Motorola, etc. IMSAPI este un pachet opțional pentru platforma J2ME. API pune la dispoziție un set de blocuri IMS pe care dezvoltatorul le poate utiliza în realizarea aplicațiilor IMS. API-ul dispune de un nivel ridicat de abstractizare, ascunzând detaliile de tehnologie și de protocol, fiind util și puternic

pentru non-experti în IMS pentru a crea noi aplicații. Cunoștințele necesare pentru a crea aplicații IMS simple sunt acelea de a ști cum se folosește un telefon pentru obișnuitele apeluri de voce și minimă experiență în programare pe o platformă MIDP (*Mobile Information Device Profile*). Specificația acoperă servicii IMS cum ar fi Push to Talk over Cellular, prezență, managementul documentelor XML.

Specificațiile Java pentru SIP nu redefinesc nici nu modifică protocolul SIP. Acestea doar definesc specificații API standardizate specifice mediului Java cu scopul dezvoltării de aplicații simplificate și de a asigura portabilitatea aplicațiilor pentru diferite implementări ale protocolului SIP.

Relația între diferite specificații SIP pentru Java API

Deoarece diferențele între funcțiile efectuate de diferite specificații SIP pentru Java sunt relativ mici, este important să se evidențieze clar aceste diferențe. Astfel, SIP Lite și Servlet sunt destinate dezvoltatorilor care nu sunt experți în protocolul SIP în timp ce JAIN are nevoie de experți atât în protocol cât și în Java. Specificația JAIN este dezvoltată pentru platformele J2SE și J2ME pentru dispozitive mobile, specificația Servlets este pentru J2EE iar specificația SIP Lite este atât pentru J2EE cât și pentru dispozitive mobile. În figura 4 se prezintă arhitectura IMS creată de 3GPP ce integrează diferitele specificații SIP pentru Java API .

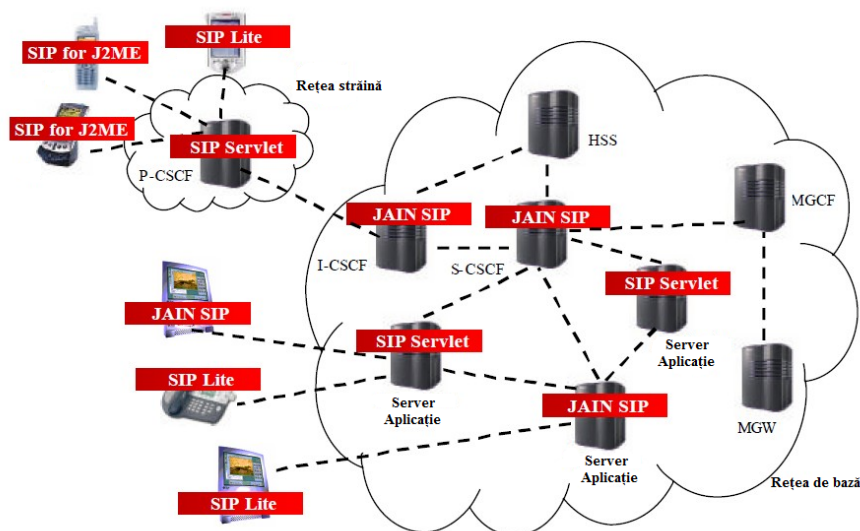


Fig. 4. Relația între arhitectura IMS și specificațiile Java SIP

Concluzii

În viitorul apropiat, telefonia mobilă poate beneficia de un set consistent de interfețe API de bază ce pot fi extinse cu altele suplimentare ce vor permite adăugarea de noi capabilități cum ar fi localizarea, servicii multimedia îmbunătățite și servicii de comunicație avansate bazate pe IMS. În prezent este în curs de dezvoltare un standard pentru convergența diverselor rețele de comunicații fixe și mobile cu comunicațiile media de voce și video bazate pe IP. Serverele de aplicații din arhitectura IMS oferă posibilitatea aplicației gazdă să furnizeze servicii utilizatorilor finali. Serverele IMS folosesc SIP ca protocol de bază pentru comunicarea cu alte nivele și entități din rețea. Convergența dintre Java și SIP oferă o platformă standard pentru serverele de aplicații IMS și permite astfel unui număr mare de dezvoltatori și furnizori să contribuie la dezvoltarea de noi servicii. Această platformă extinde modelul de programare Servlet cunoscut ca SIP Servlet API pentru crearea de elemente de rețea bazate pe SIP.

Bibliografie

- [1] **R. Spiers, N. Ventura**, "A converged IMS client for the IP Multimedia Subsystem", Southern Africa Telecommunication Networks and Applications Conference (SATNAC), 2010.
- [2] **P. Subramanian, B. PG**, "Convergence of Java EE and SIP in IMS AS", IEEE Int. Conf. On IP Multimedia Subsystem Architecture and Applications, Bangalore, December 2007.
- [3] **S. Tarkoma, R. Balu, J. Kangasharju, M. Komu, M. Kousa, T. Lindholm, M. Mäkelä, M. Saareto, K. Slavov, K. Raatikainen**, "State of the art in enablers for applications in future mobile wireless internet", HIIT Publications, 2004-2, September 2004.
- [4] **O.Rashid, P.Coulton and R.Edwards**, "Implications of IMS and SIP on the Evolution of Mobile Applications", IEEE Tenth International Symposium on Consumer Electronics, St. Petersburg, 2006.
- [5] 3GPP, www.3gpp.org
- [6] **R. Isukapalli, S. Benno, C. Park, P. M. Feder**, "Advanced IMS Client Supporting Secure Signaling", Bell Labs Technical Journal 12(4), pp. 49–66, 2008
- [7] **Sun Microsystems**, "JAIN SIP API Specification - JSR 32", <http://jcp.org/en/jsr/detail?id=32>.
- [8] **Ubiquity**, Sun Microsystems, "JAIN SIP Lite API Specification – JSR 125", <http://jcp.org/en/jsr/detail?id=125>.
- [9] **dynamicsoft**, Sun Microsystems, "SIP Servlet API Specification – JSR 116", <http://jcp.org/en/jsr/detail?id=116>.
- [10] **Nokia Corporation**, Sun Microsystems, "SIP for J2ME API Specification - JSR 180", <http://jcp.org/en/jsr/detail?id=180>.
- [11] **Sun Microsystems**, "IMS Services API", <http://jcp.org/en/jsr/detail?id=281>