

PIATTAFORME PER SERVIZI APPLICATIVI IN RETE

Le *Service Delivery Platform* costituiscono un valido strumento per supportare il rapido sviluppo, l'efficiente esecuzione e l'efficace gestione dei servizi applicativi nelle reti di telecomunicazione sia fisse che mobili. Queste piattaforme sono particolarmente interessanti in un momento in cui le tecnologie e i modelli di business legati alla fornitura dei servizi a valore aggiunto stanno evolvendo molto rapidamente. L'articolo presenta quindi le principali caratteristiche delle *Service Delivery Platform* e delle tecnologie di supporto.

1. INTRODUZIONE

La ricerca di modelli innovativi di fruizione e di nuovi modelli di business in relazione alla fornitura di servizi in rete favorisce e stimola sia l'evoluzione dell'architettura delle reti telematiche sia l'adozione delle nuove tecnologie che vengono costantemente introdotte.

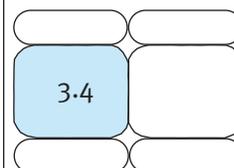
Nel settore IT (*Information Technology*) questa evoluzione ha per oggetto principalmente nuove piattaforme di livello applicativo, tipicamente basate sul paradigma *peer-to-peer*, e nuove modalità di fruizione della rete, tipicamente basate sul paradigma del Web2.0. La rete Internet, in particolare, evolve dal paradigma della "rete dell'informazione", a supporto cioè degli scambi informativi *one-to-one* per la comunicazione interpersonale (prevalentemente e-mail) e *one-to-many* per la fruizione di contenuti e servizi su siti e portali Web, ai paradigmi della "rete della conoscenza" e della "rete dei servizi" nei quali gli utilizzatori non sono più soltanto soggetti passivi fruitori di contenuti ma diventano invece soggetti attivi, e cioè autori o comunque fornitori

di contenuti, per esempio attraverso la condivisione di video (esempio, Youtube), immagini (esempio, Flickr), dati (esempio, Google), messaggistica ed informazioni di contesto (esempio, Skype, MS Messenger) e servizi o applicazioni (esempio, Facebook).

Nel settore TLC (*Telecommunications*), invece, l'evoluzione ha per oggetto principalmente la migrazione del cosiddetto "piano di controllo" delle reti telefoniche, tradizionalmente basato sull'architettura e sul protocollo *Signaling System #7 - SS7* [1], verso nuovi piani di controllo appoggiati sul *layer IP* (come per esempio nell'*IP Multimedia Subsystem - IMS* [2]), finalizzati a supportare sia l'integrazione delle reti e dei servizi su rete fissa e su rete mobile sia la realizzazione di servizi applicativi evoluti, spesso basati su paradigmi simili a quelli del mondo IT, all'interno di piattaforme e ambienti integrati per la fornitura di servizi (esempio, *Service Delivery Platform - SDP* o *Service Delivery Framework - SDF*). L'attenzione delle aziende che operano nel mondo dei servizi in rete è oggi focalizzata prevalentemente sull'integrazione tra il mondo IT e il



Pierpaolo Baglietto
Massimo Maresca
Francesco Moggia
Michele Stecca
Alberto Giordano



mondo TLC. Più specificatamente l'entrata in scena delle nuove realtà del mondo Internet, come Skype, YouTube o Google, ha portato gli operatori di TLC a ripensare alle proprie strategie in modo coordinato con Internet e non in contrapposizione ad essa. In altre parole la progressiva diminuzione dei ricavi provenienti dal traffico voce determinata dalla concorrenza derivante dalla liberalizzazione del mercato della fornitura dei servizi di telecomunicazione ha spinto gli operatori di TLC ad incrementare la propria offerta di servizi innovativi da un lato per limitare la riduzione dell'*Average Revenue per User* - ARPU, che con il solo servizio di trasporto sarebbe ridottissimo, e dall'altro per estendere la propria clientela e fidelizzare quella esistente. Vengono quindi introdotti servizi a valore aggiunto (*Value Added Service* - VAS) basati sull'integrazione di video, voce e dati come per esempio: chiamate multimediali, video-conferenze, *instant messaging* ecc..

In questo scenario evolutivo caratterizzato da una situazione di elevatissima variabilità compare un elemento infrastrutturale che ricopre un ruolo determinante nella fornitura dei servizi a valore aggiunto: la *Service Delivery Platform*. La SDP costituisce un elemento innovativo in rapido e continuo sviluppo ed incarna in modo specifico l'integrazione fra mondo IT e reti di telecomunicazione.

Nella prima sezione di questo articolo viene riassunta l'evoluzione delle reti telefoniche dal punto di vista dell'erogazione dei servizi. Vengono successivamente presentate le caratteristiche salienti delle SDP, elencati e categorizzati i servizi funzionali e applicativi che questo tipo di piattaforme forniscono ed infine si accennerà ad alcune delle tecnologie maggiormente utilizzate per la realizzazione delle SDP.

2. EVOLUZIONE DELL' EROGAZIONE DEI SERVIZI NELLE RETI TELEFONICHE

Per comprendere meglio come si è giunti alle attuali architetture per la fornitura dei servizi e quali "driver" hanno portato alla loro definizione, è opportuno ripercorrere brevemente l'evoluzione delle tecnologie utilizzate per l'erogazione dei servizi in rete partendo dalle reti telefoniche nell'era pre-IN (*Intelligent*

Network), in cui non era presente un particolare componente dedicato alla loro gestione, fino a giungere alle SDP convergenti.

Le caratteristiche peculiari della *Public Switched Telephone Network* - PSTN sono in estrema sintesi di essere progettata essenzialmente per il supporto del servizio telefonico e di non prevedere una gestione centralizzata dei servizi che erano invece gestiti da apposite applicazioni nei nodi terminali (switch telefonici) che offrivano dei servizi telefonici di base chiamati *Plain Old Telephone Service* - POTS (per esempio *voice mail*, *call waiting* ecc.). Questo richiedeva la replicazione degli apparati di controllo della rete e la replicazione degli apparati di tariffazione dei servizi, con la conseguente difficoltà di introdurre nuovi servizi data la limitata flessibilità del sistema.

A partire dalla metà degli anni '80 la rete PSTN cambia radicalmente grazie anche all'utilizzo del sistema di segnalazione SS7 (1) su canale comune (CCS, *Common Channel Signalling*) che ha introdotto la distinzione tra i livelli di trasporto, segnalazione e applicazioni. Nascono così le *Intelligent Network*, che rappresentano un punto di svolta rispetto al passato in particolare per quanto riguarda la fornitura dei servizi telefonici perché non prevedono più una gestione dei servizi decentrata e hard-coded nel sistema di *switching* ma centralizzata nel *Service Control Point* - SCP - e flessibile.

Purtroppo però le reti IN non permettevano ancora di raggiungere gli alti livelli di flessibilità richiesti dal mercato in quanto gli SCP erano *application-dependent* ovvero ogni nuova applicazione che si intendeva aggiungere richiedeva un sistema dedicato. Le *Advanced Intelligent Network* (AIN) superano parzialmente questo problema definendo SCP indipendenti dai servizi.

Parallelamente all'evoluzione della rete PSTN, la cui obsolescenza è segnata sia dall'invecchiamento tecnologico sia dalla mancanza di apertura verso il supporto di servizi a banda larga, si è assistito allo sviluppo della telefonia sulle reti a commutazione di pacchetto e in particolare al cosiddetto VoIP (*Voice over IP*). In questo ambito si colloca l'architettura *Soft-switch* e cioè quell'insieme di prodotti, protocolli e applicazioni che permettono ad un dispositivo di accedere a servizi di telecomunicazione su una rete IP.

Infine, vi è l'*IP Multimedia Subsystem - IMS* - (2) che rappresenta il futuro delle reti per le telecomunicazioni (attualmente gli operatori telefonici si trovano in un momento di migrazione verso questa architettura). IMS è la specifica standard del settore per le comunicazioni definita dal 3GPP (*3-rd Generation Partnership Project*) che descrive il modo in cui gli elementi di una rete interagiscono allo scopo di fornire servizi e applicazioni vocali, dati e video in qualsiasi combinazione, su qualsiasi rete fissa o mobile. Esso utilizza dei protocolli standard quali *Session Initiation Protocol - SIP* [3], *Diameter*, *Real Time Protocol - RTP*, e si basa sulla stratificazione orizzontale delle reti in cui i vari livelli di trasporto, di segnalazione (o controllo) e delle applicazioni sono distinti (Figura 1). In IMS i servizi applicativi vengono forniti dagli *Application Server* e da piattaforme che mettono disposizione degli utilizzatori un ambiente finalizzato a supportare la creazione e l'esecuzione efficiente di servizi applicativi: la *Service Delivery Platform - SDP*.

3. SERVICE DELIVERY PLATFORM

Con il passaggio dal modello verticale o "*stovepipes*" (in cui ogni diversa rete di accesso ha una struttura di gestione dedicata) a quello orizzontale di IMS (che permette di ridurre la replicazione delle funzionalità nella rete in quanto è presente un unico *layer* di controllo comune a tutte le differenti tecnologie di accesso), il modo di concepire la fornitura di servizi cambia completamente perchè la visione diventa di tipo convergente ed è possibile prevedere un'unica piattaforma - la SDP - che possa supportare vari servizi relativi anche a reti diverse.

3.1. Funzionalità

Nonostante non esista una definizione universalmente riconosciuta di SDP, è possibile identificare un insieme di funzionalità che essa fornisce agli sviluppatori di servizi:

- permettere la rapida realizzazione dei *Value Added Services - VAS* (riduzione del *time-to-market*); si va sempre più affermando l'utilizzo di *Service Creation Environment - SCE* di tipo grafico che permettono di sviluppare i VAS mediante operazioni di *drag&drop* di blocchetti

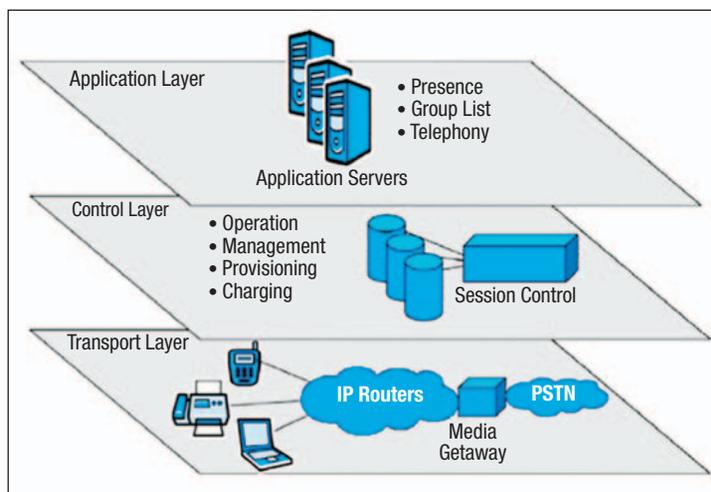


FIGURA 1
L'architettura a tre livelli di IMS. (Fonte: spie.org)

che rappresentano delle funzionalità di base altamente riutilizzabili;

- fornire un completo framework per il *deployment*, esecuzione, gestione e tariffazione dei VAS; favorire la portabilità dei servizi attraverso differenti fornitori (concetto "*write once, run anywhere*"); questo permette agli operatori di acquisire più libertà nella scelta dei *Network Equipment Provider - NEP*;

- fornire un meccanismo di astrazione dello strato di accesso; essendo la SDP un mezzo attraverso cui raggiungere la convergenza fisso-mobile, essa deve assicurare l'indipendenza dalla rete di accesso attraverso l'utilizzo di interfacce aperte che permetta agli sviluppatori di utilizzare le varie funzioni in modo uniforme;

- consentire una facile integrazione con gli altri elementi presenti nelle reti degli operatori con particolare attenzione per gli *Operational Support System - OSS* e i *Business Support System - BSS*;

- supportare la *Service Exposure* e cioè prevedere una modalità di fornitura di alcune funzionalità e/o servizi a terze parti che sia standard e sicura;

- effettuare *run-time monitoring, logging and reporting*; tracciare il comportamento della piattaforma attraverso un *Service Management Engine - SME* è molto importante per poter risolvere i malfunzionamenti e per effettuare un adattamento dei parametri della stessa di tipo *context-aware*;

- prevedere l'interoperabilità con i sistemi *legacy* ma anche essere estensibile per quanto

riguarda l'aggiunta di nuove tecnologie (importanza dell'utilizzo di interfacce standard).

3.2. Modello di riferimento per i servizi

Prima di passare ad esaminare l'architettura che può essere presa come riferimento per il funzionamento di una SDP, si ritiene opportuno chiarire il concetto di "servizio" perchè in questo ambito esso può assumere diversi significati a seconda del contesto a cui ci si riferisce. Si possono identificare tre differenti tipologie di servizi:

- Servizi Infrastrutturali: si occupano della gestione degli aspetti di più basso livello e sono gestiti dalla piattaforma;
- Servizi Abilitanti: si occupano di fornire delle funzionalità standard che possono essere utilizzate dagli sviluppatori di servizi applicativi; possono essere visti come dei mattoncini di base che vengono assemblati opportunamente per creare delle entità più complesse;
- Servizi Applicativi: questi rappresentano i servizi finali che vengono effettivamente utilizzati dagli utenti.

Il modello proposto prevede che lo sviluppatore definisca i Servizi Applicativi che verranno resi disponibili agli utenti finali mediante l'utilizzo combinato di alcuni Servizi Abilitanti con parti di codice peculiari di ogni specifica applicazione. I vari Servizi Applicativi saranno eseguiti e gestiti nella piattaforma la quale assicura il corretto ed efficiente funzionamento degli stessi grazie ai Servizi Infrastrutturali.

3.2.1. SERVIZI INFRASTRUTTURALI

I Servizi infrastrutturali risultano forniti in modo trasparente agli sviluppatori di Servizi Applicativi, che possono quindi concentrare la loro attenzione solo sulla logica applicativa, lasciando in carico alla piattaforma tutte quelle funzionalità di più basso livello che vengono ora elencate:

- *Scalability*: essa deve gestire opportunamente il *load balancing* delle richieste e prevedere un meccanismo automatico per l'aggiunta di nodi di elaborazione in modo che il sistema sia in grado di auto-riconfigurarsi senza dover sospendere la sua esecuzione.
- *Fault Tolerance*: le SDP devono continuare a funzionare anche in caso di malfunzionamenti in quanto sospendere l'erogazione dei servizi può causare ingenti perdite economiche per

l'operatore; è quindi necessario che la piattaforma fornisca un meccanismo automatico di *failure-detection* che individui gli eventuali guasti e che riorganizzi "a caldo" il bilanciamento del traffico sui nodi disponibili.

- *Low Latency/High Throughput*: questi parametri sono tipici del mondo delle telecomunicazioni e sottolineano il fatto che è molto importante che le richieste in entrata alla SDP siano servite in modo rapido.
- *Service Management*: deve essere realizzata una facile integrazione con i sistemi di OSS e BSS.
- *Security Management*: è richiesta un'opportuna gestione degli aspetti di *Authorization, Authentication e Accounting* - AAA.

Il supporto dei Servizi Infrastrutturali è imprescindibile in una SDP di qualità *Carrier Grade* ovvero che assicuri un'affidabilità pari al 99.999% (i famosi *five-nines*). Questa denominazione implica il fatto che la SDP riesca ad erogare i suoi servizi anche quando gli utenti raggiungono un numero considerevole.

3.2.2. SERVIZI ABILITANTI

I Servizi Abilitanti, o *Service Enabler*, sono quelli che vengono resi disponibili dalla SDP allo sviluppatore di Servizi Applicativi per consentire a quest'ultimo di contare su un insieme di funzionalità trasversali che possono essere assemblate secondo le esigenze specifiche di ogni applicazione e, dove possibile, riutilizzate, per creare le applicazioni che vengono proposte all'utente finale e che costituiscono i *Value Added Services*. Esempi di Servizi Abilitanti sono: *Session Enabler* (gestione sessioni utente), *AAA Enabler* (gestione aspetti di sicurezza, spesso bastato sul protocollo DIAMETER), *Presence Enabler* (gestione dello "stato/presenza" degli utenti), *Media Enabler* (gestione del mixing e della gestione di media differenti), *Messaging Enabler* (gestione unificata della messaggistica), *Charging Enabler* (gestione della tariffazione dei servizi che si divide in *online* e *offline charging*) ecc..

3.2.3. SERVIZI APPLICATIVI

3.2.3.1. Servizi User-to-User

Per quanto concerne le comunicazioni tra singoli utenti, si tratta di estendere le classi di servizi già disponibili per gli utenti come le tradi-

zionali chiamate, gli SMS, gli MMS ecc.. Alcuni servizi di esempio possono essere:

- i *Rich Voice Services*, nei quali si aggiunge alla classica chiamata telefonica la possibilità di scambiare contenuti come immagini o file di vario tipo. Una SDP dovrebbe anche essere in grado di gestire le *video-chiamate* oppure sessioni di *chat (Instant Messaging)*;

- il *Multi Device Service* permette all'utente chiamante di essere identificato dall'utente chiamato sempre con lo stesso identificativo (per esempio un SIP-ID) indipendentemente dall'apparecchio che esso utilizza per la comunicazione (telefono fisso, cellulare, IP-phone ecc.);

- il *Multi Ring Service* permette di distribuire simultaneamente le chiamate entranti su vari apparecchi;

- il *Follow-Me /Line Hunting Service* permette di decidere in che ordine devono suonare i differenti apparecchi (per esempio si può impostare che suoni per primo il telefono cellulare e, qualora non vi sia risposta, dirottare la chiamata sull'IP-Phone).

3.2.3.2. Servizi Multi-User

In questo scenario sono disponibili diversi servizi che prevedono il coinvolgimento di più utenti contemporaneamente:

- le Conferenze Multimediali che costituiscono un importante servizio che viene utilizzato dalle aziende per effettuare incontri a distanza e diminuire i relativi costi; è possibile inoltre classificare in questa categoria anche il servizio Mobile Gaming che, per sua natura, richiede il coinvolgimento di più utenti;

- il *Multi Channel Televoting System* che permette di gestire le operazioni di voto collegate, per esempio, ai programmi televisivi;

- il *Convergent Messaging* che è un sistema di messaggistica che gestisce tutti i vari formati (testuali, SMS, e-mail, *instant messaging*) attraverso cui gli utenti possono comunicare tra di loro (*buddy list*).

3.2.3.3. Servizi Server-to-User

Per quanto riguarda le applicazioni *Server-to-User*, risulta particolarmente interessante la categoria di servizi che viene definita *Dynamic Push Service* che è possibile grazie anche alla presenza di una funzionalità prevista dall'architettura IMS, anche se non necessariamente

supportata, che provvede alla *localizzazione geografica* dell'utente. Sottoscrivendosi a questo tipo di servizi, gli utenti possono ricevere dei contenuti in modo particolarmente personalizzato in quanto il loro invio si basa su molti fattori come la localizzazione geografica, lo stato di presenza, l'orario, profili di preferenze pre-caricate ecc.. Il tipo di contenuto che viene ricevuto può essere estremamente variabile e dipende dallo specifico servizio (per esempio, informazioni stradali, il filmato del gol della propria squadra, l'indirizzo di un determinato negozio ecc.). Gli operatori possono progettare i più diversi tipi di *Dynamic Push Service* basandosi su appropriate indagini di mercato (per un esempio si veda [8]).

3.3. Architettura di riferimento

Il problema degli standard nell'ambito delle SDP si riflette anche nella mancanza di un'architettura che definisca nel dettaglio le componenti che devono essere presenti nella piattaforma e le relative funzionalità.

Dalla consultazione della letteratura a disposizione, sembra ormai imporsi l'architettura proposta in [7] e mostrata nella figura 2 che, sebbene molto generica, viene riconosciuta come riferimento da molti fornitori. Essa include i seguenti elementi:

- *Network Abstraction Layer*: fornisce interfacce standardizzate verso i vari tipi di reti. Lo scopo principale di questo componente è

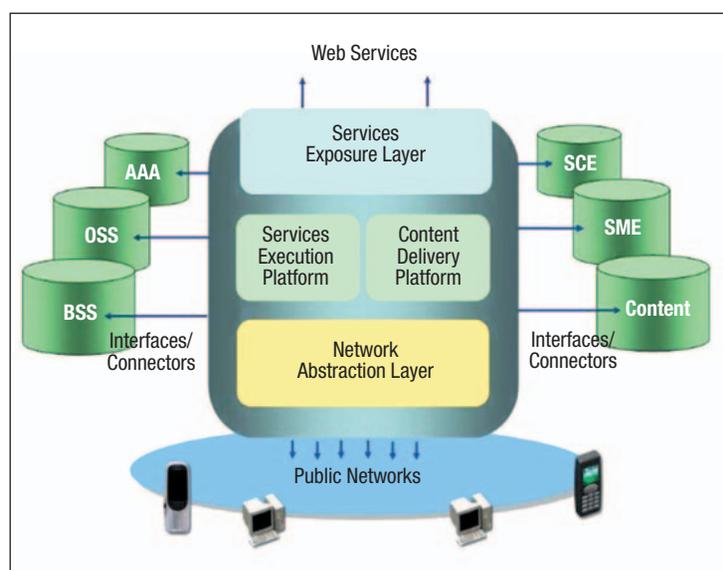


FIGURA 2
Architettura delle SDP. (Fonte: Rapporto Moriana [7])

quello di rendere l'accesso ai servizi indipendente dalla tecnologia di rete sottostante.

□ *Service Exposure Layer*: componente opzionale che serve a mettere a disposizione a terze parti alcune funzionalità. Esso permette agli operatori di "aprire" le proprie reti attraverso l'utilizzo di interfacce standardizzate e sicure.

□ *Content Delivery Platform*: componente che gestisce i contenuti multimediali degli apparecchi mobili.

□ *Service Execution Platform*: costituisce l'ambiente in cui sono presenti i servizi e in cui vengono eseguiti. Questa parte della piattaforma deve assicurare delle elevate prestazioni per poter raggiungere dei sufficienti livelli di "user experience".

4. TECNOLOGIE A SUPPORTO DELLA SERVICE EXECUTION PLATFORM

Per quanto concerne gli aspetti tecnologici, risultano particolarmente interessanti tre standard utilizzati nelle *Service Execution Platform* che sono stati definiti dal *Java Community Process* – JCP mediante la divulgazione di specifiche chiamate *Java Specification Request* – JSR:

- la specifica J2EE ([5]) descrive una piattaforma che utilizza un modello applicativo *multi-tier* e distribuito. La logica applicativa è suddivisa in componenti che rispecchiano differenti funzioni che devono essere svolte e che vengono installati talvolta in diverse macchine. J2EE è costituita da un insieme di specifiche (per esempio Java Transaction API-JTA, Java DataBase Connector-JDBC ecc.) che standardizzano i molti aspetti relativi alle applicazioni enterprise.

- La specifica JSLEE ([4]) descrive un'architettura che permette di creare delle applicazioni multiprotocollo orientate agli eventi (*Event Driven Applications*, EDA). L'ambiente gesti-

sce tutti gli aspetti legati alle prestazioni, alla concorrenza, alle transazioni, ai fallimenti e alla persistenza in modo tale da assicurare i due requisiti chiavi alle applicazioni orientate agli eventi: bassa latenza e alto *throughput*.

- Le applicazioni basate sulle SIP Servlet ([6]) sono implementate attraverso API che permettono di gestire i messaggi del protocollo SIP e viene effettuato il loro "deploy" all'interno di un *Application server* che ne dà supporto equivalente a quello fornito alle HTTP Servlet.

Una comparazione dei tre standard citati precedentemente è proposta nella tabella 1.

Vista l'estrema eterogeneità delle tecnologie proposte, si deduce che non esiste una scelta ottima valida in assoluto e, infatti, le diverse aziende manifatturiere si stanno muovendo in direzioni diverse per la scelta delle piattaforme di supporto per l'esecuzione dei servizi. Le motivazioni che stanno alla base di questa eterogeneità vanno ricercate nelle diverse esperienze e nei diversi modelli di business a cui fanno riferimenti le aziende produttrici di piattaforme ma anche nell'imaturità di alcune tecnologie.

5. I SERVIZI APPLICATIVI NELL'ERA DEGLI SMARTPHONE: GLI APPLICATION STORE

La recente diffusione degli *Smartphone*, ovvero dei dispositivi mobili forniti di elevate capacità computazionali, numerose periferiche, connessione ad Internet ecc. - il più famoso dei quali è l'*iPhone* prodotto da Apple - ha portato numerose novità sia sotto l'aspetto tecnologico che dei modelli di business.

In particolare gli *Smartphone* sono dei dispositivi dotati di sistema operativo su cui è possibile installare delle applicazioni che vengono rese disponibili in un apposito sito chiamato *Application Store*. Gli utenti finali visitano que-

TABELLA 1
Tabella comparativa degli standard a supporto dell'esecuzione dei servizi

	J2EE	JSLEE	SIP-SERVLET
Gestione efficiente degli eventi	NO	SI	Solo protocollo SIP
Gestione concorrenza	SI	SI	NO
Multiprotocollo	NO	SI	NO
Esistenza di prodotti open source	SI (JBoss, Sun GlassFish ecc.)	SI (Mobicents)	SI (Red Hat Comm. Platform, Saifin ecc.)

sto negozio virtuale, identificano l'applicazione di loro gradimento (che può essere gratuita oppure a pagamento) e, dopo averla scaricata, la installano sul proprio terminale dove l'applicazione viene eseguita. Il *Service Execution Environment* è quindi parzialmente (e in alcuni casi completamente) localizzato nel dispositivo mobile dell'utente e questo costituisce un importante cambiamento di paradigma che ha delle importanti ripercussioni sulla progettazione delle SDP (per esempio si deve gestire opportunamente un aumento di carico sul *Service Exposure Layer*).

È interessante sottolineare il fatto che le applicazioni disponibili negli *Application Store* possono essere sviluppate dai fornitori degli Smartphone ma nella grande maggioranza dei casi sono sviluppate da "terze parti" (aziende oppure privati) che realizzano dei servizi sfruttando gli appositi SDK (*Software Development Kit*) che vengono messi a disposizione. Emerge quindi la necessità di definire modelli di business che tengano conto dei diversi attori che entrano in gioco ovvero il gestore dell'*Application Store*, lo sviluppatore dell'applicazione e l'utente finale che scarica e utilizza l'applicazione. Il modello proposto dall'*Application Store* di Apple, che rappresenta il punto di riferimento in questo ambito, prevede che il ricavo ottenuto dalla vendita di una determinata applicazione sia suddiviso secondo il seguente schema: 70% del ricavo allo sviluppatore e 30% ad Apple.

Esempi di *Application Store* attualmente esistenti sono, oltre *AppStore* di Apple, *Ovi* di Nokia, *Android Market* per i dispositivi basati sul sistema operativo Android di Google e *360* di Vodafone.

6. CONCLUSIONI

In questo articolo è stata ripercorsa la storia dell'evoluzione della fornitura dei servizi telefonici partendo dalla più datata architettura della rete PSTN, in cui erano disponibili pochi semplici servizi, fino ad arrivare alla rete IMS (la cui definizione da parte degli enti di standardizzazione non è ancora giunta ad una versione definitiva) che dovrebbe ampliare l'insieme dei servizi disponibili grazie alla stratificazione orizzontale della rete che permette di velocizzare la fornitura

di servizi convergenti (e quindi non solo telefonici).

In questo nuovo scenario ricoprono un ruolo cruciale le *Service Delivery Platform* per le quali non è ancora disponibile la descrizione di una architettura standard universalmente riconosciuta ma per le quali si possono identificare una serie di requisiti comuni come favorire il veloce sviluppo di nuovi servizi applicativi (al fine di abbassare il *time-to-market* dei servizi), assicurare la facile integrazione con i sistemi già presenti presso gli operatori (con particolare attenzione ai sistemi di OSS e BSS), fornire delle funzionalità trasversali come la scalabilità e la tolleranza ai guasti (fondamentali per poter definire una SDP *carrier grade*) e il rispetto dei più importanti standard tecnologici. Per quanto riguarda quest'ultimo punto, si è discusso della presenza di una certa varietà di possibilità (J2EE, JSLEE, SIP Servlet) e di come la tecnologia più adatta da scegliere possa variare a seconda del tipo di servizi che si intende fornire.

Oltre alla classificazione dei numerosi standard presenti per l'implementazione delle varie componenti delle SDP, rimane ancora aperta la questione di categorizzare precisamente l'insieme dei servizi che si possono realizzare e fornire mediante queste piattaforme software (applicazioni telefoniche, Internet, *MashUp* ecc.) e quale collocazione possano trovare sul mercato. Per quanto riguarda quest'ultima questione, più legata agli aspetti di business che a quelli tecnologici, è stato introdotto il tema degli *Application Store* che si sta affermando prepotentemente sulla scia del successo dell'*Apple Store* e che dimostra come il coinvolgimento mediante appropriati schemi di *revenue-sharing* delle cosiddette "terze parti" possa essere un fattore fondamentale per la realizzazione di servizi applicativi che siano più appetibili per gli utenti finali.

Bibliografia

- [1] Cisco: *SS7 Fundamentals Cisco Tutorial*. [disponibile sul Web all'indirizzo: http://ciscosystems.com/univercd/cc/td/doc/product/tel_pswt/vco_prod/ss7_fund/index.htm]
- [2] 3 GPP Group: *IMS Releases*, [<http://www.3gpp.org/article/ims>]

- [3] Internet Engineering Task Force - IETF, Session Initiation Protocol (RFC 3261).
- [4] JAVA Community Process: JAIN SLEE 1.1 Specification, Final Release. <http://jcp.org/en/jsr/detail?id=240>, Luglio 2008.
- [5] JAVA Community Process: JAVA Platform Enterprise Edition v1.4 Specification, Final Release. <http://jcp.org/en/jsr/detail?id=151>, Novembre 2003.
- [6] JAVA Community Process: SIP Servlet 1.1 Specification, Final Release. <http://jcp.org/en/jsr/detail?id=289>, Agosto 2008.
- [7] Moriana Group: *Service Delivery Platform and Telecom Web Services*. Giugno 2004.
- [8] Metafacile: *Guida turistica GPS*. [http://www.metafacile.com/ita/itinerari_gps_franciacorta.php]

PIERPAOLO BAGLIETTO si è laureato in Ingegneria Elettronica presso l'Università di Genova dove nel 1994 ha conseguito il Dottorato di Ricerca in Ingegneria Elettronica ed Informatica. È attualmente Professore Associato presso la Facoltà di Ingegneria dell'Università di Genova dove è docente dei corsi di Calcolatori Elettronici e di Sistemi Informativi e Servizi in Rete. I suoi interessi di ricerca sono centrati sui sistemi distribuiti, i sistemi di monitoraggio e gestione delle reti di nuova generazione e le piattaforme per lo sviluppo e l'erogazione di servizi.
E-mail: p.baglietto@cipi.unige.it

MASSIMO MARESCA si è laureato in Ingegneria Elettronica presso l'Università di Genova dove nel 1986 ha conseguito il Dottorato di Ricerca in Ingegneria Elettronica ed Informatica. Ha svolto attività di ricerca presso IBM T.J. Watson Research Centre di New York (USA) e presso l'International Computer Science Institute di Berkeley (USA). È attualmente professore ordinario presso l'Università di Padova e Direttore del Centro di Ricerca Interuniversitario sull'Ingegneria delle Piattaforme Informatiche (CIPi). I suoi interessi di ricerca sono centrati sulle piattaforme per la composizione e l'orchestrazione di servizi e sui sistemi di monitoraggio e gestione delle reti a banda larga.
E-mail: m.maresca@cipi.unige.it

FRANCESCO MOGGIA ha conseguito la laurea in Ingegneria delle Telecomunicazioni nel 2000 e il titolo di Dottore di Ricerca in Ingegneria Elettronica ed Informatica presso l'Università di Genova nel 2006. Ha svolto attività di ricerca presso il National Institute of Standards and Technology (Gaithersburg, MD) dal 2003 al 2005 contribuendo allo sviluppo del progetto open source Mobicents. Dal 2005 al 2007 è stato ricercatore in Telecom Italia, contribuendo alla JSR-289 (Servlet Sip) e JSR-240 (JSLEE). È attualmente responsabile tecnico di M3S srl. I suoi interessi di ricerca attuali sono le Service Delivery Platforms, le architetture SOA e REST e le tecnologie per i protocolli di messaggistica e VoIP.
E-mail: f.moggia@m3s.it

MICHELE STECCA si è laureato in Ingegneria Informatica all'Università degli Studi di Padova nel 2007. Attualmente è impegnato come dottorando di ricerca presso il CIPi (Centro Interuniversitario di Ricerca sull'Ingegneria delle Piattaforme Informatiche - Università di Padova e di Genova). In passato ha partecipato al progetto OPUCE (Open Platform for User-centric service Creation and Execution) finanziato dalla Comunità Europea nell'ambito del Sesto Programma Quadro. I suoi interessi di ricerca sono centrati sulle piattaforme per la composizione e l'orchestrazione di servizi (Mashups) e sulle piattaforme per la fornitura di servizi in ambito telecom.
E-mail: m.stecca@cipi.unige.it

ALBERTO GIORDANO si è laureato in Ingegneria Elettronica presso l'Università di Genova nel 1977. Ha successivamente svolto attività di ricerca presso l'Università di Karlsruhe e presso la Fraunhofer Gesellschaft di Karlsruhe. Dal 1984 ha coperto il ruolo di ricercatore ed attualmente è Professore Associato presso la Facoltà di Ingegneria dell'Università di Genova dove è docente dei corsi di Fondamenti di Informatica e Sistemi di Elaborazione. I suoi interessi di ricerca riguardano le architetture dei sistemi di elaborazione, le reti di telecomunicazione wired e wireless ed i sistemi orientati all'erogazione di servizi.
E-mail: a.giordano@cipi.unige.it