



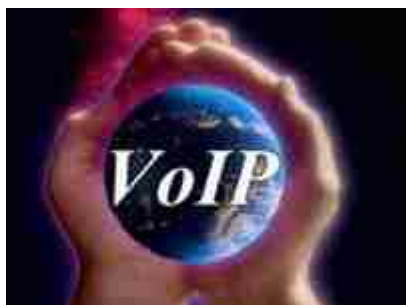
Introduzione allo standard SIP e possibili integrazioni con RFID e wiMax

Nino Guarnacci
Educational Services
K-Tech



Voice Over IP

come insieme di standard e tecnologie usate per trasportare la voce su qualsiasi rete di trasmissione per dati fondata su Ip, da una piccola rete locale all'intera Internet.



I servizi di telefonia Internet sono una tecnica di emulazione della rete telefonica tradizionale che usa Voip per trasmettere la voce, ma usa anche altri protocolli per gestire chiamate da e verso i normali numeri di telefono fissi, cellulari e relativi servizi (segreteria telefonica etc).

Secondo una recente indagine promossa da Siemens, entro il **2009** la crescita delle linee Voip residenziali sarà tale da garantire il raggiungimento di una porzione di mercato **pari ad un quarto** rispetto alle linee telefoniche tradizionali.



VoIP Base

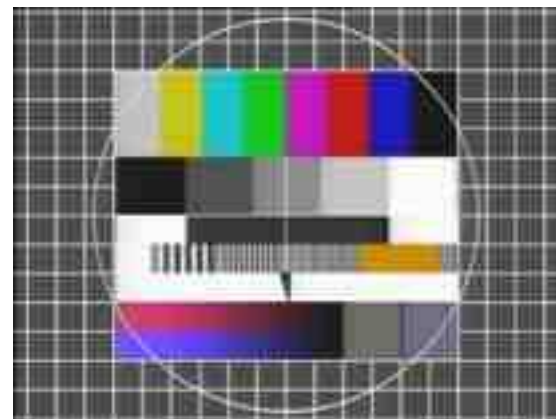
Il cuore della tecnologia Voip è il protocollo **RTP (Realtime Transport Protocol)**, progettato per trasportare tra due punti qualsiasi della rete un flusso di dati, ricavato digitalizzando segnali audio o video.

Per utilizzare RTP è necessario un telefono speciale denominato telefono IP (può essere Hardware o Software).

Si possono utilizzare anche degli **ATA (Analog Telephone Adaptor)**, Box o Gateway.

Durante la comunicazione, i telefoni IP si mantengono sincronizzati grazie al protocollo **RTCP (Realtime Transport Control Protocol)** che adatta fra l'altro dinamicamente il livello di compressione del suono al grado di congestione della rete.

I telefoni IP "comunicano" fra di loro solo se usano lo stesso standard, se ciò non avviene devono passare attraverso dei Gateway che offrono servizi di traduzione.





Quality of Service

Tecnologia, parametri e caratteristiche che misurano la qualità di una comunicazione Volp al fine di renderla confrontabile alla rete tradizionale.

- ▶ Una connessione Internet deve avere una **banda minima garantita** (MCR Minimum Cell Rate) di 32Kbps
- ▶ QoS assegna priorità in caso di bassa banda ai **pacchetti RTP**
- ▶ Access point e schede **Wi-Fi** non supportano QoS



Session Initiation Protocol

come “*signaling protocol*” operante al livello delle applicazioni.

E' utilizzato per attivare, gestire e chiudere *sessioni Multimediali* del tipo:

- ▶ Telefonia via Internet
- ▶ Conferenze
- ▶ Istant Messaging
- ▶ Applicazioni generiche che coinvolgono audio video e dati.
- ▶ Gaming Services





SIP

I messaggi SIP consentono di attivare sessioni e di condurre lo stato successivamente. Il protocollo SIP supporta sessioni e chiamate **punto-punto e multipoint**.

Sip può associare allo stesso numero **più telefoni IP**

Si possono instaurare e rilasciare le comunicazioni utilizzando i seguenti elementi SIP:

- ▶ Posizione dell'Utente
- ▶ Funzionalità dell'Utente
- ▶ Disponibilità dell'Utente
- ▶ Instaurazione della Chiamata
- ▶ Gestione della Chiamata





RFC - SIP

Il Protocollo SIP, su cui si basa il documento ***RFC (Request For Comment) 2543***, è un protocollo descritto testualmente nell'architettura multimediale generale dell'IETF (Internet Engineering Task Force).

L'approccio IETF è quello di creare un'architettura a livelli in cui le varie funzionalità vengono realizzate tramite protocolli ad elevata ottimizzazione.

L'IETF include anche i protocolli:

- ***RSVP (Resource Reservation Protocol - RFC 2205)***
- ***RTP (Realtime Transport Protocol - RFC 2326)***
- ***SAP (Session Announcement Protocol)***
- ***SDP (Session Description Protocol - RFC 2327)***

Le funzionalità del protocollo SIP sono indipendenti e quindi non dipendono da nessuno di questi protocolli.



SIP Component

- **SIP Server**

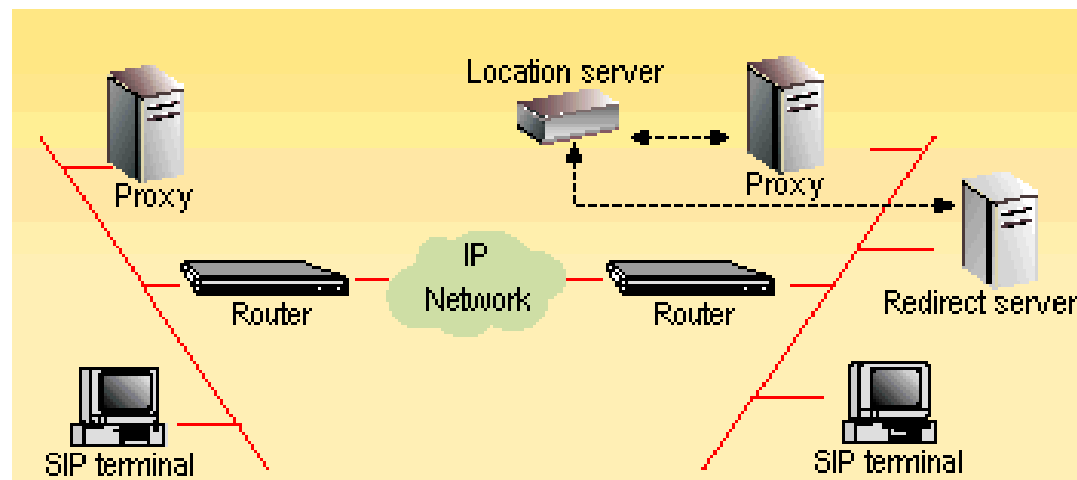
Agenti Utente + Server di Rete
(proxy e/o redirectione)

- **Registrar Server**

"elenco telefonico" centrale gestito dal provider del servizio
che si incarica di indirizzare la chiamata in arrivo verso il telefono Ip abbinato.

- **Location Server**

Per conoscere la sua ubicazione, il registrar server interroga il location server a cui il telefono è associato in quel momento. Se il telefono Ip non è associato a nessun location server, e quindi è spento o scollegato, la chiamata in arrivo può essere instradata verso una segreteria telefonica.





Java Specification Request SIP

Suddivisione delle JSR SIP in merito alle 3 piattaforme JAVA



J2ME

JSR 180 SIP for J2ME



J2SE

JSR 32 JAIN Sip



J2EE

JSR 116 Sip servlet



JSR 180 “SIP for J2ME”

Sip per j2me definisce un set di interface per supporti a connettività limitata come telefoni cellulari e pda completamente integrate con un profilo generico **MIDP** (*Mobile Information Device Profile*).

L'obiettivo è di trattare la comunicazione SIP attraverso factory comuni alle **SocketConnection e HttpConnection**, semplificandone quindi l'uso e lo sviluppo:

- **Gestione integrata degli headers in requests**
- **Responses acknowledgments automatizzati**
- **Supporto a SIP dialogs conversazionali**



Sip diventerà un protocollo molto importante nel futuro dei supporti **IP Mobile**.

MIDP 2.0 già in grado di supportare la comunicazione via TCP e UDP, necessita di standard dettati dalla rigidità della **gestione della sicurezza** che richiede SIP.



JSR 32 “JAIN Sip”

Jain sip come prima jsr proposta per la standardizzazione delle comunicazione **SIP via Java**.

Interfacce standard per la realizzazione di applicazioni diverse:

API per la piattaforma j2se che permettano lo sviluppo di applicazioni standalone, che fungano da **User Agent Client (UAC)**, **User Agent Server (UAS)**, **proxy** e **registrar** server.

API per la realizzazione di un **ambiente di esecuzione** permettendo quindi il supporto e la comunicazione verso UA in un ambiente **event-driven** :

- **JAIN Service Logic Execution Environment (SLEE).**

Le specifiche jain permettono la costruzione di applicazioni in grado di comunicare via SIP in modalita asincrona, utilizzando dei codici identificati per la **correlazione dei messaggi delle singole sessioni utente**.

Definisce inoltre vari oggetti factory che permettono la rapida creazione di oggetti **request, response ed header field specifici**, permettendone quindi una successiva customizzazione in ambienti diversi.



SIP Packages

JAIN come implementazione java delle specifiche standard SIP:

Packages

- ☉ **General package**
– Definisce le interface architetturali, transazionali e gestori degli eventi.
- ☉ **Address package**
– Contiene dei Wrapper URI generici e le interface di SIP URI e TEL URI
- ☉ **Message package**
Interface necessarie ai message Request e Response
- ☉ **Header packages**
– Definisce le interface per gli Header standard nella comunicazione



SIP Message Type

Il body del SIP Message contiene la cosiddetta “[session description](#)”.

JAIN SIP definisce il formato comune dell'oggetto, permettendo di essere una String o un Object type definito a livello di [Session Description Protocol \(SDP\)](#).

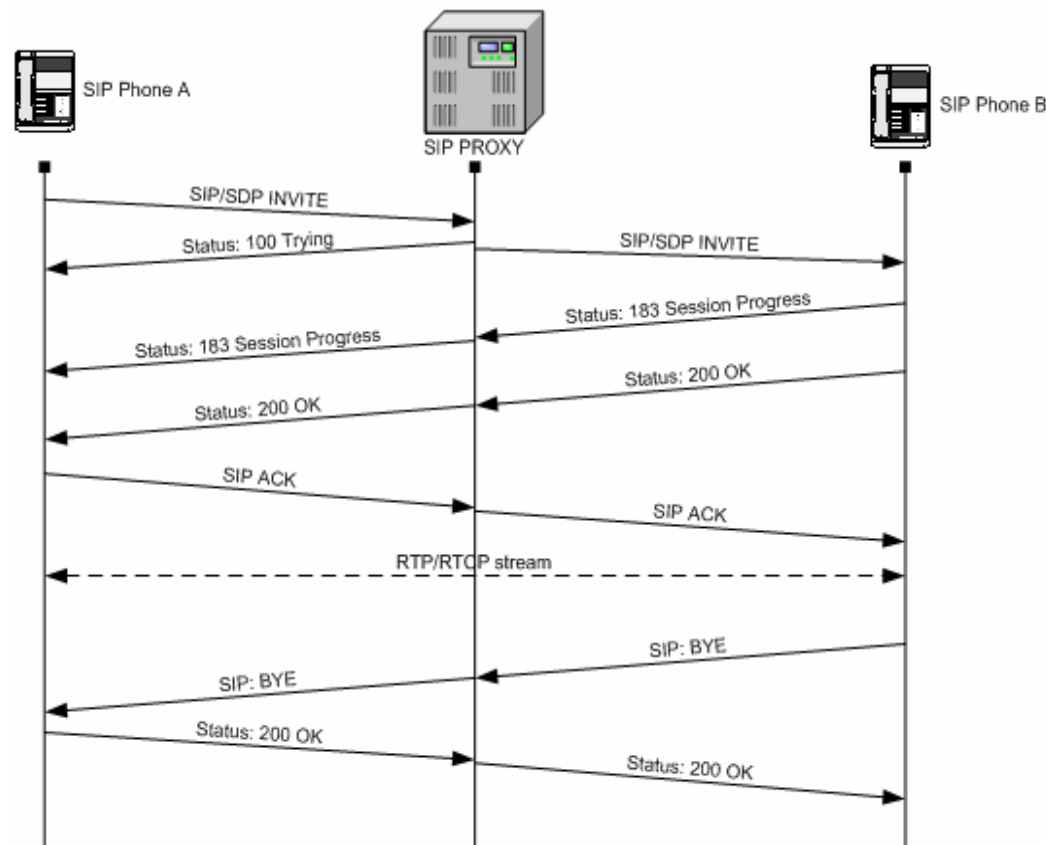
Request Message Type definiti dal sip protocol:

- ▶ **INVITE** : Richiesta di invito a partecipare ad una sessione
- ▶ **BYE** : Avviso di abbandono della sessione da parte di un partecipante
- ▶ **CANCEL** : Richiesta di cancellazione della ricerca utenti e sessioni attive
- ▶ **OPTIONS** : Richiesta di informazioni sui partecipanti alla sessione in merito alla loro disponibilità dei supporti media.
- ▶ **ACK** : Conferma di ricezione richiesta ed accettazione
- ▶ **REGISTER** : Richiesta di registrazione da parte di un partecipante verso il Sip Server



SIP Response Code

- 1yz Informational
 - 100 Trying
 - 180 Ringing (processed locally)
 - 181 Call is Being Forwarded
- 2yz Success
 - 200 ok
- 3yz Redirection
 - 300 Multiple Choices
 - 301 Moved Permanently
 - 302 Moved Temporarily
- 5yz Server failure
 - 500 Server Internal Error
- 6yz Global Failure
 - 600 Busy Everywhere
- 4yz Client error
 - 400 Bad Request
 - 401 Unauthorized
 - 482 Loop Detected
 - 486 Busy Here





JSR 116 “Sip servlet”

SIP servlet API definiscono un **ambiente di esecuzione** per applicazioni SIP based. Implementate all'interno di un **application server** che ne da supporto equivalente a quello http, attraverso **API di sviluppo e standard di packaging e deploy**.

SIP servlet per la costruzione e la gestione di applicazioni che fungano da UAC, UAS, proxy e registrar server rendendo trasparente allo sviluppatore gli **aspetti non essenziali e complessi di SIP**.

La principale differenza tra applicazioni Sip servlet e non server-based application, è quella di avere a disposizione un **ambiente di autocontenimento** delle risorse in grado di **gestire per noi il ciclo di vita** delle stesse, secondo specifiche standard, offrendo quindi ai fini pratici un ambiente molto simile a quello già largamente collaudato per il mondo **web based**.

Sip servlet definisce un **deployment descriptor standard** simile a quello usato per le web application classiche, permettendo di definire il flusso dell'applicazione e il mapping tra request e resource “**sip.xml**”.



Sip & HTTP Servlet

La **versione 1.0 delle SIP Servlet** api è stata pubblicata nel febbraio 2003. J2EE propone la tecnologia Java Servlet per la realizzazione di web application. Java HttpServlet per lo sviluppo di applicazioni client server su protocollo HTTP,

Le SIP servlet utilizzano le funzioni di base delle java servlet con **specifiche funzioni SIP**.

Sip servlet molto simili alle Servlet HTTP, così da velocizzare il processo di apprendimento nella realizzazione di applicazioni SIP based da parte di **sviluppatori già web enabled**.

```
public class SimpleSIPServlet extends SipServlet {  
  
    protected void doMessage(SipServletRequest req)  
        throws ServletException, IOException  
    {  
        SipServletResponse res = req.createResponse(200);  
        res.send();  
    }  
}
```



Sip & HTTP Servlet

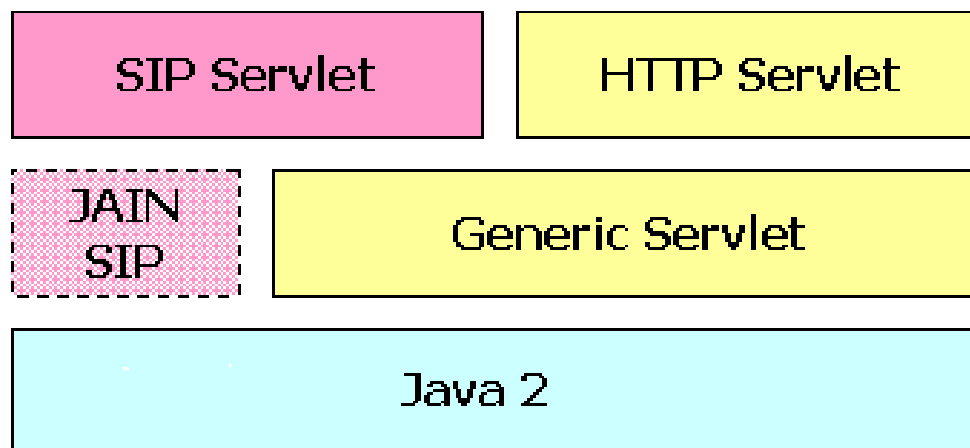
SIP Servlet e HTTP Servlet in comune:

Ereditano da una **superclasse**, che ne fornisce le funzioni di base HttpServlet e SipServlet.

I metodi vengono sovrascritti e implementati con il **prefisso doXXX**.

Il **ciclo di vita** di questi oggetti viene gestito dal container:

- **init**,
- **destroy**,
- **gestione della sessione utente e dei suoi attributi**.





Sip Response

Risposte multiple:

La SIP servlet è in grado di ritornare *più risposte ad una singola richiesta* a differenza di un ambito request/response tipico della servlet http.

```
protected void doInvite(SipServletRequest req);
```

```
protected void doAck(SipServletRequest req);
```

```
protected void doOptions(SipServletRequest req);
```

```
protected void doBye(SipServletRequest req);
```

```
protected void doCancel(SipServletRequest req);
```

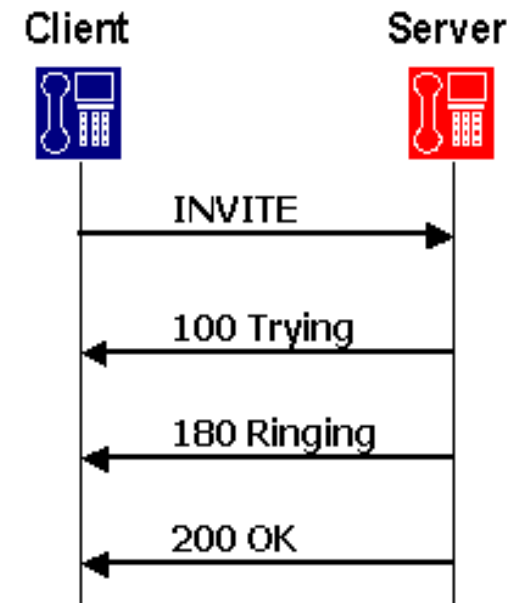
```
protected void doSubscribe(SipServletRequest req);
```

```
protected void doNotify(SipServletRequest req);
```

```
protected void doMessage(SipServletRequest req);
```

```
protected void doInfo(SipServletRequest req);
```

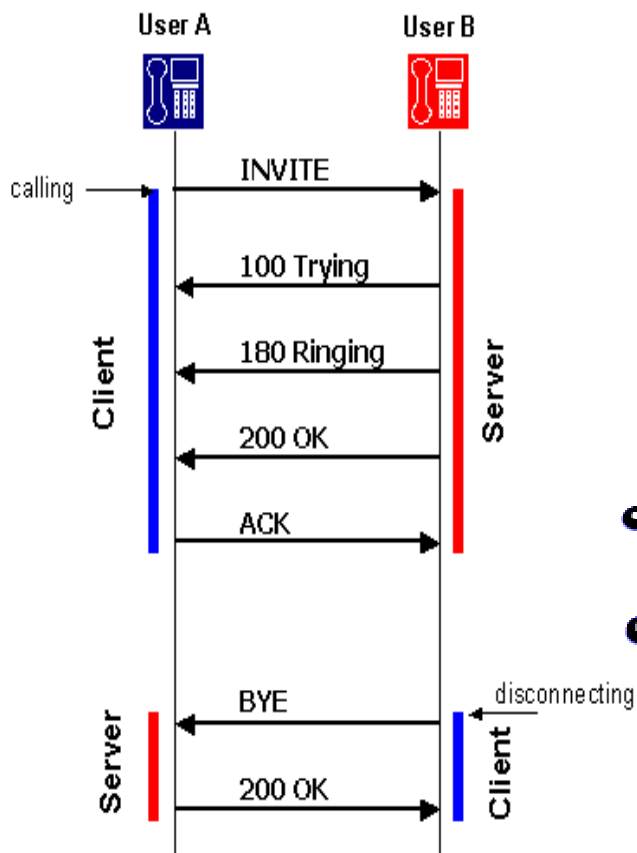
```
protected void doPrack(SipServletRequest req);
```





UAC vs UAS

Una delle maggiori funzionalità che SIP offre, è la disponibilità ad avere entità **client e server su ogni endpoint della sessione**, offrendo quindi la possibilità di avere una comunicazione bidirezionale.



```
protected void doProvisionalResponse(SipServletResponse res);
protected void doSuccessResponse(SipServletResponse res);
protected void doRedirectResponse(SipServletResponse res);
protected void doErrorResponse(SipServletResponse res);
```

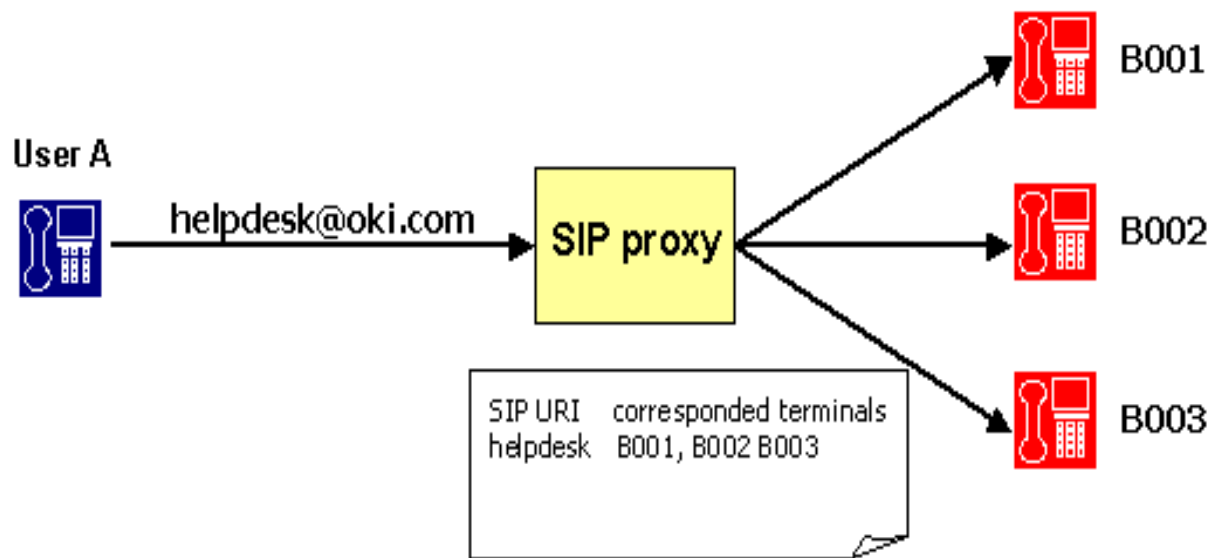
- La funzionalità client di un endpoint viene definita **UAC** (**User Agent Client**),
- mentre la funzionalità server viene definita **UAS** (**User Agent Server**).



Proxy SIP

Un'altra funzione che contraddistingue la tecnologia SIP è la possibilità di eseguire un *forking sulla richiesta* .

Ovvero la possibilità di fare da *proxy simultaneo o sequenziale*.





Sip Message

La struttura di un SIP message è simile a quella http.

In SIP, tuttavia, soltanto una bassa quantità di informazioni è immagazzinata nel corpo del messaggio poiché utile alla **comunicazione in tempo reale**.

Body:

SDP (Session Description Protocol)

Come protocollo per la definizione di sessioni multimediali tra terminali, definito dalla RFC2373.

IM (instant message) body.

Messaggi scritti ed inviati dall'utente

```
INVITE sip:itou@oki.com SIP/2.0
```

Starting Line

```
Via: SIP/2.0/UDP 10..0.100;branch=z9hG4bK1234
Max-Forwards: 70
To: <sip:tanaka@oki.com>
From: <sip:itou@oki.com>;tag=123456
Call-ID: 3d45f59a12b54
CSeq: 1 INVITE
Contact: sip:10.2.0.100:5060
Content-Type: application/sdp
Content-Length: 100
```

Header Field

```
Blank Line
```

Separator

```
v=0
o=- 5894032 5894032 IN IP4 10.2.0.100
S=SDP Media Session
... The rest is omitted.
```

Message Body



Sip Container

Il ruolo del Container delle nostre SipServlet è quindi quello di definire e gestire aspetti di:

- ***Application Management***

- Funzioni di application management, ciclo di vita ed inizializzazione.

- ***SIP Massaging***

- Funzione di parsing dei messaggi in entrata e delivering degli stessi alle giuste SipServlet. Invio dei messaggi creati dalle nostre applicazioni e creazione automatica dei rispettivi header.

- ***Utility Functions***

- Funzioni di gestione delle sessioni delle factory e delle funzionalità di proxing.



sip.xml

Così come per le web application un descrittore di deployment permette di istruire il container nei confronti della nostra applicazione SIP, mappando quindi *method type alla rispettiva Sip Servlet*.

```
<servlet-mapping>
  <servlet-name>registrar</servlet-name>
  <pattern>
    <equal>
      <var>request.method</var>
      <value>REGISTER</value>
    </equal>
  </pattern>
</servlet-mapping>
```

Il *ciclo di vita* delle SIPServlet come nelle web application è interamente gestito dal servlet context e configurabile dai system administrator.



RFID

Radio frequency identification (RFID) è un modo generalizzato per descrivere un sistema in grado di trasmettere l'identità univoca di una entità (oggetto, persona, animale) utilizzando le onde radio.

Tale tecnologia viene classificata come sistema di “**automatic identification**”, tra cui vi sono anche la tecnologia del **codice a barre** e alcuni sistemi di **scansione biometria** come ad esempio gli scanner della retina, che impegnano comunque l'utilizzatore ad un processo manuale di riconoscimento diversamente dall' RFID.

L'identificazione avviene mediante un “**tag**” che contiene l'identificativo e uno o più lettori “**reader**”.



Tipicamente un reader è formato da una o più antenne che emettendo onde radio ricevono di contro il segnale dal tag (**backscatter**).



Tag Type

La base architetture è abbastanza semplice, un **trasponder** (microchip e antenna) e un **reader** che si occupa di leggere le informazioni e passarle ad un sistema di middleware in grado di sfruttare tali informazioni per eseguirne della “**business logic**”.

Active Tag

Si intende l'utilizzo di tag muniti di **batteria**, **antenna** ed un supporto memoria “**read e write**”, in grado di offrire un “read range” di circa **20 -100 metri**, con un costo medio che va da **8 a 40 euro** per tag in base alle sue caratteristiche.

Passive Tag

Si intende l'utilizzo di tag senza nessun supporto batteria e nessun segnale in trasmissione attiva, ma semplicemente un microchip ed antenna inseriti all'interno di supporti diversi. Offrono un costo piuttosto contenuto (**30 centesimi di euro** circa) e cmq hanno un “**read range**” piuttosto contenuto fino ad un massimo di **10 metri**. Possono operare su basse frequenze ma anche ad alte frequenze **UHF (ultra-high frequency)**.

Un classificazione valida per i diversi tipi di tag presenti:

- ◆ **Class 0:** tag passivo, di tipo read-only programmato una sola volta in fase di creazione del CHIP.
- ◆ **Class 1:** tag passivo, di tipo read-only programmabile una sola volta, senza nessuna forma di memoria volatile.
- ◆ **Class 2:** tag passive con più di 65 kb di memoria “read-write”.
- ◆ **Class 3:** tag semi passive, con più di 65 kb di memoria “read-write” e supporto batteria per incrementare il “read-range”.
- ◆ **Class 4:** tag attivo, corredato di batteria e sistema di broadcast integrato per la comunicazione verso i reader.
- ◆ **Class 5:** tag attivi in grado di comunicare con tag di classe 5 e con altri sistemi (reader).



RFID Market

Uno degli usi più diffusi dell'RFID è munire le varie utenze con card-tag per il pagamento di **biglietti bus, metro, per l'apertura dei varchi elettronici** all'interno delle grandi aziende, e in fase sperimentale anche come strumento alternativo ai **buoni pasto**.



In ambito automobilistico questa tecnologia viene comunemente usata per inibire l'accensione dell'auto senza la corretta key-code dove all'interno è presente un tag RFID (**car-immobilizer**).

Comunemente presente anche all'interno di **società di trasporto e spedizioni**, per il tracciamento delle merci e la gestione dei punti di raccolta.



Tag passivi vengono utilizzati anche per censire e controllare specie di **animali**, attraverso l'installazione di un chip sottopelle.

In Italia la **Merloni Elettrodomestici** ha messo sul mercato un lavatrice in grado di calcolare automaticamente il programma di lavaggio da utilizzare in base agli indumenti presenti all'interno del cestello e muniti di RFID tag.



Java RFID

Sun Java System RFID come soluzione sun per la comunicazione e la gestione degli “**EPC**” (**Electronic product code**) presenti all’interno dei tag rfid ed acquisiti dai reader.

Composto da due sottosistemi:



RFID event manager:

che si occupa di comunicare con i vari readers attraverso una architettura event-driven.

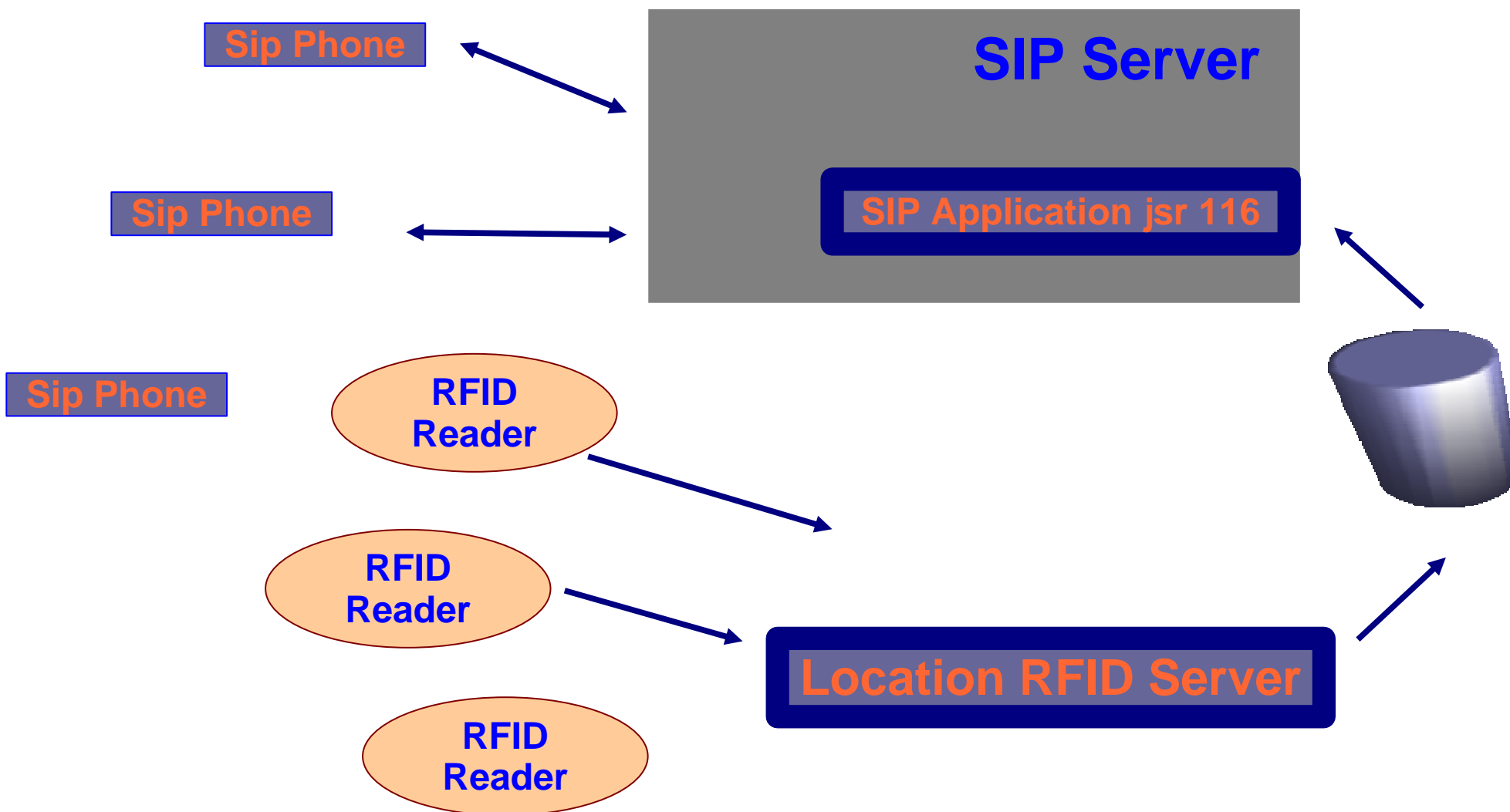


RFID Information Server:

responsabile del salvataggio e dell'accessibilità ai dati acquisiti attraverso delle logiche di filtering impostate in fase di messa in produzione.

JSR 257 Contactless Communication API come richiesta di specifiche standard in materia RFID, per la comunicazione verso read-only tag e bi-directional tag mediante la piattaforma **j2me**.

```
<AutoidEvent>
  <EventId>AAAABBBBCCCCDDDD0000111120021106130434</EventId>
  <SensorId>AAAABBBBCCCCDDDD00001111</SensorId>
  <TagId>000011112222333344445555</TagId>
  <TagId>000011112222333344445554</TagId>
  <Timestamp>2002-11-06T13:04:34-06:00</Timestamp>
</AutoidEvent>
```



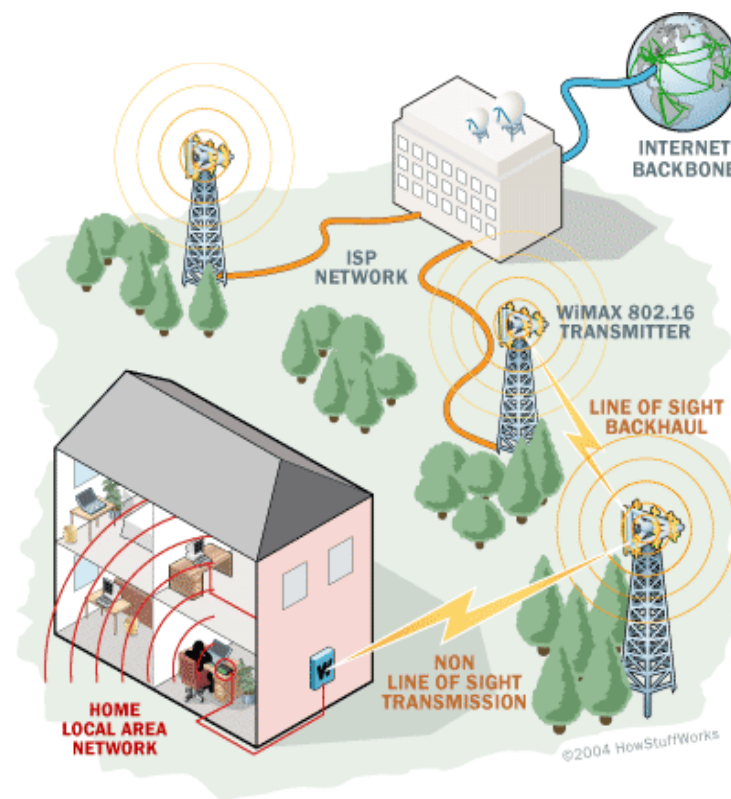
DEMO



WiMAX

WiMAX (Worldwide Interoperability for Microwave Access), come tecnologia wireless a basso costo che fornisce connessioni a banda larga con una portata massima di 50 Km anche in zone non ancora cablate, ben superiore alle poche centinaia di metri di Wi-Fi, e una larghezza di banda che può spingersi fino a 75 Mbps.

Tra alcuni anni il neo standard potrà porsi come alternativa economica alle linee xDSL e agli altri tipi di accesso ad Internet su cavo, incluse le soluzioni di connettività, cablate o wireless, oggi proposte per coprire l'ultimo miglio.



Wi-Fi, come tecnologia di rete locale, Wi-MAX come MAN, rete di area metropolitana.



WiMAX come

- Valida come tecnologia di ultimo miglio per la consegna dei servizi all'utente finale.
- In Paesi più sviluppati il WiMax, come tecnologia per la larga banda nelle aree dove l'x-DSL ha un basso livello di penetrazione a causa della bassa densità di popolazione.
- Nei Paesi in via di sviluppo dove il tradizionale servizio telefonico voce/dati non ha avuto grande penetrazione nemmeno nelle aree urbane.
- Per la pubblica amministrazione, come soluzione efficiente, flessibile ed economica per dotarsi di una rete privata metropolitana.
- Per gli operatori di rete fissa, WiMax come tecnologia parallela all'x-DSL nella copertura delle aree rurali, mentre per gli operatori alternativi rappresenterà un modo per dotarsi a costi relativamente bassi di una propria infrastruttura di rete.
- Portable Internet, cioè la possibilità di rimanere collegati ad internet anche in condizioni di mobilità dell'utente attraverso la rete.





Java Italian Portal

www.javaportal.it
nino.guarnacci@javac.it



Sip Server
<http://www.bea.com/>



Sip Softphone
<http://www.xten.com/>



Wireless Network Community
<http://www.ninux.org>

Special thanks to F. Marini e P. Pesci