Linux Day – 25 ottobre 2014

Giuseppe Fabiano

“Utilizzo e amministrazione di Asterisk”

fabianosoftware.it

@beppefabiano

PBX

Private Branch Exchange

E' una centrale telefonica per uso privato, utilizzata nelle aziende per fornire una rete telefonica interna.

Perchè un PBX?

Funzioni principali di un PBX:

- Risparmio sul numero di canali di fonia da attivare sulla rete pubblica
- Comunicazione diretta tra le utenze interne senza utilizzare la rete pubblica
- Trasferimento delle chiamate
- **Servizi e funzionalità aggiuntivi (interni ed esterni)**

Qualche esempio...

- Menù vocale (IVR, Interactive Voice Response)
- Segreteria telefonica unificata
- Documentazione del traffico in entrata ed uscita
- Automatic Call Distribution (ACD)

Molto altro, ma soprattutto:

- **Computer Telephony Integration (CTI)**

Computer Telephony Integration

Tecnologia che permette di interfacciare
un sistema telefonico
con un sistema informatico aziendale.

Computer Telephony Integration

Le interazioni tra sistema telefonico ed informatico avvengono tipicamente:

- mediante interrogazioni ad un **database**
- mediante **interfacce API**

Quale PBX?

Si possono individuare due tipi di soluzioni PBX:

PBX proprietario

Basato su hardware specifico, modifiche al software costose o impossibili

Esempi:

Siemens, Samsung, Ericsson/Aastra, ...

PBX software, open source

Basato su hardware generico, software con codice aperto, gratuito e modificabile

Esempi:

Asterisk, Elastix, Freeswitch, ...

PBX Open Source

Implementazione libera (GPL) di un software PBX,

con le stesse funzioni di altri sistemi proprietari,

ma con un costo inferiore ed una maggiore flessibilità.

Asterisk

Principali caratteristiche:

- codice sorgente in C, licenza GPL
- utilizza hardware generico in ambiente linux
- esistono varie distribuzioni ”pronte all'uso”
- estremamente modulare e flessibile
- non è soltanto un ”centralino”, ma un vero e proprio ”framework” per applicazioni CTI

Asterisk: non solo VoIP

Asterisk supporta tutti i principali protocolli VoIP:

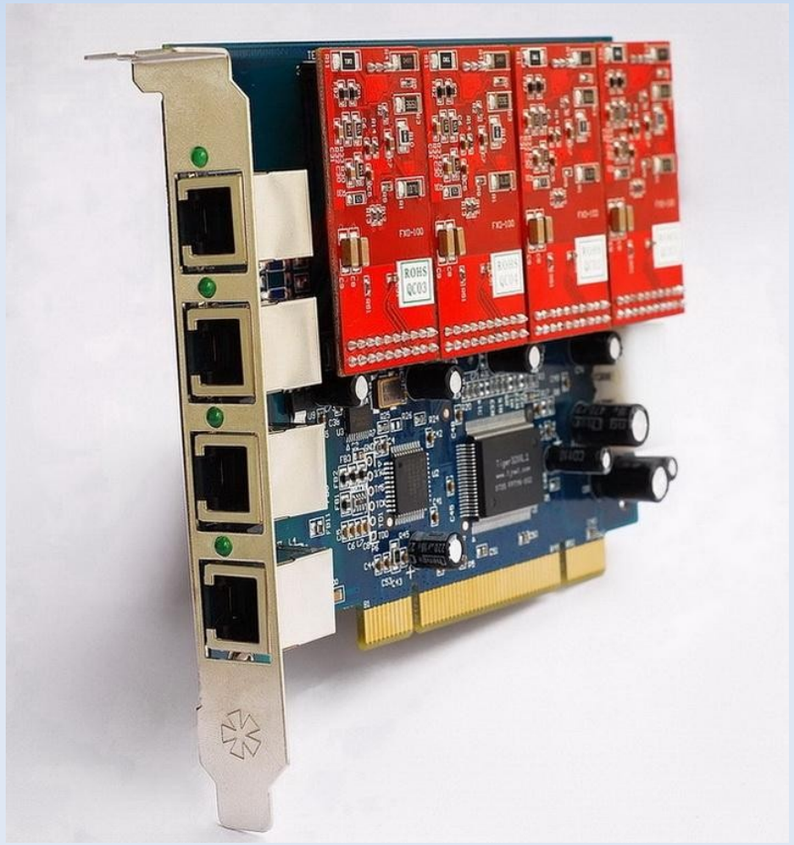
- **SIP, H.323, IAX**

Ma anche le tradizionali linee digitali e analogiche:

- **ISDN, PSTN**

Asterisk: non solo VoIP

Esempio di hardware per il supporto di linee analogiche:



Digium TDM400P

- One (1) - four (4) analog ports
- PCI or PCI Express bus architectures
- Combine line (FXO) and station (FXS) modules
- Accepts up to four (4) single-port modules
- Optional DSP-based carrier grade echo cancellation module

La struttura modulare permette di configurare la scheda con interfacce FXO ed FXS secondo le necessità.

Asterisk: non solo VoIP

Esempio di hardware per il supporto di linee ISDN:

Patton SmartNode™ 4630 Series

- 3/5 ISDN BRI So ports, 4/8 low-bandwidth voice or T.38 fax calls
- CLIP/CLIR, hold, transfer, and much more
- Web-based management, SNMP, command line interface, & auto-provisioning for automated configuration & SW upgrades
- Two 10/100 Ethernet ports, access router with NAT, Firewall, PPPoE, DHCP, DynDNS & VPN with IPsec
- **SIPv2, H.323v4, ISDN, DSS1, QSIG, T.38, fax bypass**

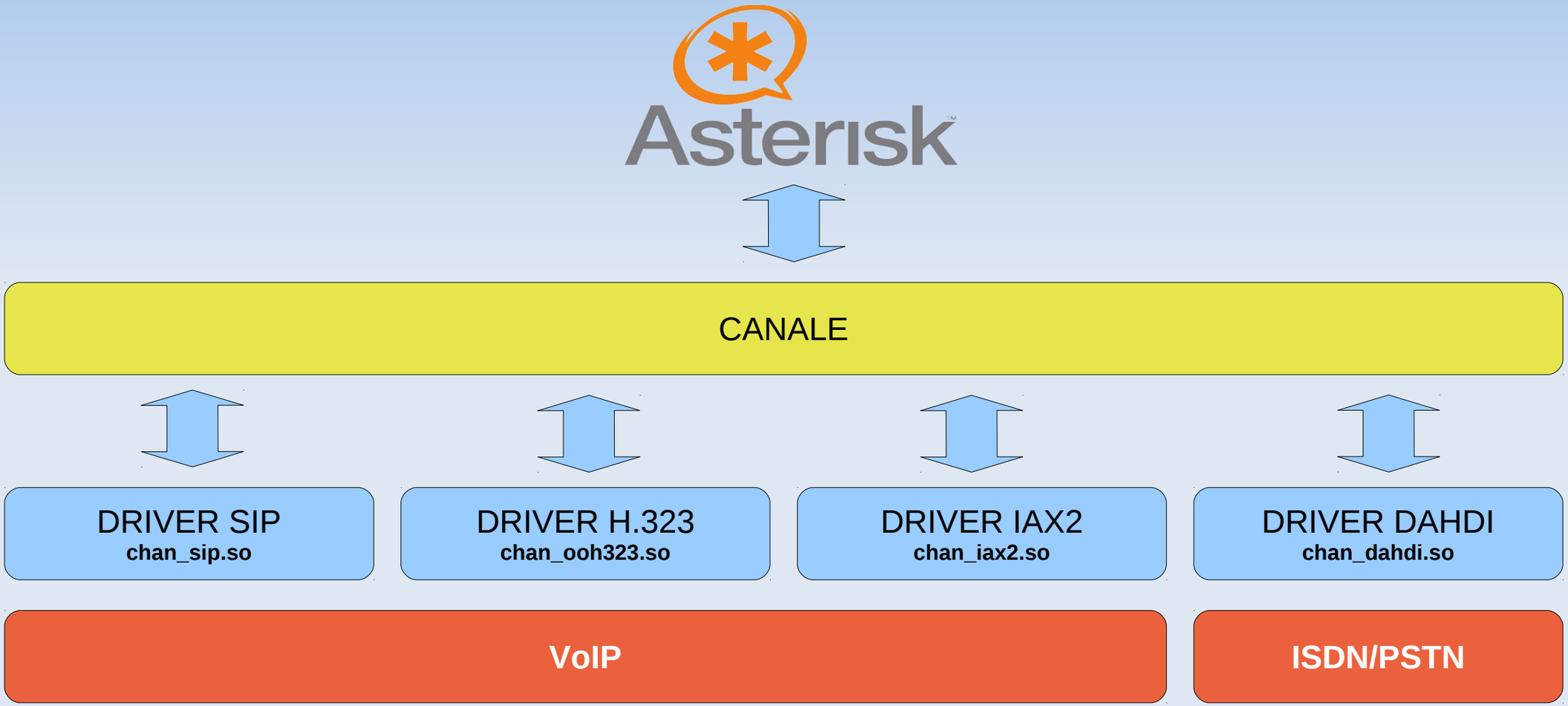
Asterisk: il concetto di canale

Il **canale** rende trasparenti per Asterisk i dettagli relativi alla comunicazione con una specifica tecnologia o protocollo.

Ogni tecnologia o protocollo supportato da Asterisk deve disporre di un proprio **”driver di canale”**

E' un esempio di struttura **modulare** e **flessibile**.

Asterisk: il concetto di canale



Asterisk: esempi di canale

- canale SIP
- canale H.323
- canale IAX2
- canale DAHDI (ISDN, PSTN)

Ogni canale possiede un file di configurazione specifico, di solito in ***/etc/asterisk***

Vediamo un esempio: **sip.conf**

Struttura di sip.conf

[general]

language=it_IT

[SIP-1001]

type=friend

host=dynamic

secret=1234

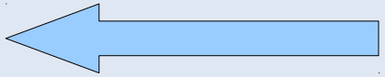
context=office

[SIP-1002]

...

Parametri generali di configurazione del canale

Definizione del dispositivo ”**SIP-1001**”



Password autenticazione dispositivo

Contesto che gestirà le chiamate in ingresso

Definizione di un altro dispositivo ”**SIP-1002**”

Registrazione dispositivo SIP

Host: 192.168.X.X

Username: SIP-1001

Password: 1234

Verifichiamo se il dispositivo è registrato correttamente:

```
$ asterisk -r
*CLI> sip show peers
```

Altre tipologie di canale

Altri tipi di canale si configurano attraverso uno specifico file di configurazione: **h323.conf**, **iax.conf**, ecc.

Per ogni dispositivo collegato deve essere definito un ”**contesto**” per le chiamate in ingresso.

La struttura ”**modulare**” caratterizza ogni aspetto funzionale di Asterisk, e non soltanto la comunicazione sui ”**canali**”.

Gestione dei moduli

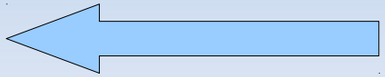
All'avvio Asterisk carica dinamicamente una serie di moduli, ognuno dei quali implementa determinate funzionalità.

Si tratta fisicamente di file **.SO (shared object)** presenti nella cartella **/usr/lib/asterisk/modules**

Il caricamento dinamico dei moduli è gestito attraverso il file di configurazione **modules.conf**, presente in **/etc/asterisk**.

modules.conf

[modules]

autoload=yes  Carica automaticamente tutti i moduli in **/usr/lib/asterisk/modules** ...

noload => chan_modem.so  ... tranne quelli specificati (es. obsoleti)

Tipologie di moduli

Ciascun “**modulo**” di Asterisk appartene ad una specifica **area funzionale**:

- Applications
- Bridging
- Call detail recording (CDR)
- Channel event logging (CEL)
- Channel drivers
- Codec translators
- Format interpreters
- Dialplan functions
- PBX
- Resource

Il prefisso del nome identifica la tipologia (es. **chan_sip, app_dial, res_odbc**)

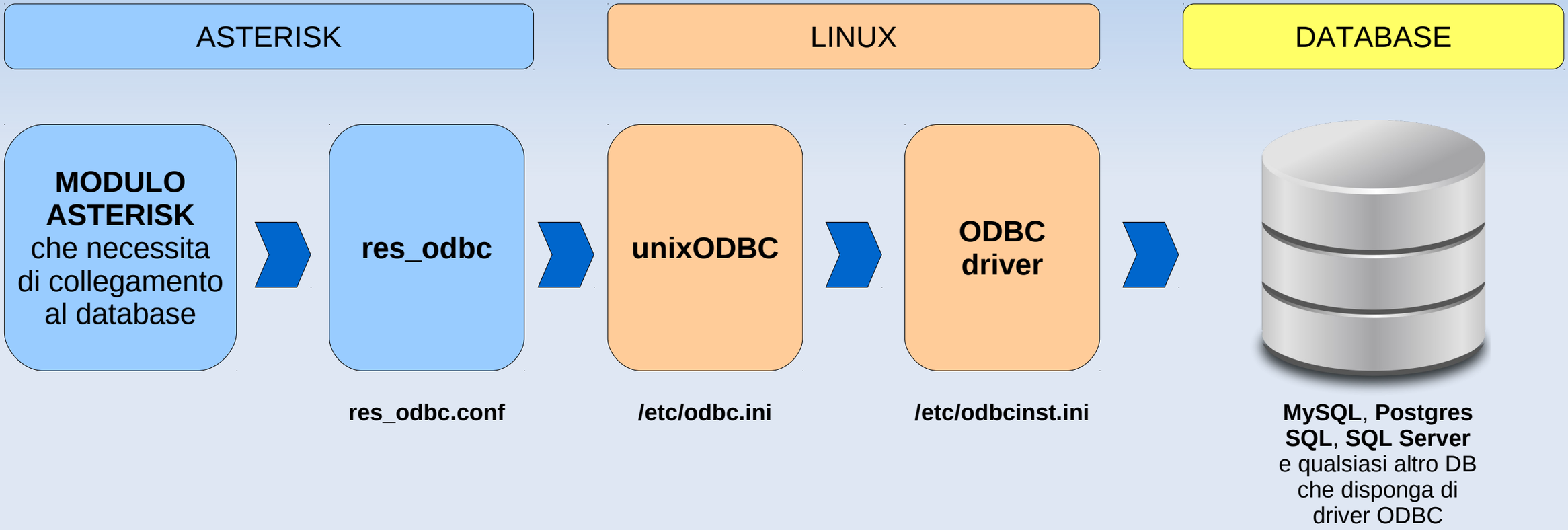
res_odbc

Attraverso il modulo **res_odbc** Asterisk è in grado di interagire con varie tipologie di database esterni, utilizzando il layer ODBC.

A cosa può servire?

- Configurazione di Asterisk su tabelle invece che su file
- Gestione strutturata dei log chiamate (CDR)
- **Computer Telephony Integration (CTI)**

Struttura del collegamento ODBC



Il layer ODBC astrae la particolare tecnologia di database utilizzata.

Ogni implementazione di una particolare tecnologia di database è confinata all'interno del driver ODBC.

Asterisk al lavoro: il dialplan

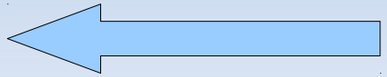
Il **dialplan** è il cuore del funzionamento di Asterisk:

attraverso **script**, controlla il flusso delle chiamate e definisce il comportamento del sistema in risposta a particolari eventi.

Il dialplan si configura in **/etc/asterisk/extensions.conf**

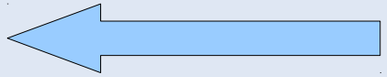
Struttura di extensions.conf

[general]  Impostazioni generali del dialplan

[global]  Definizione delle variabili globali

[context1]  Definizione del contesto **”context1”**

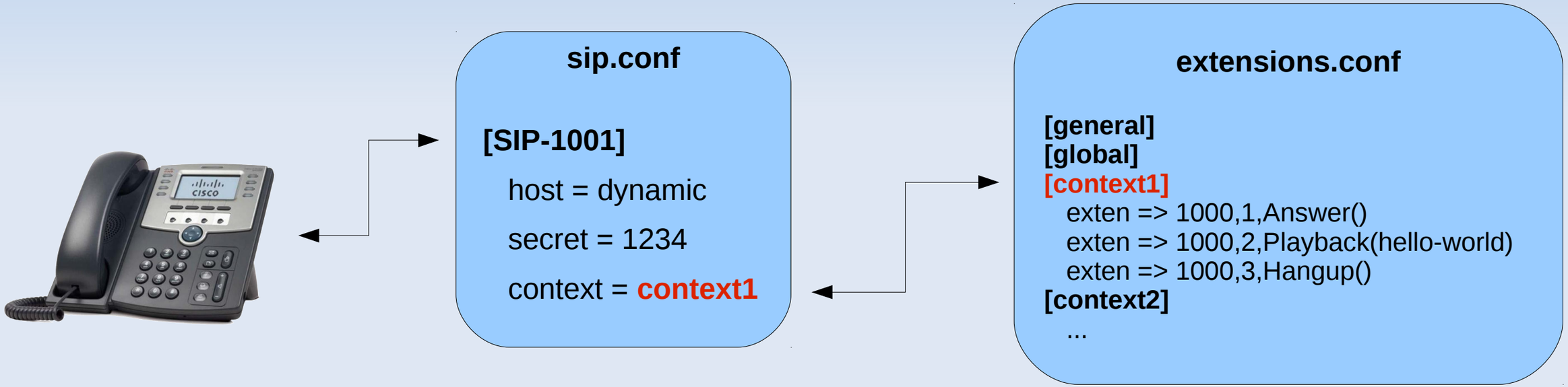
```
   exten => 1000,1,Answer()
   exten => 1000,2,Playback(hello-world)
   exten => 1000,3,Hangup()
```

[context2]  Definizione di un altro contesto

...

Contesto

Il **contesto** è il **punto di ingresso** nel dialplan per un dispositivo:

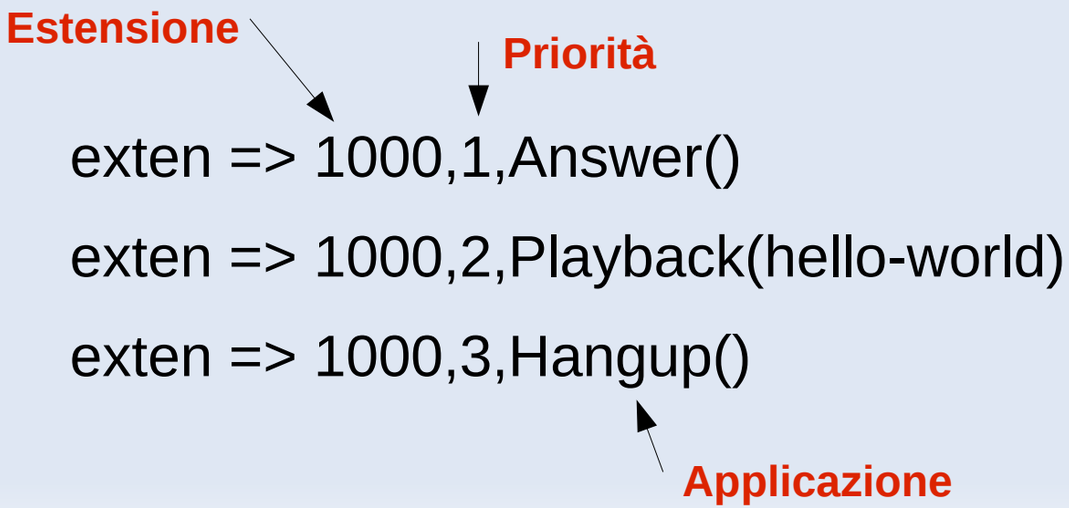


Usando i contesti è possibile implementare **politiche di sicurezza**.

Estensione

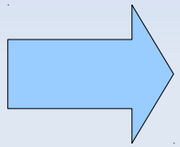
In Asterisk l'**estensione** è una serie univoca di passi, ognuno contenente una **priorità** ed un'**applicazione**, con i quali Asterisk gestirà la chiamata entrante.

Definiamo nel dialplan l'estensione ”**1000**”:



Miglioriamo la scrittura

```
[context1]
    exten => 1000,1,Answer()
    exten => 1000,2,Playback(hello-world)
    exten => 1000,3,Hangup()
```



```
[context1]
    exten => 1000,1,Answer()
    same => n,Playback(hello-world)
    same => n,Hangup()
```

Applicazione Dial()

Scenario

- due dispositivi SIP registrati nello stesso contesto (SIP-1001 e SIP-1002)

Obiettivo

- inviare e ricevere chiamate tra i due dispositivi

```
exten => _100Z,1,Answer()
    same => n,Playback(stai-chiamando)
    same => n,SayDigits(${EXTEN})
    same => n,Dial(SIP/SIP-${EXTEN})
    same => n,Hangup()
```

Pattern matching

Lettura del numero chiamato

Collegamento dei due dispositivi

Esempio: menù vocale (IVR)

Scenario

- tre dispositivi SIP registrati (SIP-1001, SIP-1002, SIP-1003)

Obiettivo

- il terzo dispositivo chiama l'estensione **1011**, risponde un menù vocale che lo mette in contatto con **1001** se digita 1, oppure con **1002** se digita 2
- digitando **9**, la conversazione termina
- digitando un numero non valido viene pronunciato un messaggio di errore
- se non si digita nulla, viene ripetuto il messaggio di benvenuto

Esempio: menù vocale (IVR)

```
[office]

   exten => 1011,1,Goto(voice-menu,start,1)

[voice-menu]

   exten => start,1,Answer()
      same => n,Background(custom/voice-menu)
      same => n,WaitExten(5)

   exten => 1,1,Playback(digits/1)
      same => n,Playback(followme/pls-hold-while-try)
      same => n,Wait(2)
      same => n,Dial(SIP/SIP-1001)

   exten => 2,1,Playback(digits/2)
      same => n,Playback(followme/pls-hold-while-try)
      same => n,Wait(2)
      same => n,Dial(SIP/SIP-1002)

   exten => 9,1,Playback(custom/goodbye)
        same => n,Wait(2)
        same => n,Hangup()

   exten => i,1,Playback(custom/invalid-option)
        same => n,Wait(1)
        same => n,Goto(voice-menu,start,1)

   exten => t,1,Goto(voice-menu,start,1)
```

Esempio: interazione con database (CTI)

Scenario

- implementazione di un servizio CTI per la comunicazione dell'autolettura

Obiettivo

- l'utente chiama il servizio e digita il proprio **codice cliente** e l'**autolettura**
- viene chiesta conferma all'utente dei dati inseriti
- viene verificata l'esistenza del codice cliente nel database CRM
- se il cliente esiste, l'autolettura viene registrata nel database CRM

Nota

- In un contesto reale, sarebbe probabilmente necessaria qualche forma di autenticazione dell'utente (PIN), che in questo esempio viene tralasciata

Esempio: interazione con database (CTI)

func_odbc.conf

```
[USER_EXISTS]
dsn=asterisk-crm
readsql  =  SELECT IF(COUNT(1)>0, 1, 0)
     FROM users
     WHERE id = '${SQL_ESC(${ARG1})}'

[REGISTER _VALUE]
writesql  =  UPDATE users
     SET counter = '${SQL_ESC(${VAL1})}'
     WHERE id = '${SQL_ESC(${ARG1})}'
```

Esempio: interazione con database (CTI)

extensions.conf

```
exten => start,1,Answer()
same => n,Playback(welcome&prompt-user)
same => n,Read(USERID)
same => n,Playback(prompt-value)
same => n,Read(USERVALUE)
    ... riascolta il valore immesso e premi 1 per confermare
exten => 1, 1, GotoIf($[${ODBC_USER_EXISTS(${USERID})}] ? register : unknown)
same => n(register), Set(ODBC_REGISTER_VALUE(${USERID})=${USERVALUE})
same => n, GotoIf($[${ODBCROWS} > 0] ? ok : ko)
    ... conferma esito della registrazione
```

Esempio: pulsante di chiamata da CRM

Scenario

- implementazione del **pulsante ”chiama”** nel software CRM

Obiettivo

- l'ultilizzatore del software CRM clicca sul pulsante ”chiama” accanto al nome
- il software CRM invia comandi attraverso la Asterisk Manager Interface (AMI)
- il telefono sulla scrivania dell'utente squilla
- quando l'utente alza il ricevitore, viene automaticamente composto il numero di telefono presente nel CRM

Asterisk Manager Interface (AMI)

```
$ telnet localhost 5038
...
Action: Login
Username: admin
Secret: 1234
<CR>
Response: Success
Message: Authentication accepted
Action: Ping
<CR>
Response: Success
Ping: Pong
```

Connessione attraverso TCP socket sulla porta 5038 (default)

Richiesta di autenticazione

Invio del comando ”ping”

Esempio: organizzazione del dialplan

Scenario

- gestione di logiche complesse nel dialplan

Obiettivo

- implementare funzionalità che richiedono molto codice, o che necessitano dell'utilizzo di un vero e proprio linguaggio di programmazione
- rendere robusto e mantenibile il codice del dialplan
- riutilizzare codice già scritto per le funzionalità da implementare
- superare il limiti del linguaggio di "scripting" di Asterisk

Asterisk Gateway Interface (AGI)

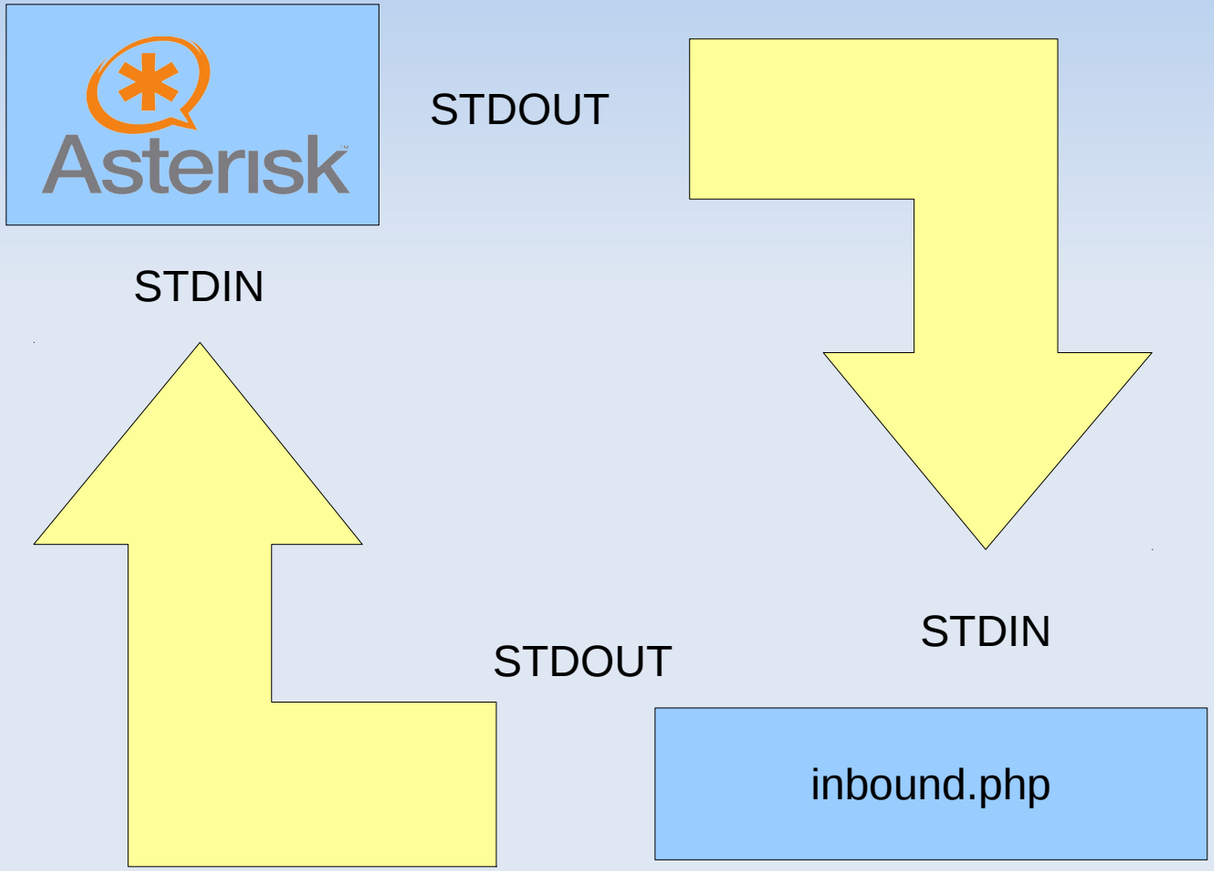
Dialplan Asterisk

```
[phpagi]
exten => 800,1,AGI(inbound.php)
```

inbound.php

```
#!/usr/bin/php -q
<?php
    require('phpagi.php');
    $agi = new AGI();
    $agi->answer();
    $agi->stream_file("hello-world");
    $agi->hangup();
?>
```

Asterisk Gateway Interface (AGI)



- comunicazione bidirezionale con lo script esterno
- indipendente dal linguaggio di programmazione utilizzato
- comunicazione attraverso STDIN / STDOUT, implementabile con qualsiasi linguaggio di programmazione

Esempio: scalabilità

Scenario

- ambiente domestico o micro azienda
- budget molto limitato
- impossibilità a mantenere attivo 24/7 un server fisico o virtuale

Asterisk... in una tasca

- Scheda Raspberry PI
- Scheda SD
- Alimentatore USB

... circa 50 euro!

- Ingombro e consumo minimi
- fino a 8-10 conversazioni simultanee
- ideale per sperimentare o per utenze domestiche

Linux Day – 25 ottobre 2014

fabianosoftware.it

@beppefabiano