

Présentation du PBX Asterisk

Par Mickaël Cissé

Asterisk

L'auteur – Mickaël Cissé

- **Directeur Technique en systèmes domotiques à Athemium**
- **Ingénieur R&D logiciel en VoIP à NeoTIP/Comverse**
- **Expérience de plus d'un an sur Asterisk**
- **Conception d'un serveur de VoIP basé sur Asterisk**

Plan

- **Introduction**
- **Installation**
- **Architecture**
- **Terminaux**
- **Codecs**
- **Protocoles**
- **Configuration**
- **Applications**
- **Dialplan**
- **Administration**
- **Programmation**
- **Plus loin**
- **Points forts/faibles**
- **Conclusion**

Introduction – Qu'est-ce qu'Asterisk?

- **Un Private Branch eXchange (PBX) Open Source**
- **Créé et géré par Digium**
 - **Serveur de sources, gestion des bugs, mailing lists, gestion des contributions, copyright**
 - **Originellement écrit en C par Mark Spencer**
- **Sous licence GPL.**
 - **Mais Digium se réserve le droit de re-licenser (et vendre) les sources**
 - **Tous les contributeurs doivent abandonner tous droits sur leurs soumissions**
- **Développé par une large communauté**
 - **En plus des développements de Digium, il y a des dizaines de contributeurs importants gérant chacun quelques fonctionnalités.**

Introduction – Qu'est-ce qu'Asterisk?

- **Fonctionne principalement sous Linux**
 - **Supporte quelques autres systèmes de type Unix et BSD**
 - **Support basique pour plateformes Win32 (pas de support matériel)**
- **Fonctionne avec quasiment n'importe quel matériel**
 - **Un Pentium II à 400MHz suffit pour un usage basique**
- **Supporte des cartes matérielles permettant de gérer des interfaces ISDN/RTC, etc.**
- **Multi-protocoles**
 - **VoIP : H323, SIP, MGCP, SCCP, IAX2**
 - **RTC : ZAP (FXO/FXS), CAPI (RNIS)**
- **Multi-codecs**
 - **G711, GSM, Speex, iLBC, G726, LPC10, G729, G723, ...**

Introduction – Qu'est-ce qu'Asterisk?

- **Un pont de conférence**
- **Un serveur vocal de menus interactifs (IVR)**
- **Une boîte vocale audio (voicemail)**
- **Un système de distribution automatique des appels**
- **Un système complètement personnalisable**
 - **Programmation d'un dialplan**
 - **Scriptable**

Introduction – Qu'est-ce qu'Asterisk n'est pas?

- **Un système de facturation**
 - Mais il serait souhaitable d'en avoir un extérieur
- **Un serveur web/XML/tftp**
 - Mais il sera intéressant d'en avoir un pour les téléphones qui en ont besoin (pour leur configuration)
- **Un système de reconnaissance vocale**
 - Mais on peut toujours en utiliser un externe
- **Un système de CRM (Customer Relationship Management)**
 - Mais il est envisageable d'en utiliser un extérieur

Introduction – Ambitions d'Asterisk?

- **Migrer toutes les fonctionnalités majeures des PBX propriétaires et matériels vers du logiciel open source**
- **Etre aussi générique que possible et ne pas dépendre d'un constructeur.**
- **Avoir le plus de fonctionnalités possibles**
- **Etre flexible et permettre d'ajouter facilement de nouvelles fonctionnalités**
- **Mais aussi ... Permettre à Digium de vendre ses cartes TDM ;-)**
- **Asterisk est une petite révolution qui vient empiéter sur les plates bandes des plus grands constructeurs de PABX**
- **Les constructeurs de téléphones en tiennent compte**

Installation – Généralités

- **Installation (relativement) facile et standard**
 - A partir de packages (conseillé)
 - A partir de la dernière version stable
 - A partir du SVN (nouvelles fonctionnalités)
- **Installation de modules annexes**
 - Accès base de données (db_odbc)
 - Gestion H323 (chan_oh323)
 - Gestion CAPI (chan_capi)
 - Voix françaises
 - ...
- **Aucune dépendance matérielle**
 - Fonctionne sur tous systèmes Linux (de préférence x86)

Installation – Répertoires

- **Répertoires utilisés (voir asterisk.conf)**
 - **/etc/asterisk**
 - **Fichiers de configuration**
 - **/usr/lib/asterisk/modules**
 - **Modules chargeables dynamiquement (codecs, channels, ...)**
 - **/var/lib/asterisk**
 - **Base de données (astdb)**
 - **/var/lib/asterisk/agi-bin**
 - **Scripts AGI**
 - **/var/lib/asterisk/mohmp3**
 - **Musiques d'attente**
 - **/var/lib/asterisk/sounds**
 - **Les voix d'Asterisk**

Installation – Répertoires

- Répertoires utilisés
 - `/var/spool/asterisk/voicemail`
 - Messages vocaux
 - `/var/spool/asterisk/outgoing`
 - Fichiers de génération d'appels
 - `/var/log/asterisk`
 - Logs
 - `/var/log/asterisk/cdr-csv`
 - Fichiers CDR

Installation – Démarrage

- Options de démarrage:
 - -c Permet d'obtenir le CLI
 - -v Augmente la verbosité
 - -r Connexion à un Asterisk en cours d'exécution

- Lancement:
 - asterisk -cvvv

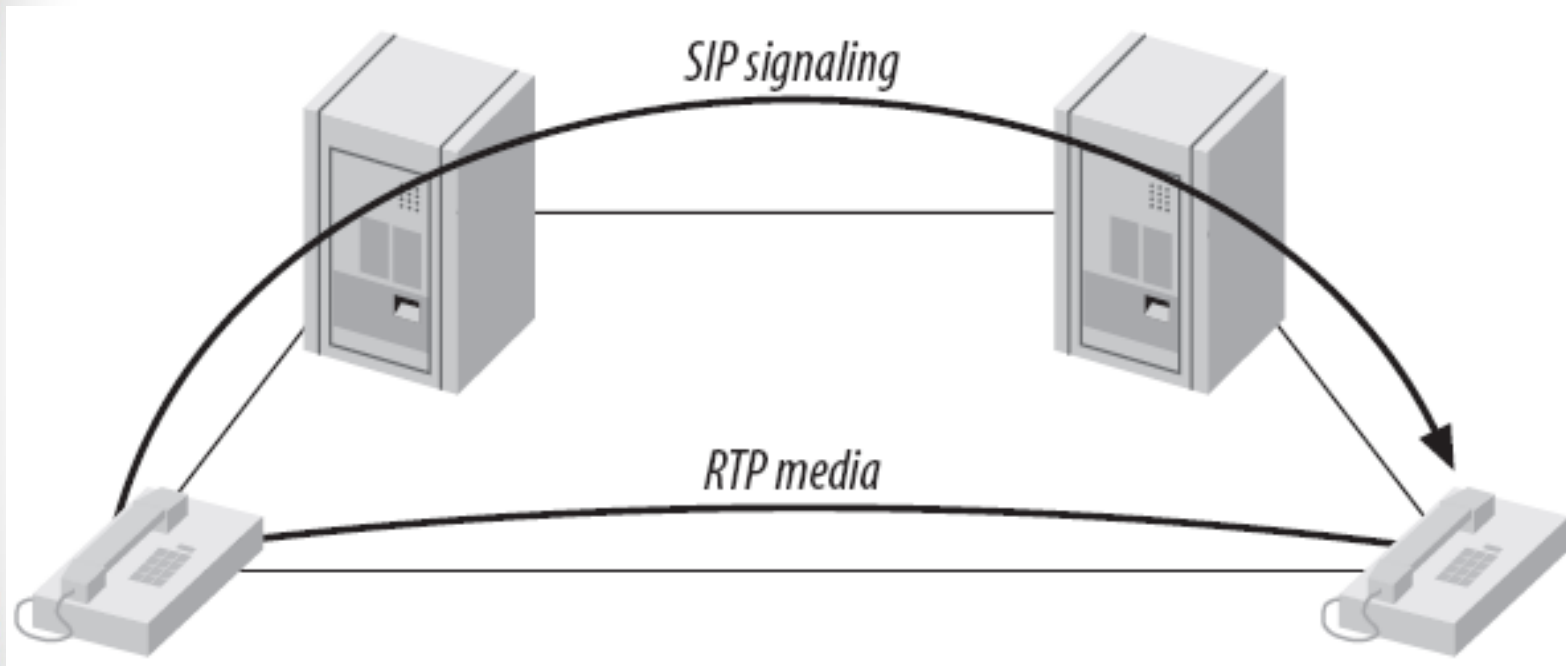
- Connexion d'un autre terminal:
 - asterisk -rcvvv

Installation – Le réseau

- **Bande passante**
 - **Peu importe ; suivant le nombre d'appels/codecs**
- **Lattence**
 - **150 ms maximum pour un bon confort**
- **Perte de paquets**
 - **Aussi faible que possible! (<1%)**
- **Gigue (jitter)**
 - **Aussi faible que possible**
- **Sur un réseau local SWITCHE, pas de problèmes**
- **Sur internet, ça se passe bien en pratique**
 - **Attention au choix du type de connexion et au fournisseur d'accès ;-)**

Architecture – Signalisation/flux 1

- Chemins possiblement différents pour la signalisation et les flux audio

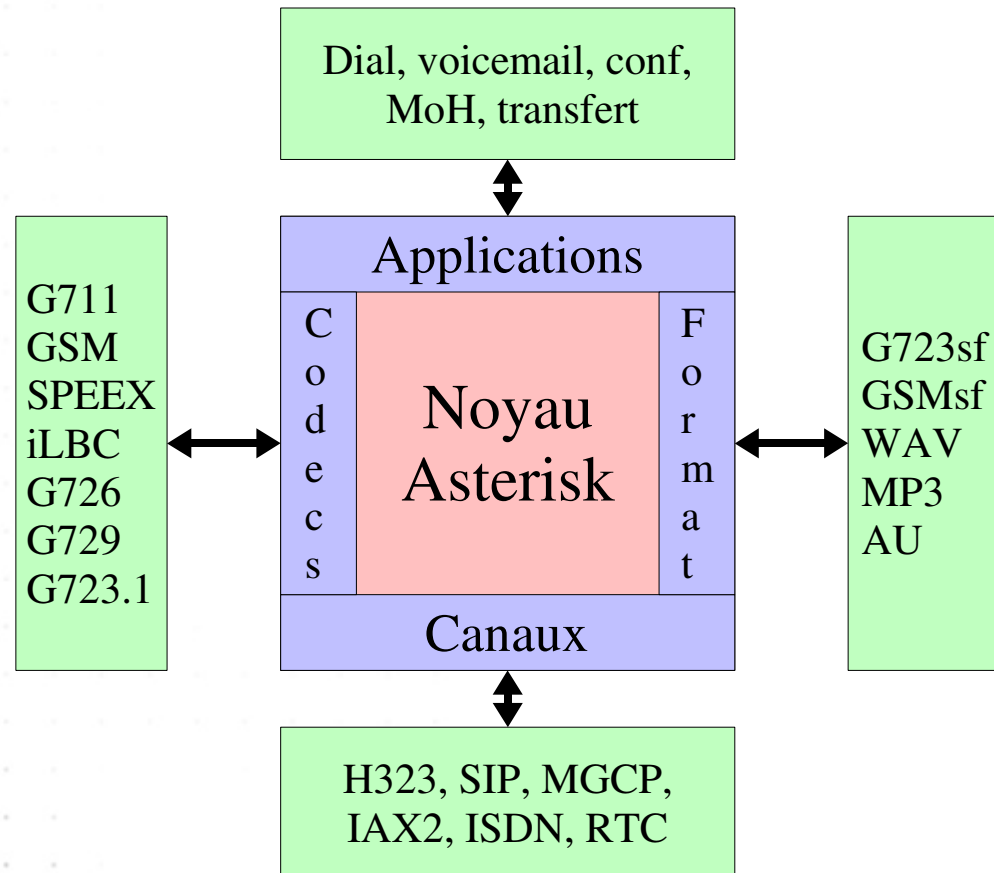


Architecture – Signalisation/flux 2

- **Quand Asterisk garde t-il la maîtrise des flux?**
 - **Quand un des deux terminaux ne supporte pas**
 - **S'il n'y a pas de codec commun (transcodage)**
 - **Si l'un des clients est derrière un routeur avec NAT**
 - **S'il a besoin d'écouter les DTMF (Dual Tone Multi-Frequency). Pour les menus IVR et les transferts par exemple.**
- **Quels protocoles supportent cette différenciation?**
 - **SIP**
 - **MGCP**
- **En IAX2, seul un transfert complet (SIG/RTP) est possible**

Architecture – 1

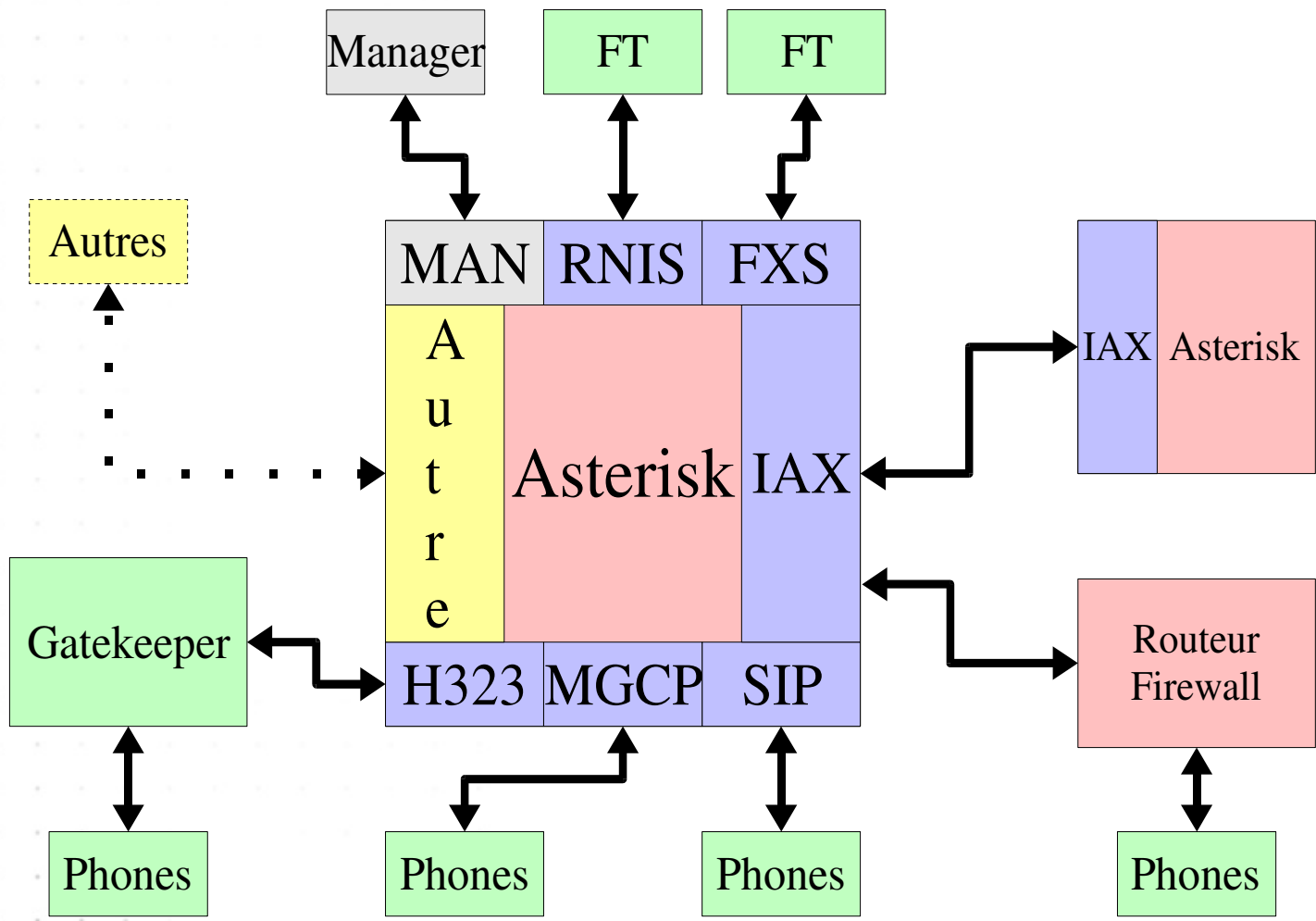
- Architecture Modulaire – Un noyau, des plugins (>100!)



Asterisk

Architecture - 2

- Architecture générale



Terminaux – Hardphones

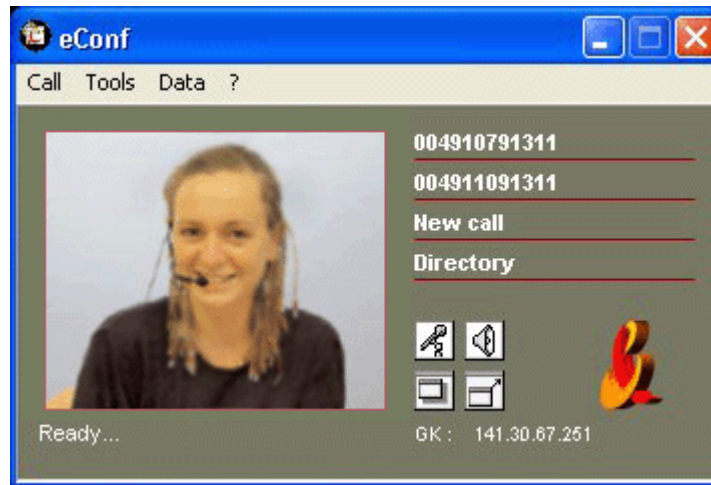
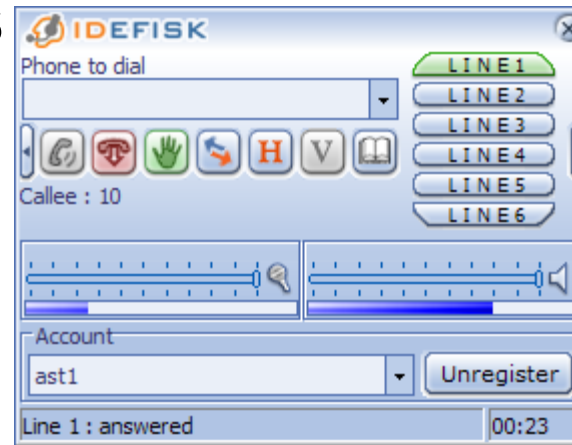
- Terminaux communs
 - Swissvoice IP10S (SIP/H323/MGCP)
 - Grandstream BudgetTone (SIP)
 - Grandstream GXP2000 (SIP)
 - Polycom SoundPoint IP500 (SIP)
 - Cisco 79XX (SIP/H323)
 - Grandstream GXV3000 (SIP)



Terminaux – Sofphones

• Softphones

- IdeFisk
- Xten
- Polycom PVX
- Econf



Asterisk

Terminaux – le Grandstream GXP2000



- **Grandstream GXP-2000**
 - SIP
 - Grand écran
 - Switch ethernet
 - Mise en attente
 - Transfert aveugle/avec consultation
 - Conférence à 3
 - Alimentation possible par ethernet
 - 4 lignes simultanées
 - Touche de numérotation rapide
 - Etat des postes (compatible Asterisk)
 - Extensible

Terminaux – le Grandstream GXP2000

- Extensible!





Adjustable LCD screen

Message Waiting Indicator

Line 1-4 Keys

Menu Keys

Mute/Delete

Message

Conference

Transfer

Speed Dial / Configurable line indicators

RJ11

Hold

Speaker

Send/Re-Dial

Standard Keypad

GRANDSTREAM

Terminaux – Gateways

- Pour chacun des protocoles, sont supportés les terminaux (téléphones), mais aussi les gateways (passerelles) vers d'autres protocoles.
 - Permet de se connecter à des lignes France Telecom en utilisant une gateway H323/T0 par exemple.



Terminaux – Configuration

- Configuration locale du réseau (DHCP/statique)
- Protocole
- Adresse IP/port du serveur
- Authentification (login/passwd)
- Codecs

Codecs – Introduction

- Permet de COder/DECoder et COmpresser/DECcompresser un signal audio (ou autre) analogique en une forme numérique
- Tient généralement compte de l'effet psycho acoustique. Le but n'est pas d'avoir un signal identique à l'original, mais que le cerveau ne distingue pas la différence
- Le but est souvent de compresser la quantité de données à transporter tout en gardant une bonne qualité et sans consommer trop de ressources processeur.

Codecs	Bande passante (kbps)
G.711	64
G.726	16/24/32
G723.1	5.3/6.3

Codecs	Bande passante (kbps)
G.729A	8
GSM	13
Speex	2.15 à 22.4

Codecs – Détails G.711 et G.726

- **G.711**
 - 8bits/8KHz (64000 bits par seconde)
 - C'est le codec de base
 - Considéré non compressé même s'il utilise une échelle non linéaire
 - Deux versions: μ -law (Amérique du Nord) et A-law (reste du monde)
 - Consomme très peu de ressources CPU

- **G.726**
 - Un des plus anciens codec
 - Type ADPCM (Adaptive Differential Pulse-Code Modulation)
 - Qualité quasi identique à G.711
 - Utilise la moitié de la bande passante de G.711
 - Codage par différentiation d'un échantillon sur un autre
 - Ne supporte pas les signaux de FAX/Modem
 - Consomme peu de ressources CPU

Codecs – Détails G.723.1 et G.729A

- **G.723.1**
 - Optimisé pour la parole
 - Compression très forte
 - Très mauvaise qualité avec la musique
 - Encombré de brevets et très couteux
 - Consomme beaucoup de ressources CPU

- **G.729A**
 - Très bonne qualité en comparaison de la bande passante requise!
 - Encombré de brevets, mais très abordable
 - Très populaire
 - Consomme beaucoup de ressources CPU

Codecs – Détails GSM et Speex

- **GSM**
 - LE codec de choix pour Asterisk
 - Bonne compression
 - Pas de soucis de brevets
 - Très bonne performance par rapport aux ressources consommées
 - Consomme assez peu de ressources CPU

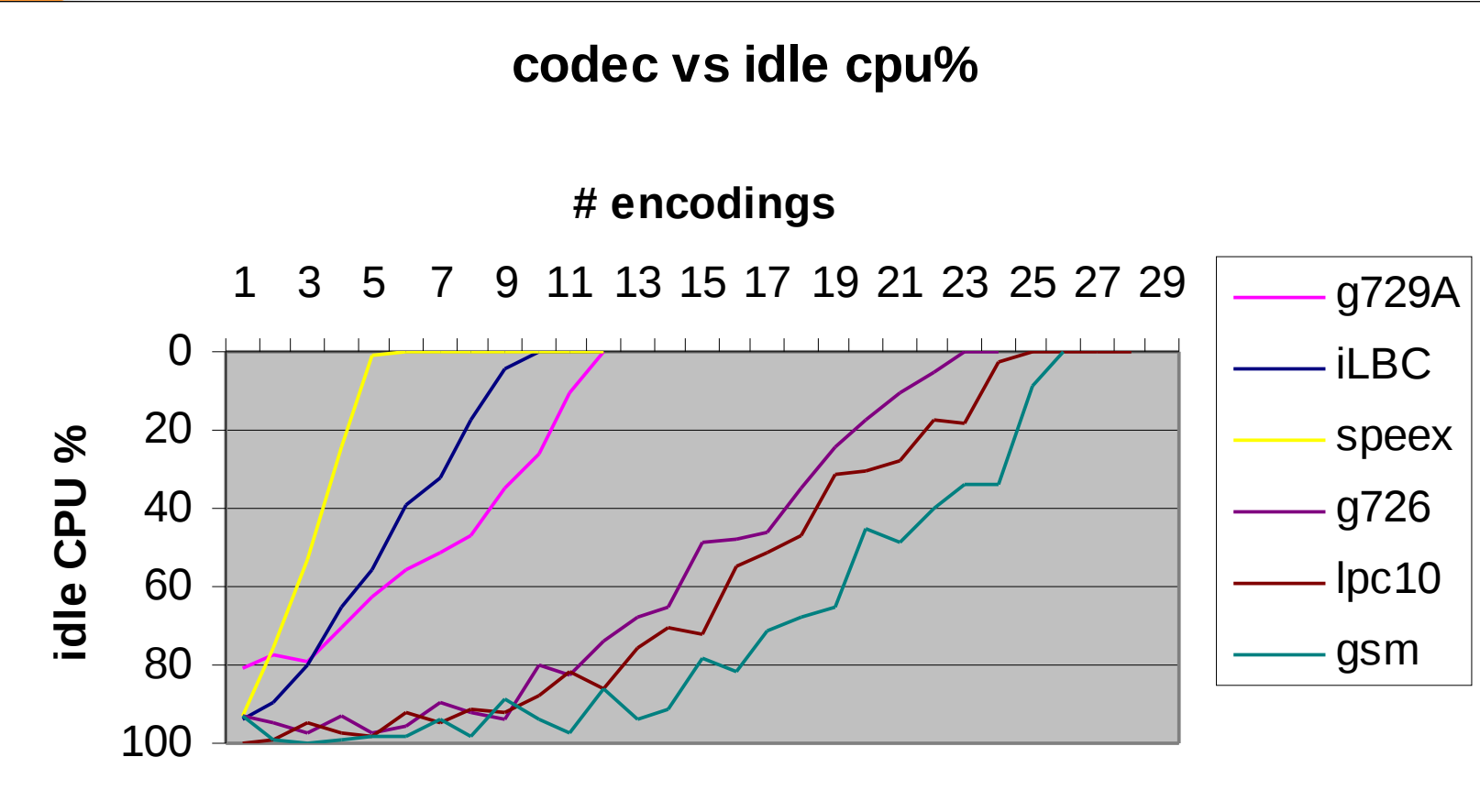
- **Speex**
 - Totalement libre
 - Support du changement dynamique de bitrate (VBR)
 - Support de bande large et étroite (16KHz par ex)
 - Consomme énormément de ressources CPU

- **Que choisir?**
 - G.711 et GSM

Codecs – Gestion par Asterisk

- **Asterisk peut laisser passer tous les codecs**
 - **Mais cela limite les services vocaux (voicemail, transferts)**
- **Conversion de n'importe quel codec vers n'importe quel autre codec**
 - **Garanti une très bonne interopérabilité**
 - **Très gourmand en ressources**
 - **A réserver à de rares cas ; comme sur Internet**
- **Asterisk sait utiliser les fichiers vocaux (pour la voicemail) pré-enregistrés dans le codec du canal**
 - **Permet de limiter la charge CPU**

Codecs – Comparatif



Protocoles – Comparatif - 1

• SIP

- LE protocole principal d'Asterisk (réseau local)
- Assez récent
- Assez simple
- NAT possible

• IAX2

- LE protocole d'Asterisk
- Récent
- Simple
- NAT très facile

• H.323

- Assez bien supporté par Asterisk
- L'un des plus anciens
- Très complexe (RAS/Q931/H245/RTP)
- NAT quasi impossible

Protocoles – Comparatif - 2

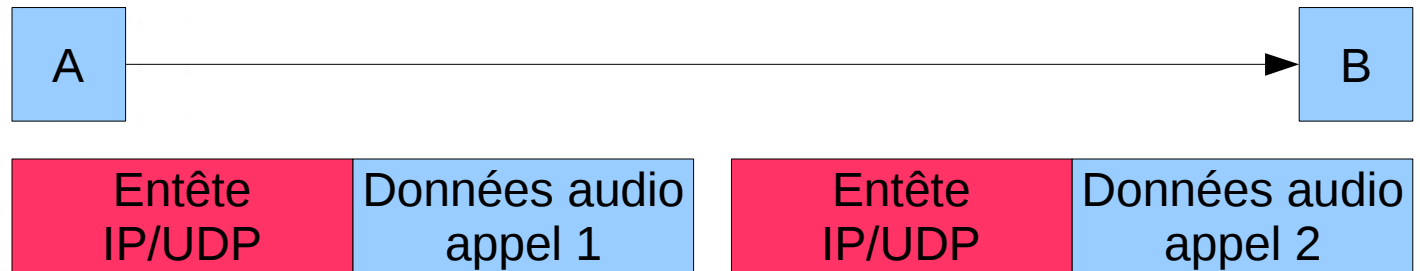
- **MGCP**
 - Bien supporté par Asterisk
 - Assez récent
 - Assez simple
 - NAT possible
 - Rarement utilisé
- **Autres (Skinny, ...)**
 - Peu supporté par Asterisk
- **Que choisir?**
 - SIP et IAX2

Protocoles – IAX2

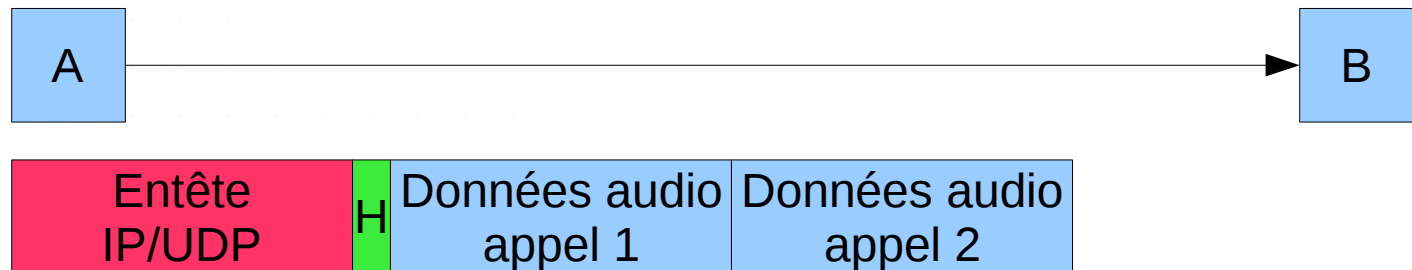
- **Inter Asterisk eXchange**
- **Protocole binaire très simple**
- **Résultat d'un travail communautaire**
 - Il y a maintenant un draft IETF
- **Utilise seulement un port UDP (4569)**
 - Très pratique pour le NAT
- **Très efficace au niveau de la taille des paquets**
- **Support du transfert**
- **Mode "tronc" (trunk) entre IPBX**
 - Efficace

Protocoles – IAX2 en mode tronc 1

- Prenons le cas de deux appels simultanés de A vers B
- Cas standard (SIP/IAX2):



- Cas IAX2 trunked



Protocoles – IAX2 en mode tronc 2

- **Gain de bande passante très intéressant entre deux IPBX**
- **Exemple pratique avec G.729A**
 - **1 appel -> 100 paquets/s et 30 kbps**
 - **2 appels -> 100 paquets/s et 39.7 kbps**
- **Economie de 20.3kbps sur le deuxième appel!**
- **L'overhead moyen est d'autant plus faible qu'il y a un grand nombre d'appels (approximativement)**

Configuration – Généralités

- **Nombreux fichiers de configuration (~50)**
 - **Chaque plugin peut en avoir un.**
 - **Organisés en sections**

```
; Commentaire  
[section1]  
keyword=value
```

```
[section2]  
keyword=value
```

- **Exemples:**
 - **asterisk.conf**
 - **modules.conf**
 - **sip.conf**
 - **iax.conf**
 - **meetme.conf**
 - **voicemail.conf**
 - **musiconhold.conf**

Configuration – asterisk.conf

[directories]

astetcdir => /etc/asterisk

astmoddir => /usr/lib/asterisk/modules

astvarlibdir => /var/lib/asterisk

astagidir => /var/lib/asterisk/agi-bin

astspooldir => /var/spool/asterisk

astrundir => /var/run

astlogdir => /var/log/asterisk

[options]

verbose=<value> ; niveau de verbosité initial (-v)

debug=yes|no|<val/> ; niveau de debug (-d)

nofork=yes|no ; ne pas lancer en arrière plan (-f)

console=yes|no ; activer la console (-c)

highpriority=yes|no ; utiliser une haute priorité (-p)

dumpcore=yes|no ; générer un fichier core en cas de plante (-g)

quiet=yes|no ; exécution silencieuse (-q)

Configuration – modules.conf

```
[modules]
autoload=no           ; Mettre « yes » pour laisser asterisk charger
tous                  ; les modules

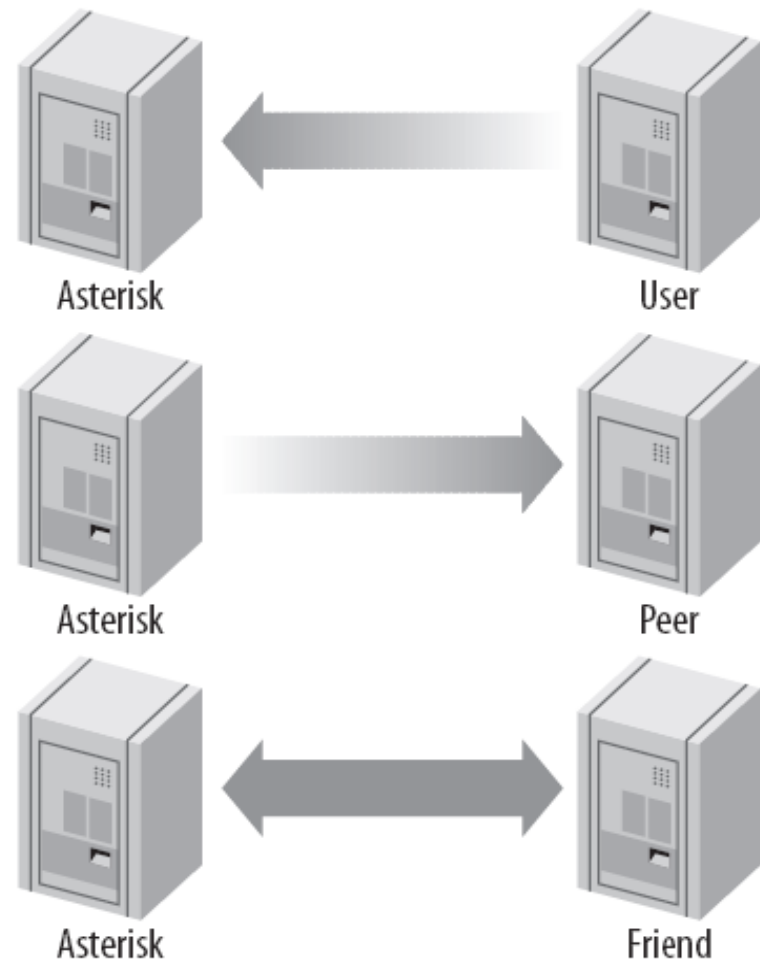
load => res_adsi.so
load => pbx_config.so
load => chan_iax2.so   ; Requier : res_crypto.so, res_features.so
load => chan_sip.so    ; Requier : res_features.so
load => codec_alaw.so
load => codec_gsm.so
load => codec_ulaw.so
load => format_gsm.so
load => app_dial.so    ; Requier : res_features.so, res_musiconhold.so
load => res_features.so

noload => chan_oss.so
```

Configuration – sip.conf

```
[general]
port=5060
bindaddr=192.168.0.1
context=sip
allow=gsm
```

```
[phone1]
type=friend
username=8307
secret=pwd
callerid= « Mickaël Cissé » <8307>
host=dynamic
nat=no
allow=ulaw
```



Configuration – iax.conf

```
[general]
bindport = 4569
bindaddr = 0.0.0.0
disallow=all
allow=ulaw
allow=gsm
```

```
[trunk1]
type=friend
username=trunk1
secret=password123
auth=plaintext
host=192.168.1.1
context=fromiax
qualify=yes
trunk=yes
```

Création d'un tronç IAX:
Sur 192.168.1.2

Sur 192.168.1.1

```
[trunk1]
type=friend
username=trunk1
secret=password123
auth=plaintext
host=192.168.1.2
context=fromiax
qualify=yes
trunk=yes
```


Configuration – Voicemail

- Protégée par mot de passe
- Message de bienvenue par défaut ou personnalisable
- Différenciation occupé/indisponible
- Multiples dossiers pour ranger les mails vocaux
- Notification par email (message en pièce jointe)
- Message Waiting Indication (MWI)
- Transfert des messages
- Indication visuelle de nouveaux messages sur le téléphone

Configuration – voicemail.conf

[general]

; Send voice file attachments in email notifications

attach=yes

; Use wav format for all voicemail messages

format=wav

; Limit the maximum message length to 180 seconds

maxmessage=180

; Limit the minimum message length to 3 seconds

minmessage=3

; voicemail body

**emailbody=\n\tHi \${VM_NAME},\n\n\tYou have a \${VM_DUR} long new
voicemail message (number \${VM_MSGNUM}) in mailbox \$
{VM_MAILBOX}\nfrom \${VM_CIDNAME} (\${VM_CIDNUM}), on \$
{VM_DATE}\nso you might want to check it when you get a chance.\n\n**

[default]

**; Define user 123 with password 2048 named Joe User with email address
for voicemail notification, no pager, timezone set to san-diego, and allow
attachment of the voice mail message to the email notification.**

123=>2048, Joe User, juser@somewhere.net,,tz=san-diego|attach=yes

Applications – 1

Les applications sont des commandes pouvant être exécutées dans le dialplan.

- **Answer**

->Répondre à un appel (établissement du flux audio)

- **ODBCGet(varname=family/key)**

->Obtenir de la base de données la valeur d'une clé pour une famille donnée

- **Goto([[context]]extension][priority)**

->Sauter dans un nouveau contexte, extension, priorité

- **SayNumber(digits)**

->Prononcer un nombre dans la langue courante

- **Background(filename[&filename2...])**

->Jouer les fichiers indiqués en arrière plan

Applications – 2

- **Hangup**
 - Raccrocher l'appel
- **Voicemail(extension)**
 - Laisser des messages pour une extension
- **VoicemailMain(extension)**
 - Consulter les messages d'une extension
- **System(command)**
 - Exécuter une commande système
- **WaitExten(delay)**
 - Attendre une entrée DTMF pendant <delay>
- **WaitMusicOnHold(delay)**
 - Jouer une musique d'attente pendant <delay> seconds
- **AGI(commande|arguments)**
 - Exécuter un AGI (Asterisk Gateway Interface). Ceci permet de contrôler un canal, jouer de l'audio, lire des DTMF à partir de n'importe quel langage.

Dialplan – Introduction

- Configuré dans « extensions.conf »
- Organisé en contextes
- Chaque contexte contient une liste d'extensions
- Chaque extension contient une liste d'applications

Syntaxe d'une ligne d'extension:

```
exten => <exten>, <priorité>, <application>, [args]
```

Exemple (contexte demo, extension s) :

```
[demo]
```

```
exten => s,1,Answer
```

```
exten => s,2,Playback(hello-world)
```

```
exten => s,3,Hangup
```

Dialplan – Contextes

- Permettent de découper le dialplan en parties indépendantes

[sip]

```
exten => s,1,Answer
```

```
exten => s,2,Playback(you_are_sip)
```

```
exten => s,3,Hangup
```

[h323]

```
exten => s,1,Answer
```

```
exten => s,2,Playback(you_are_h323)
```

```
exten => s,3,Hangup
```

Dialplan – Extensions 1

- Permettent de donner les instructions à exécuter pour chaque contexte
- Syntaxe:
 - exten => <nom>, <priorité>, <application>([args, ...])
- Indiquent QUAND exécuter les instructions
 - L'extension correspondant au numéro appelé est utilisée
 - Par défaut, l'extension "s" (start) est utilisée

```
[sip]
```

```
exten => s,1,Answer
```

```
exten => s,2,Playback(you_are_sip)
```

```
exten => s,3,Hangup
```

Dialplan – Extensions 2

• Extensions possibles:

- N'importe quel numéro ou nom
- "s" (start) est la première utilisée si présente
- "i" (invalid) utilisée lorsqu'aucune autre entrée ne correspond
- "t" (timeout) utilisée en cas de ... timeout ;-)
- "h" (hangup) utilisé en cas de raccrochage
- Expressions "régulières":

- Commencent par « _ »
- X -> chiffre de 0 à 9
- Z -> chiffre de 1 à 9
- N -> chiffre de 2 à 9
- [1235-9] -> 1,2,3,5,6,7,8,9
- . (point) -> tout ce qui reste

Eviter d'utiliser "_." car cela peut vérifier TOUS les cas ; même "i"

```
[sip]
exten => 02,1,Playback(numero_02)
exten => _02.,1,Playback(prefix_02)
exten => _08XX,1,Playback(prefix_08xx)

exten => i,1,Playback(pbx-invalid)
exten => t,1,Playback(vm-goodbye)
```


Dialplan – Priorités

- Permettent de donner l'ordre des instructions à exécuter
- Commencent à 1
- Asterisk continue jusqu'à ce qu'une priorité manque

```
[sip]
exten => s,1,Answer
exten => s,2,Playback(you_are_sip)
exten => s,3,Hangup
```

- La priorité « n » (pour next) permet une numérotation automatique et l'assignation de labels

```
[sip]
exten => s,1,Answer
exten => s,n(playback1),Playback(you_are_sip)
exten => s,n,Hangup
```

Dialplan – Applications

- Instructions permettant réellement de tout faire
 - Décrocher
 - Jouer un message
 - Raccrocher
 - ...
- Nécessitent souvent des arguments
 - Mis entre parenthèses
 - Séparé par des virgules (',') ou des pipes ('|')

```
[sip]  
exten => s,1,Answer  
exten => s,2,Playback(you_are_sip)  
exten => s,3,Hangup
```

Dialplan – Menu IVR 1

- Menu Vocal Interactif

```
[ivr]
exten => s,1,Answer
exten => s,2,Background(menu)
exten => s,3,WaitExten(10)
exten => s,4,Goto(ivr,s,2)

exten => 1,1,Playback(digits/1)
exten => 1,2,Goto(ivr,s,2)

exten => #,1,Playback(vm-goodbye)
exten => #,2,Hangup

exten => i,1,Playback(pbx-invalid)
exten => i,2,Goto(ivr,s,2)

exten => t,1,Playback(vm-goodbye)
exten => t,2,Hangup
```

Dialplan – Menu IVR 2

;Exemple de menu IVR

[mainmenu]

; press 1 for sales, 2 for support or 0 for an operator

; 8001 is the support number

; 8900 is the operator number

exten => s,1,Background(welcome-mainmenu)

exten => s,2,WaitExten(10)

exten => s,3,Goto(s,1)

exten => 1,1,Goto(sales, s, 1)

exten => 2,1,Dial(SIP/8001)

exten => 0,1,Dial(SIP/8900)

[sales]

; press 1 for Aurélie, press 2 for Stéphanie

exten => s,1,Background(welcome-sales)

exten => s,2,WaitExten(10)

exten => s,3,Goto(s,1)

exten => 1,1,Dial(SIP/7001)

exten => 2,1,Dial(SIP/7002)

Dialplan – Variables de channel

- Spécifiques à un channel
- Définies par l'application "Set"

```
[sales]
; press 1 for Aurélie, press 2 for Stéphanie
exten => s,1,Set(Aurelie=SIP/7001)
exten => s,2,Set(Stephanie=SIP/7002)
exten => s,3,Background(welcome-sales)
exten => s,4,WaitExten(10)
exten => s,5,Goto(s,3)

exten => 1,1,Dial(${Aurelie})
exten => 2,1,Dial(${Stephanie})
```

Dialplan – Variables globales

- Définies dans le contexte spécial "[globals]"
- Définies par l'application "Set" et le mot clef "GLOBAL"

```
[globals]
```

```
Stephanie=SIP/7002
```

```
[sales]
```

```
; press 1 for Aurélie, press 2 for Stéphanie
```

```
exten => s,1,Set(GLOBAL(Aurelie)=SIP/7001)
```

```
exten => s,2,Background(welcome-sales)
```

```
exten => s,3,WaitExten(10)
```

```
exten => s,4,Goto(s,2)
```

```
exten => 1,1,Dial(${Aurelie})
```

```
exten => 2,1,Dial(${Stephanie})
```

Dialplan – Variables prédéfinies

- **`{CALLERID(num)}`** - numéro de l'appelant
- **`{CALLERID(name)}`** - nom de l'appelant
- **`{DNID}`** - numéro composé
- **`{UNIQUEID}`** - identifiant unique de l'appel
- **`{CONTEXT}`** - le contexte courant
- **`{EXTEN}`** - l'extension courante
- **`{PRIORITY}`** - la priorité courante
- **`{CHANNEL}`** - channel

Dialplan – Utilisation des variables chaîne

- Longueur d'une chaîne
 - `${LEN(chaine)}`
- Concaténation de chaînes
 - `${var1}${var2}`
- Extraction d'une sous-chaîne
 - `${var:offset:length}`
 - Si offset est négatif, part de la fin

```
[sip]
exten => s,1,Set(dest=8400)
exten => s,2,NoOp(${LEN(${dest})}) ; -> 4
exten => s,3,NoOp(1${dest}) ; -> 18400
exten => s,4,NoOp(${dest:1}) ; -> 400
exten => s,5,NoOp(${dest:-2}) ; -> 00
exten => s,5,NoOp(${dest:1:2}) ; -> 40
```


Dialplan – Utilisation des variables numériques

- Calculs
 - `[$expression numérique]`
- Opérateurs logiques
 - `|` `&` `!`
- Opérateurs de comparaison (entière si possible)
 - `=` `!=` `<` `>` `<=` `>=`
- Opérateurs arithmétiques
 - `+` `-` `*` `/` `%`

```
[sip]
exten => s,1,Set(dest=8400)
exten => s,2,NoOp([$${dest} + 1]) ; -> 8401
```

Dialplan – Application Macro

- **Macro(nom,arg1,arg2,...)**
- **Permet d'exécuter une macro en utilisant le contexte « macro-nom » en allant à l'extension « s »**
- **Les arguments sont accessibles dans `${ARG1}`,`${ARG2}`...**
- **Les variables `${MACRO_EXTEN}`, `${MACRO_CONTEXT}`, `${MACRO_PRIORITY}` contiennent l'origine de l'appel**
- **`${MACRO_OFFSET}` peut-être fixé pour influencer la priorité de retour**

```
[sip]
```

```
exten => s,1,Macro(settimeouts,5,10,3600)
```

```
exten => s,2,Dial(SIP/8400)
```

```
[macro-settimeouts]
```

```
exten => s,1,Set(TIMEOUT(digit)=${ARG1})
```

```
exten => s,2,Set(TIMEOUT(response)=${ARG2})
```

```
exten => s,3,Set(TIMEOUT(absolute)=${ARG3})
```

Dialplan – Application Dial

• **Dial(technologie/ressource[&technologie2/ressource2...]
[,timeout][,options])**

- Place un appel vers une ou plusieurs lignes. La première ligne qui répond est mise en relation. Les autres lignes sont raccrochées.
- L'état de l'appel est retourné dans la variable DIALSTATUS (peut être CHANUNAVAIL, CONGESTION, NOANSWER, BUSY, ANSWER, CANCEL, ...)
- Options:
 - t – Autoriser l'appelé à transférer l'appelant (#)
 - T – Autoriser l'appelant à transférer l'appelé (#)
 - w – Autoriser l'appelé à enregistrer l'appel
 - W – Autoriser l'appelant à enregistrer l'appel
 - m – Fournir la musique d'attente jusqu'au décrochage
 - L(x) – Limiter l'appel à x millisecondes
 - M(y) – Appeler la macro y lors de l'établissement de l'appel

Dialplan – Application Voicemail

- **VoiceMail(boxnumber[@context], flags)**

- Sauve l'audio du channel dans un fichier associé à la boîte vocale "boxnumber"
- Les flags peuvent être:
 - (rien) - jouer les instructions
 - s - ne rien jouer
 - su - jouer le message d'indisponibilité
 - u - message d'indisponibilité + instructions
 - sb - message d'occupation
 - b - message d'occupation + instructions

```
[sip]  
exten => s,1,VoiceMail(8400,b)
```

Dialplan – Application Gotolf

- **Gotolf(condition?label_true[:label_false])**
 - Aller à label_true si la condition est vraie
 - Aller à label_false sinon
 - Les labels sont de la forme [[context,]extension,]priority

```
[sip]
exten => s,1,Gotolf("${CALLERID(num)}" == "8400"?3:2)
exten => s,2,Hangup
exten => s,3,Dial(SIP/8401)
```

Dialplan – Fonction DB

- Pour récupérer/fixer/supprimer une valeur dans la base de données d'Asterisk
- **Set(varname=\${DB(family/key)})**
- **Set(DB(family/key)=value)**
- **DB_EXISTS(family/key)**
 - Test si la clé existe
- **Set(old_val=\${DB_DELETE(family/key)})**

```
[macro-setforward]
; ${ARG1} is the number to redirect to
exten => s,1,Set(DB(CF/${CALLERID(NUM)})=${ARG1})
```

```
[macro-call]
; ${ARG1} is the number to call
exten => s,1,GotoIf(${DB_EXISTS(CF/${ARG1})}?4)
exten => s,2,Dial(SIP/${ARG1}) ; CF not found!
exten => s,3,Hangup
```

```
exten => s,4,Dial(SIP/${DB(CF/${ARG1})})
exten => s,5,Hangup
```

Dialplan – Exemple simple

```
[sip]
exten => _8.,1,Macro(call,${EXTEN})

; extension invalide
exten => i,1,Hangup

[macro-call]
; ${ARG1} is the number to call
exten => s,1,Gotof(${DB_EXISTS(CF/${ARG1})}?50)
exten => s,2,Dial(SIP/${ARG1})
exten => s,3,Gotof("${DIALSTATUS}" = "BUSY"?100)
exten => s,4,Hangup

exten => s,50,Dial(SIP/${DB(CF/${ARG1})})
exten => s,51,Hangup

;busy
exten => s,100,Voicemail(${ARG1})
exten => s,101,Hangup
```

Dialplan – Conférences

- **MeetMe([numconf][, [[options],[pin]]**
 - Joindre une conférence
 - Options:
 - m – écouter uniquement
 - t – parler uniquement
 - i – annoncer les entrées/sorties
 - p – autoriser à quitter par #
 - d – ajouter automatiquement la conférence
 - M – Activer MoH pour un utilisateur seul

[conf]

exten => s,1,Answer

exten => s,2,meetme(1,dM)

exten => s,3,Hangup

Dialplan – Include

- Ce qu'il ne faut pas faire

```
[main]
exten => _X11,1,Dial(Zap/1/${EXTEN},500,r)
exten => _9.,1,Dial(SIP/${EXTEN}@mysipprovider,60,r)
exten => _011.,1,Dial(SIP/${EXTEN:3}@int-sip,60,r)
exten => h,1,Hangup
```

Dialplan – Include

- Ce qu'il faut faire

```
[main]
include => emergency
include => outside-line
include => international
exten => h,1,Hangup

[emergency]
exten => _X11,1,Dial(Zap/1/${EXTEN},500,r)

[outside-line]
exten => _9.,1,Dial(SIP/${EXTEN}@mysipprovider,60,r)

[international]
exten => _011.,1,Dial(SIP/${EXTEN:3}@int-sip,60,r)
```

Administration – Le CLI

- **Le CLI (Command Line Interface)**

- > Permet un accès interactif à Asterisk

- > Nombreuses commandes (>100) permettant de connaître l'état du système:

- "core show applications"

- Lister les applications

- "core show application dial"

- Donner la description de l'application « dial »

- "sip show peers"

- Lister les terminaux SIP

- "stop when convenient"

- Arrêter Asterisk quand il n'y a plus d'appels

- "core show functions"

- Lister les fonctions

- "manager show commands"

- Afficher les commandes du manager

Administration – Le CLI - exemples

- "sip show peers"

SIP endpoints:

Name/username	Host	Dyn Nat	ACL Mask	Port	Status
8409/8409	192.168.18.7	D	0.0.0.0	5060	Unmonitored
8500/8500	192.168.18.5	D	0.0.0.0	5060	Unmonitored

2 sip peers [2 online , 0 offline]

- "core show channels"

Current calls:

Channel	(Context Extension Pri)	State	Appl.	Data
SIP/8307-8587	(help_service_neoparker 2000 3)	Up	MusicOnHold	(Empty)

1 active channel(s)

Administration – Le CLI - debug

```
[default]
exten => 10,1,Answer
exten => 10,2,MusicOnHold(default)
exten => h,1,NoOp("call completed")
```

```
*CLI>
```

```
-- Accepting AUTHENTICATED call from 192.168.220.1, requested format = 2, actual format = 2
-- Executing Answer("IAX2/user1@user1/10", "") in new stack
-- Executing MusicOnHold("IAX2/user1@user1/10", "default") in new stack
-- Started music on hold, class 'default', on IAX2/user1@user1/10

-- Stopped music on hold on IAX2/user1@user1/10
== Spawn extension (default, 10, 2) exited non-zero on 'IAX2/user1@user1/10'
-- Executing NoOp("IAX2/user1@user1/10", "call completed") in new stack
-- Hungup 'IAX2/user1@user1/10'
```

```
*CLI>
```

Administration – Les interfaces graphiques

- Il y en a maintenant des dizaines:
 - <http://www.voip-info.org/wiki-Asterisk+GUI>
- La plupart sont des interfaces web
 - Interfaces administrateur
 - Interfaces utilisateur
- FreePBX semble assez complet

Administration – AsteriskNow GUI

The screenshot displays the AsteriskNow GUI for configuring a voice menu. The interface is divided into several sections:

- Header:** "digium | Asterisk" logo on the left, "About Digium | Help" and "Logout" on the right.
- Left Sidebar:** A vertical menu with options: Home, Users, Conferencing, Voicemail, Call Queues, Service Providers, Calling Rules, Networking, Voice Menus, Record a Menu, Active Channels, System Info, Backup, and Options. Below "Voice Menus" is a description: "Menus allow for more efficient routing of calls from incoming callers."
- Main Content Area:**
 - Page Title:** "Voice Menus Configuration"
 - Section:** "Voice Menu: VoiceMenu - Sales"
 - Form Fields:**
 - Name:** "Support"
 - Steps:** "Answer the Call" (with "Up" and "Down" buttons)
 - Add a new Step:** "-- Select --" (dropdown), "Add", "Delete" (buttons)
 - Dial other Extensions?:**
 - Keypress' Events:** A table with columns "Key" and "Action".
 - Keypress' Events Table:**

Key	Action
0	Goto Menu Sales
1	Goto Extension 6001
2	Play Invalid
3	Hangup
4	Custom [input field]
5	Disabled
 - Buttons:** "New", "Delete", "Save", "Cancel" (at the bottom)
 - Message:** "Creating new entry!"
- Right Panel:** "Tooltips: Keypress Events: Define the actions that occur when a user presses the corresponding digit."

- Footer:** "Copyright ©2006 Digium, Inc. Digium® and Asterisk® are registered trademarks of Digium, Inc. All Rights Reserved. Legal Information"


Programmation – Le manager d'appels

- Permet à un programme externe de connaître tout ce qui se passe sur Asterisk
- Permet d'envoyer des commandes à Asterisk
- Connexion TCP standard (un telnet suffit)
- Dialogue en mode texte
- Nombreux événements

```
Event: Newchannel  
Channel: SIP/101-3f3f  
State: Dialing  
Callerid: 101  
Uniqueid: 1094154427.10
```


Programmation – Le manager d'appels

- Exemple de transfert de deux postes en communication vers un autre numéro (une conférence, par exemple)

Action: Redirect
Channel: SIP/1234-6378
ExtraChannel: SIP/4321-45cf6c80
Exten: #1201
Context: default
Priority: 1

Response: Success
Message: Dual Redirect successful

Programmation – Le manager d'appels

- Flash Operator Panel

The screenshot displays the Asterisk Flash Operator Panel interface. At the top, there is a status bar with icons for lock, refresh, help, and a bug, followed by the text 'Info' and a search field. On the right side of the status bar, it says 'No timeout' with a dropdown arrow. The main interface is divided into several sections:

- Local Extensions:** A list of local extensions with status indicators (green or red) and phone icons. The entries are:
 - 11 Mariana
 - 12 Liliana
 - 13 Adan Monasterio
 - 14 Rodo
 - 16 Nicolás
 - 17 Martin
- Remote Extensions:** A list of remote extensions with status indicators and phone icons. The entries are:
 - Park 702
 - Nico
 - 19 Rodo (casa)
 - 18 Agustin
 - 20 Beatriz
- Conferences:** A section with a pink background containing two conference buttons:
 - Meetme 901
 - Meetme 902
- Queues:** A section with a green background containing one queue button:
 - Sales Queue
- External:** A section with a yellow background containing several external buttons:
 - Externa 1
 - Externa 2
 - Externa 3
 - Externa 4
 - GSM GW
 - Day Mode



Programmation – Asterisk Gateway Interface - 1

- **Équivalent des CGI**
- **Interface standard permettant à des programmes externes de prendre le contrôle du dialplan**
- **Permet des programmes plus élaborés que possible avec le dialplan**
- **Tous langages supportés (Perl, PHP, Python, C, ...)**
- **Communication par flux**
 - **STDIN : données d'Asterisk vers le programme**
 - **STDOUT: programme vers Asterisk**
 - **STDERR: messages de debug vers la console d'Asterisk**

Programmation – Asterisk Gateway Interface - 2

- Appel d'un script AGI

```
exten => 123,1,Answer( )  
exten => 123,2,AGI(agi-test.agi)
```

- Autres méthodes:

- EAGI() (Enhanced AGI)

- Permet au script de disposer du flux audio (flux 3)

- DeadAGI

- Permet l'exécution même sur un channel raccroché

- FastAGI

- Permet l'exécution de scripts AGI sur le réseau pour permettre à plusieurs serveurs Asterisk d'utiliser le même AGI.

Programmation – Asterisk Gateway Interface - 3

- **Lancement de agi-test.agi**
- **Connexion de stdin, stdout et stderr**
- **Envoi des données d'appel au script sur stdin**
 - **agi_request: agi-test.agi**
 - **agi_channel: SIP/8307-1224**
 - **agi_language: en**
 - **agi_callerid:**
 - **agi_context: default**
 - **agi_extension: 123**
 - **agi_priority: 2**
- **Terminaison par une ligne blanche**
- **Le script a alors le contrôle de l'appel**
 - **Envoi de commandes par STDOUT à Asterisk**
 - **Lecture des réponses sur STDIN**

Programmation – Asterisk Gateway Interface - 4

- **Commandes possibles**
 - ANSWER
 - DATABASE GET <family> <key>
 - GET DATA <file to be streamed> [timeout] [max digits]
 - GET VARIABLE <variablename>
 - HANGUP [<channelname>]
 - SAY NUMBER <number> <escape digits>
 - STREAM FILE <filename> <escape digits> [sample offset]
- **Résultats**
 - 200 Result=<number> <more data here>
 - 510 Invalid or unknown command

Programmation – Asterisk Gateway Interface - 5

- **Méthode basique**

- Utiliser "manuellement" les flux (read/write...)
 - `print "SAY NUMBER 123 \\\n";`

- **Utilisation de bibliothèques d'abstraction**

- Utiliser `Asterisk::AGI` en perl
 - `$AGI->exec('SayNumber',123);`

Programmation – Asterisk Gateway Interface - 6

- Exemple:

```
#!/usr/bin/perl
use Asterisk::AGI;

my $AGI = new Asterisk::AGI;
my %input = $AGI->ReadParse();

$AGI->exec('Playback','welcome');

# playing "menu.gsm" 10 sec 1 digit
my $digit = $AGI->get_data("menu", "10000", "1");

$AGI->stream_file('you-pressed-digit');
$AGI->exec('SayNumber',$digit);
$AGI->stream_file('vm-goodbye');

$AGI->hangup();
exit(0);
```


Plus loin – Multi-langues

- **Set(CHANNEL(language)=<language>)**
 - Permet de fixer la langue pour un appel dans le dialplan
 - Actif sur la plupart des applications (voicemail, playback, ...)

```
[conf]
exten => s,1,Set(CHANNEL(language)=fr)
exten => s,2,Answer
exten => s,3,Playback(transfer)
; Will try to play "sounds/fr/transfer.gsm"
; If failed, fallback to "sounds/transfer.gsm"
exten => s,4,Hangup
```

Plus loin – Auto dial out

- Pour générer « spontanément » des appels
 - Il suffit de créer un fichier texte spécifique dans `"/var/spool/asterisk/outgoing"`
- Permet par exemple de se faire appeler en retour

```
Channel: SIP/8400
MaxRetries: 2
RetryTime: 60
WaitTime: 30
Context: callme
Extension: 8401
Priority: 1
```

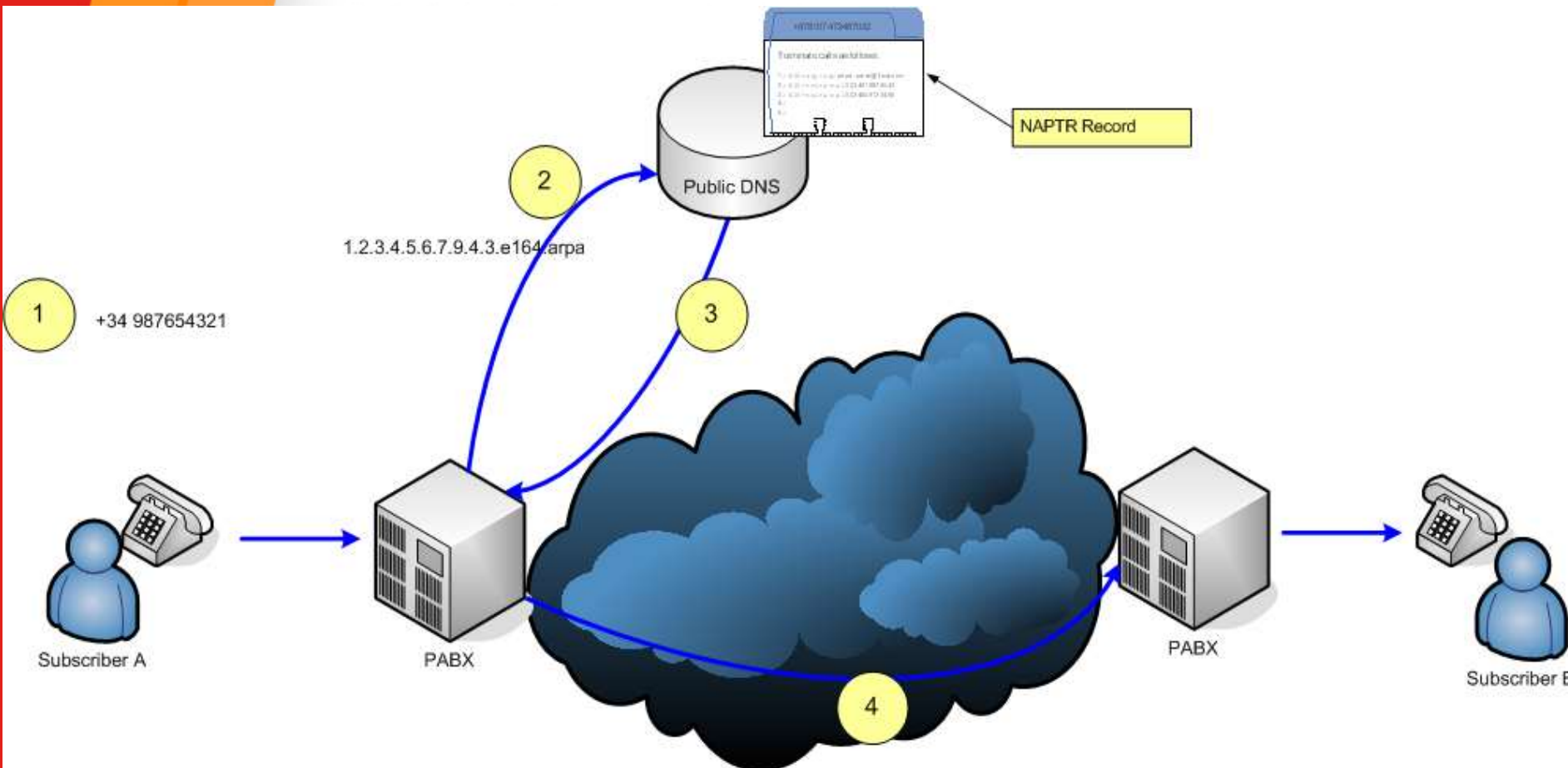
Plus loin – Fournisseurs VoIP

- **Asterisk peut-etre interconnecté avec des réseaux existant (SFR, Free, ...) ; surtout en utilisant SIP**

Plus loin – ENUM – 1

- **Electronic NUMbering (RFC3761) est très simple**
- **Basé sur la technologie DNS (domain e164.arpa)**
- **Exemple:**
 - **Appel vers le +34 987654321**
 - **Conversion en requete DNS**
1.2.3.4.5.6.7.8.9.4.3.e164.arpa
 - **Le DNS donne alors en réponse**
sip:34987654321@voipcarrier.net

Plus loin – ENUM – 2



Plus loin – QoS

- **Utiliser uniquement un réseau SWITCHÉ**
- **Dédier la machine pour Asterisk**
 - Pas de serveur X
 - Pas de base de données
 - Pas de serveurs web
 - Limiter les interruptions système
 - Utiliser du bon matériel/drivers (carte réseau)
 - Limiter l'utilisation du disque dur
 - Attention aux services de maintenance (cron, ...)
- **Activer la QoS partout où cela est possible**
 - Activer le marquage des paquets IP minimal ToS
 - Prioriser les flux marqués sur les switchs/routeurs/accès Internet
 - Utiliser les options de QoS des terminaux/gateways (buffer de gigue, marquage des paquets RTP)

Plus loin – Fiabilité logicielle

- **Assurer les fonctionnalités**
 - Dialplan simple/complet
- **Assurer les performances**
 - Priorité maximale
 - Nombre de sockets correct
- **Assurer la pérennité**
 - Démarrage automatique au boot
 - Initialisation du matériel au boot
 - Redémarrage du service en cas de plantage
 - Anticipation des problèmes (mémoire, disque)
 - Prévoir un redémarrage du service sans interruption de service
 - Maintenance aisée (mises à jour/restauration)
 - Sauvegarder la configuration lors des mises à jours

Plus loin – Fiabilité matérielle

- **Alimentation électrique**
 - La téléphonie classique est télé alimentée
 - Utiliser une alimentation de secours (PoE)
- **Disque dur**
 - Element qui lâchera un jour ou l'autre
 - Utiliser du RAID>0, SCSI, disques flash, liveCD
- **Système matériel**
 - Tout composant est susceptible de lâcher
 - Prévoir des composants de secours
- **Connectivité extérieure**
 - La connexion internet peut aussi lâcher
 - Prévoir modem/router de secours
 - Prévoir une connexion de secours
 - Prévoir une issue de secours (RTC/Internet)

Plus loin – Sécurité d'Asterisk

- **Sauts dans le dialplan**
 - Attention à bien contrôler ce que peuvent faire les utilisateurs (surtout extérieurs)
 - Attention aux boucles!
 - Les appels inconnus arrivent dans le context "default" par défaut
- **Fiabilité des données**
 - Le callerID peut ne pas être fiable!
- **Commandes dangereuses**
 - Eviter "System"!
- **Autres points**
 - Limiter le nombre d'appels simultanés
 - Limiter la durée des appels (Attention aux IVRs!)
 - Utiliser les options de sécurité de chaque channel

Plus loin – Sécurité système

- **Exécuter Asterisk**
 - Avec un utilisateur autre que root
 - Avec des permissions limitées
 - Dans un chroot
 - Derrière un firewall/NAT/shapping
- **Ne pas exécuter des programmes sans savoir ce qu'ils font!**
 - `safe_asterisk` ouvre une console sur la `tty9!`
 - Les interfaces accessibles des utilisateurs sont dangereuses!
- **Sécuriser le Linux**
 - GRSec
 - Remote Syslog

Points forts

- **Installation très simple**
- **Interopérabilité très bonne (virtuellement tous protocoles, tous codecs)**
- **Peut virtuellement tout faire avec les appels (conf, transfert, ...)**
- **Peut remplacer un PABX (avec cartes RTC/RNIS/GW)**
- **Utilise du matériel standard (remplacement aisé)**
- **Documentation très complète**
- **Communauté très active**
 - **Wiki, mailing listes**
 - **Plugins et autres addons (même des live CD)**
- **Possibilité de piloter Asterisk par connexion TCP (manager)**

Points faibles

- **Installation de modules optionnels parfois complexe**
- **Configuration complexe**
 - **Pas de GUI complet/standard**
 - **Plusieurs façons d'obtenir une même fonctionnalité (AGI, plugin, ...)**

Conclusion

- **Logiciel puissant et flexible**
- **Très connu et utilisé**
- **Logiciel libre**
- **Très évolutif**
- **Très Adaptable**

- <http://www.asterisk.org>
 - Site officiel d'Asterisk
- <http://www.voip-info.org/wiki-Asterisk>
 - La meilleure « documentation » sur Asterisk
- <http://www.digium.com/handbook-draft.pdf>
 - Un livre de Digium sur Asterisk
- <http://www.asteriskdocs.org>
 - Projet de documentation d'Asterisk

Questions