

Auto-Provisioning VOIP Telephones with Asterisk

Stephen Uhler

Sun Microsystems Laboratories

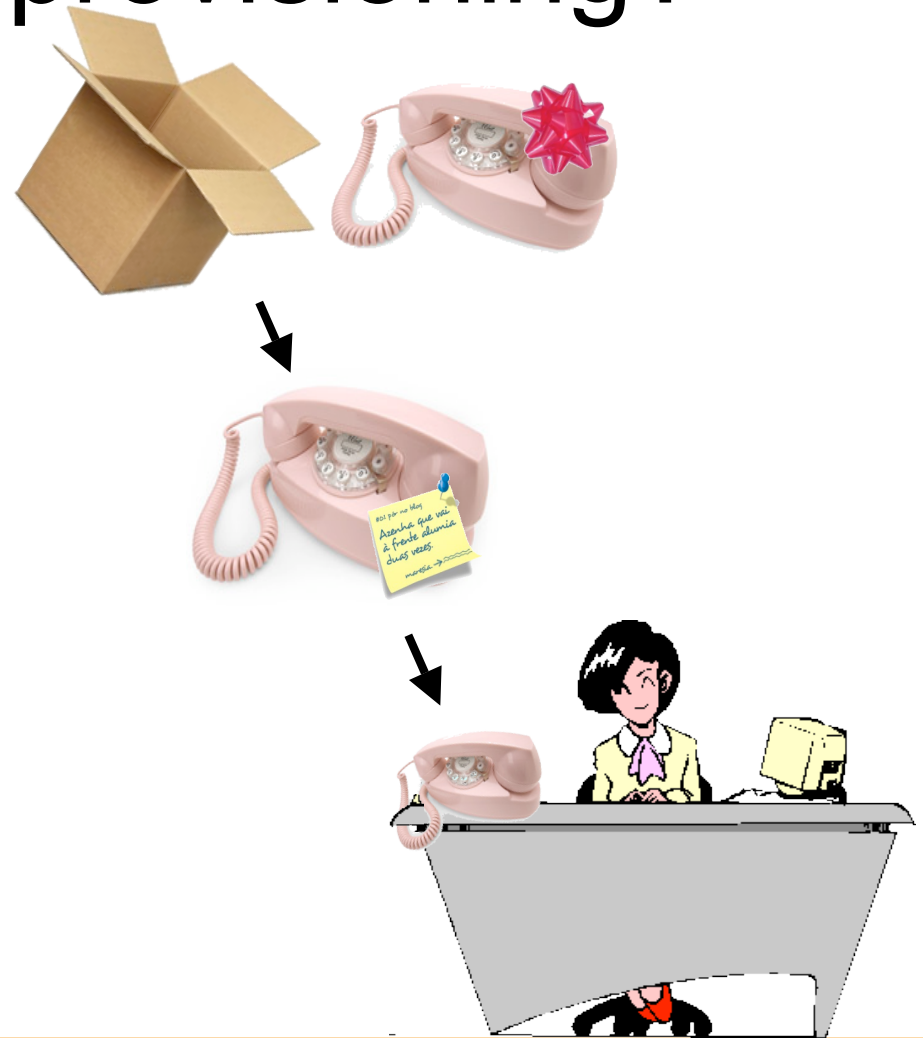
Outline

- What is auto-provisioning?
- Why do I need it?
- What phones do I use?
- Design choices
- Implementation



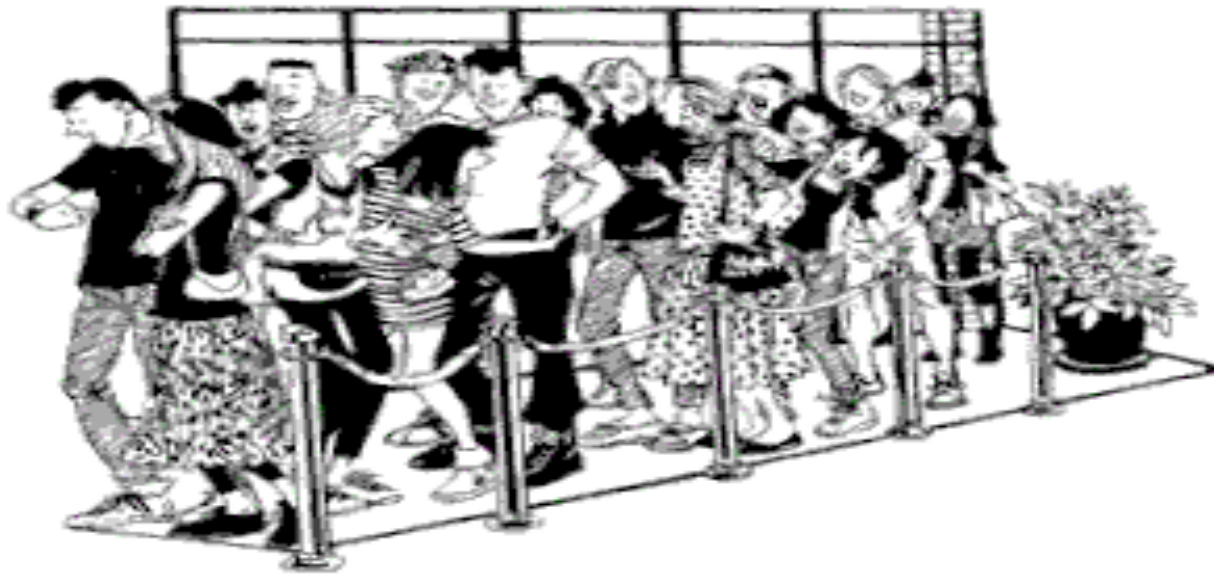
What is Auto-provisioning?

- 1) Take a new phone out of the box
- 2) Hand it to a new, relocated, or temporary employee, with simple instructions
- 3) Send them off



Why do I Need it?

- Dozens of new interns every quarter
- Ad-hoc phones for meetings and conferences
- Just me, and I already have a day job



What Phone to Choose?



What Phone to Choose?

- Every phone is configured differently
- We need to pick one model for simplicity (sanity)
- We have a nice* corporate LAN environment
- Phones need rational network configuration
- Polycom IP500/IP501
 - Acceptable quality
 - Too hard to configure manually
 - Possible to provision via the network

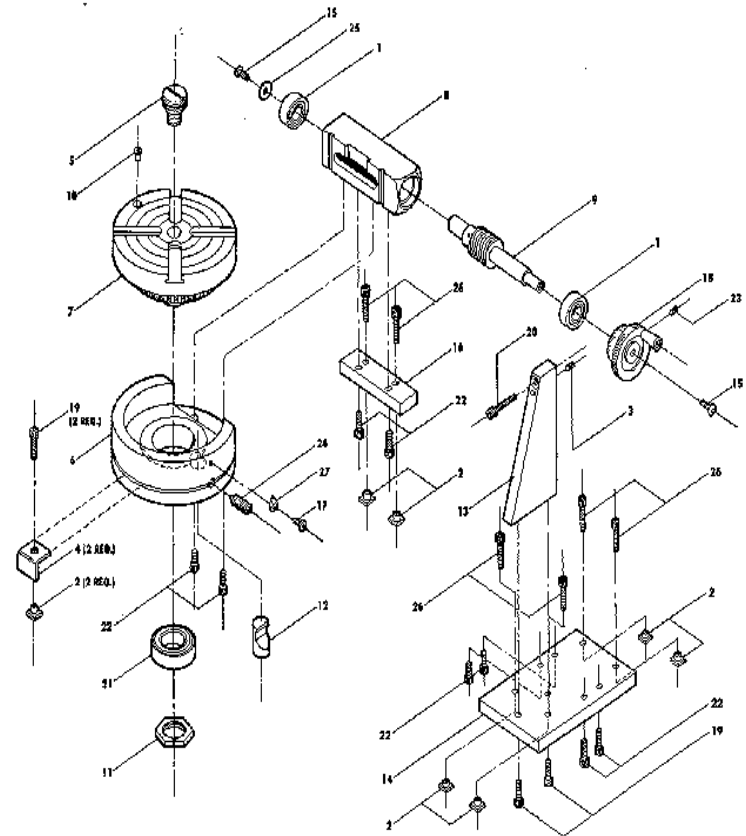
Design Choices

- Static -vs- realtime
- Extension assignment
- User authentication
- File layout/structure
- Naming the phones



The Pieces

- DHCP server
- FTP boot/configuration server
- Corporate LDAP authentication service
- Asterisk
 - IVR (extensions.conf)
 - Provisioning (AGI)

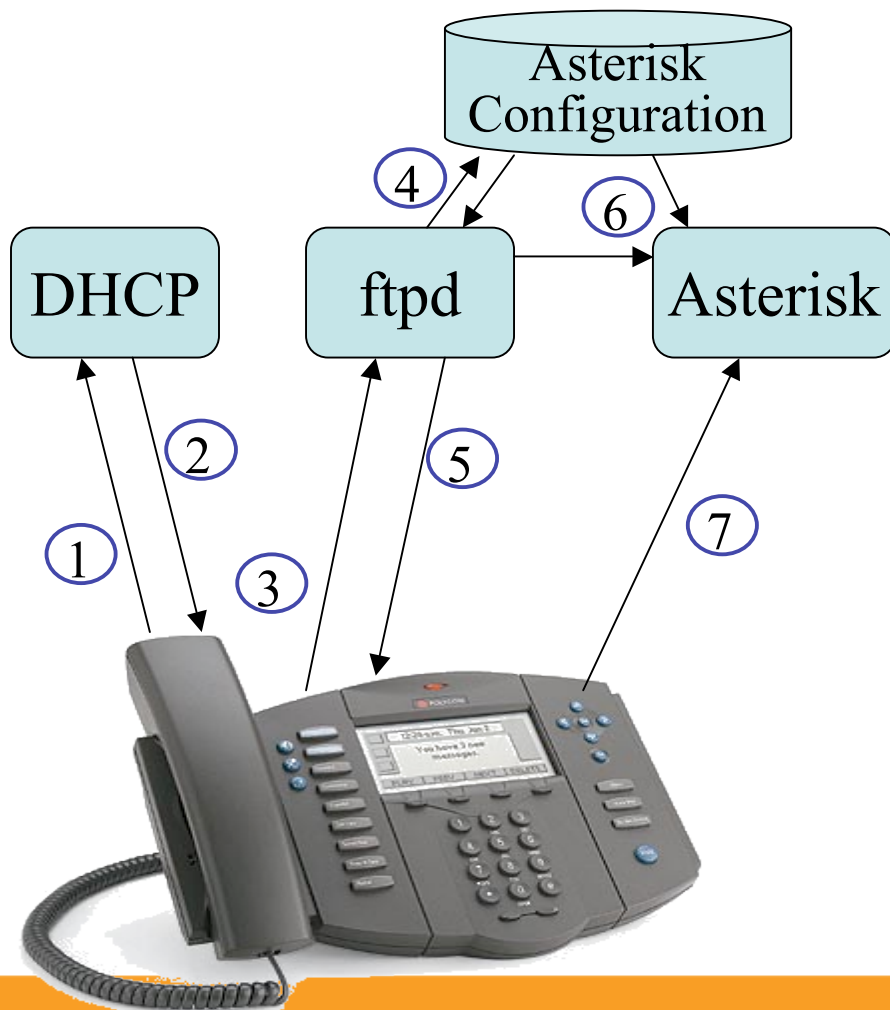


Asterisk Configuration Files

- extensions.conf
 - #include poly/extensions.conf
 - #include poly/provision.conf
- sip.conf
 - #include poly/poly_*.conf
- voicemail.conf
- sip_notify.conf

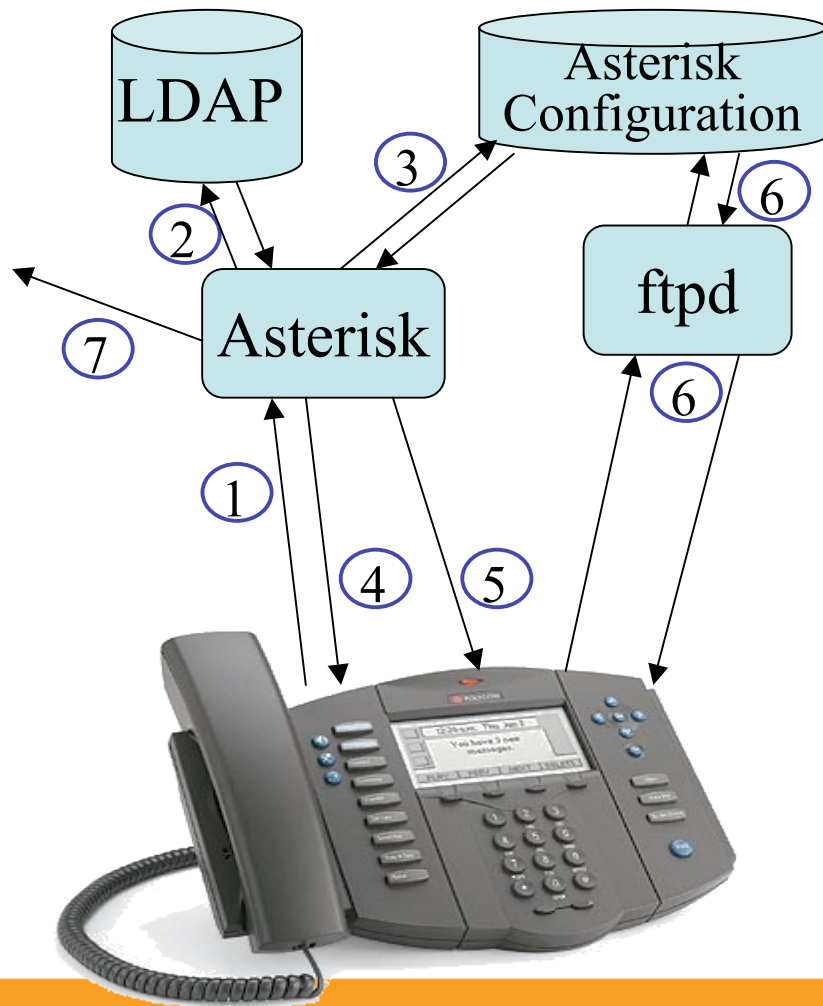


Pre Provisioning



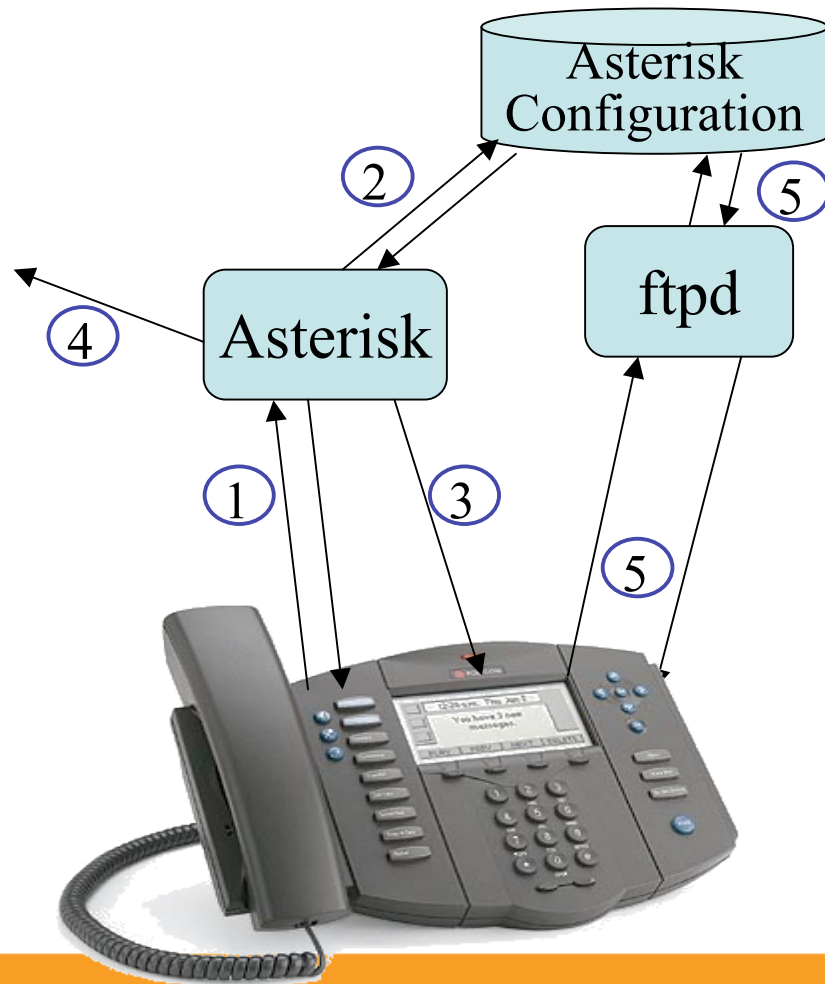
- 1) Request DHCP address
- 2) Supply network parameters, including ftp boot server
- 3) Request phone configuration using MAC address
- 4) Request SIP configuration, create as needed using MAC to derive name
- 5) Update software (if needed)
- 6) Reload SIP configuration
- 7) Register with Asterisk

Post Provisioning



- 1) Call "611", collect employee credentials
- 2) Verify credentials, collect employee information
- 3) Assign extension, update phone configuration
- 4) Confirm new settings to user
- 5) "Reboot" the phone
- 6) Fetch new phone parameters
- 7) Send email confirmation to user and administrator

Un-provisioning



- 1) Call "611", collect admin credentials
- 2) Modify Asterisk Configuration, reload Asterisk
- 3) "reboot" the phone
- 4) Send email confirmation to administrator
- 5) Fetch new phone parameters

Summary

- Statistics
 - FTPd additions: < 300 lines of TCL
 - provision agi: < 250 lines of TCL
 - dialplan additions: < 75 lines
- Conclusions
 - Need corporate wide employee “pin”
 - Works swell, integrates gracefully with changes to config files made by hand

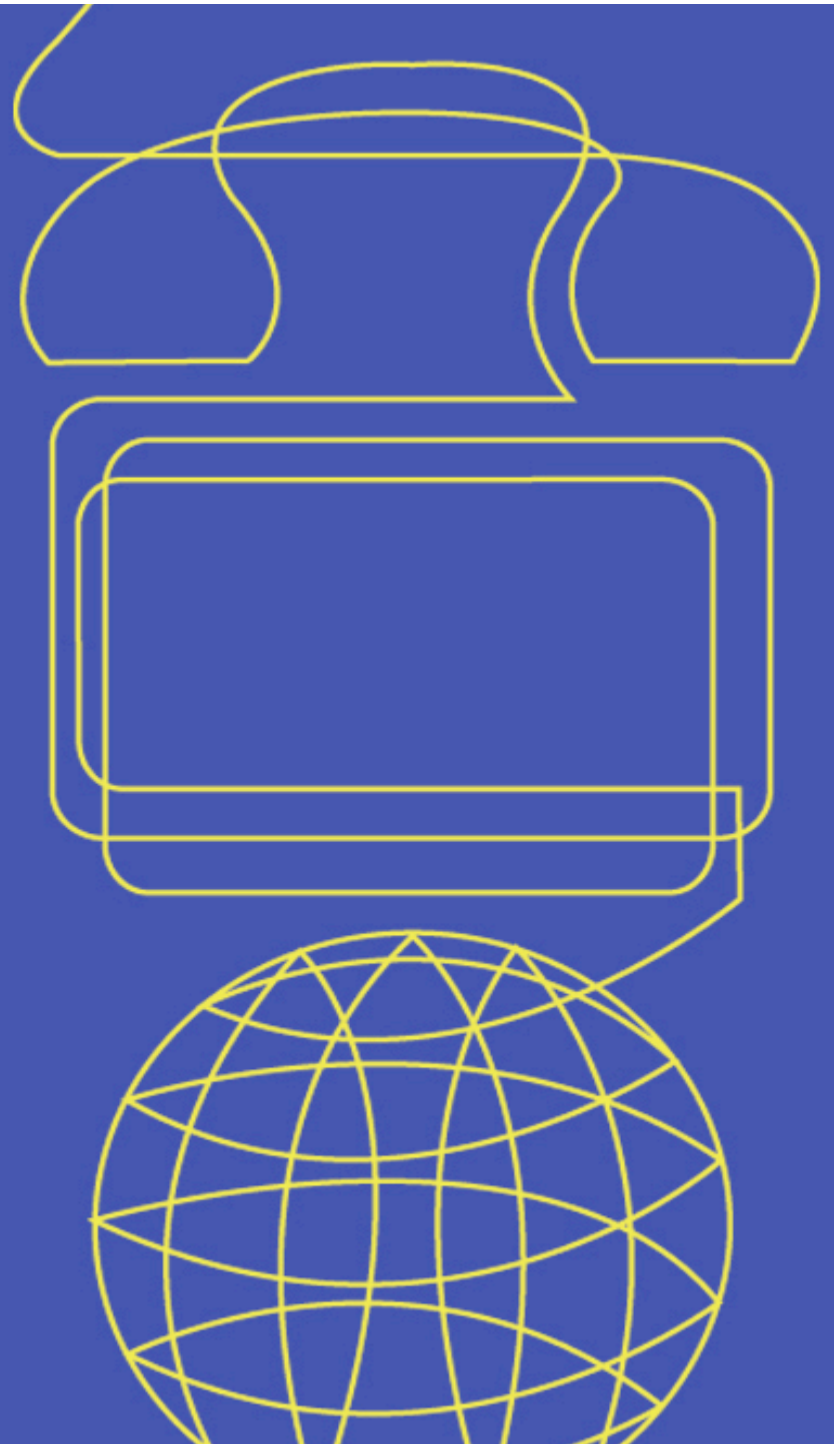


O'REILLY®

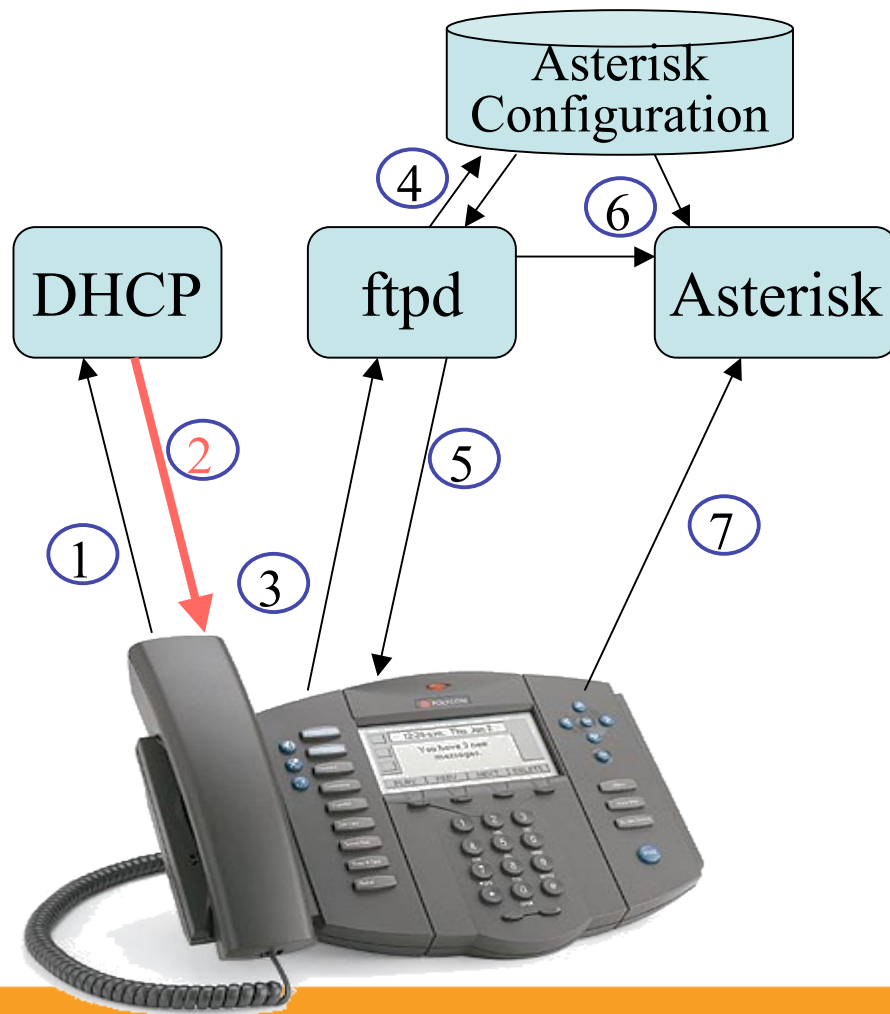
EMERGING TELEPHONY CONFERENCE

JANUARY 24-26, 2006
SAN FRANCISCO AIRPORT MARRIOTT

Boring Details



Pre Provisioning - DHCP configuration

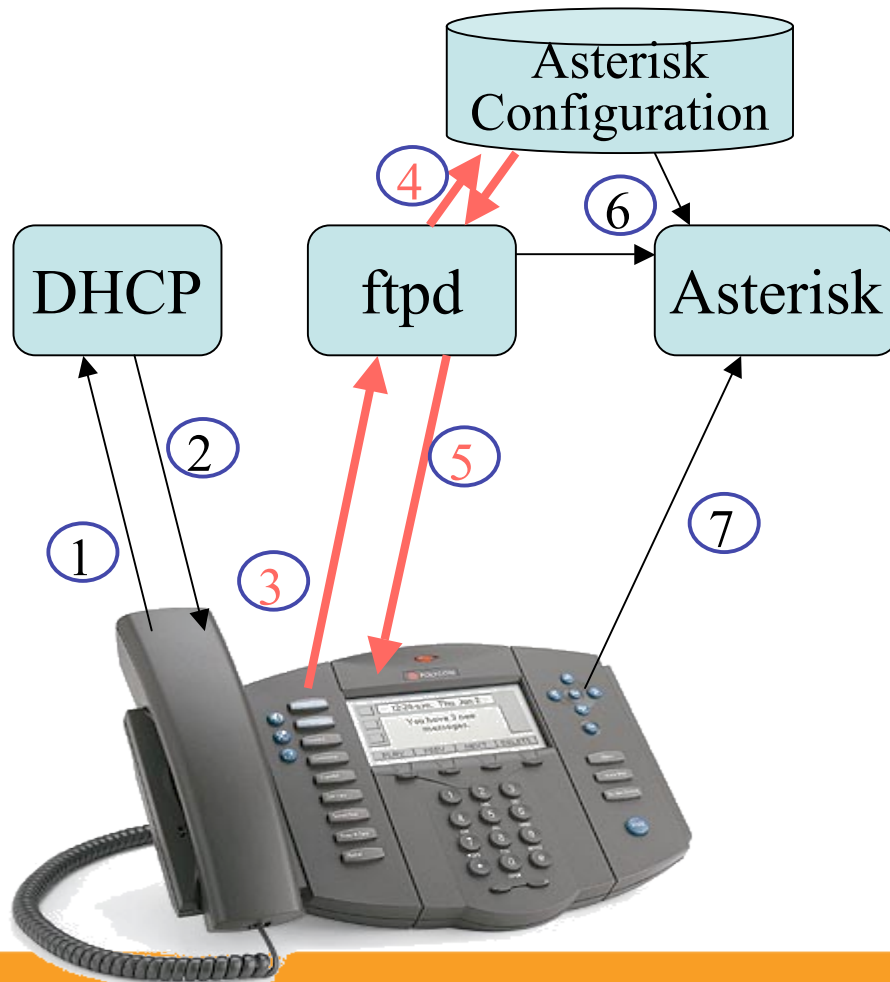


```
class "mac-address-match" {
    match substring(hardware, 1, 3) ;
}

option tftp-server-name code 66 = text ;
option tftp-server-address code 150 = ip-address ;

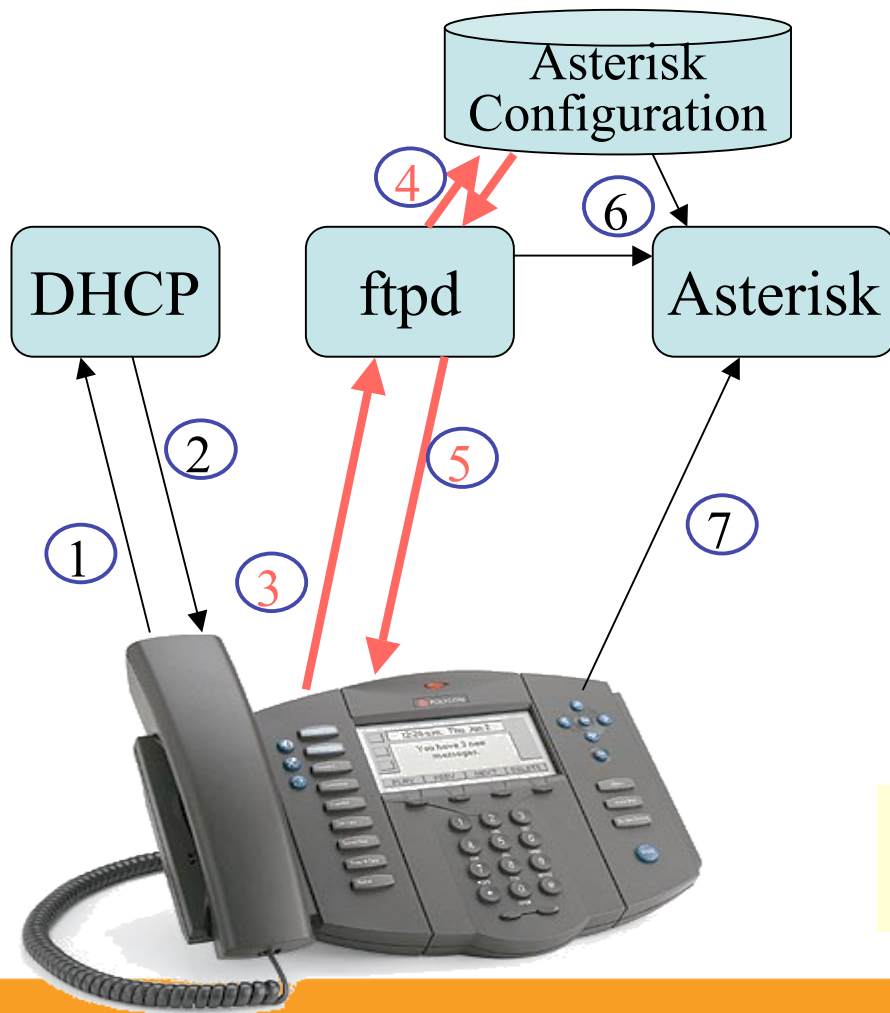
subclass "mac-address-match" 00:04:f2 {
    option tftp-server-address 152.70.67.50 ;
    option tftp-server-name "152.70.67.50" ;
}
```


Pre Provisioning – FTPd changes



- **Security**
 - Assign/accept passwords
 - Only deliver phone files out of a specific directory
- **Dynamic Content**
 - Create or obtain sip.conf entries
 - Map sip.conf to phone XML files
 - Notify Asterisk of new phones

Pre Provisioning – config generation



3) 0004f20116c9.cfg ?

4) poly/poly_6c9.conf

```
[poly_6c9]
username=[poly_6c9]
secret="*****"
...
context=unprovisioned
```

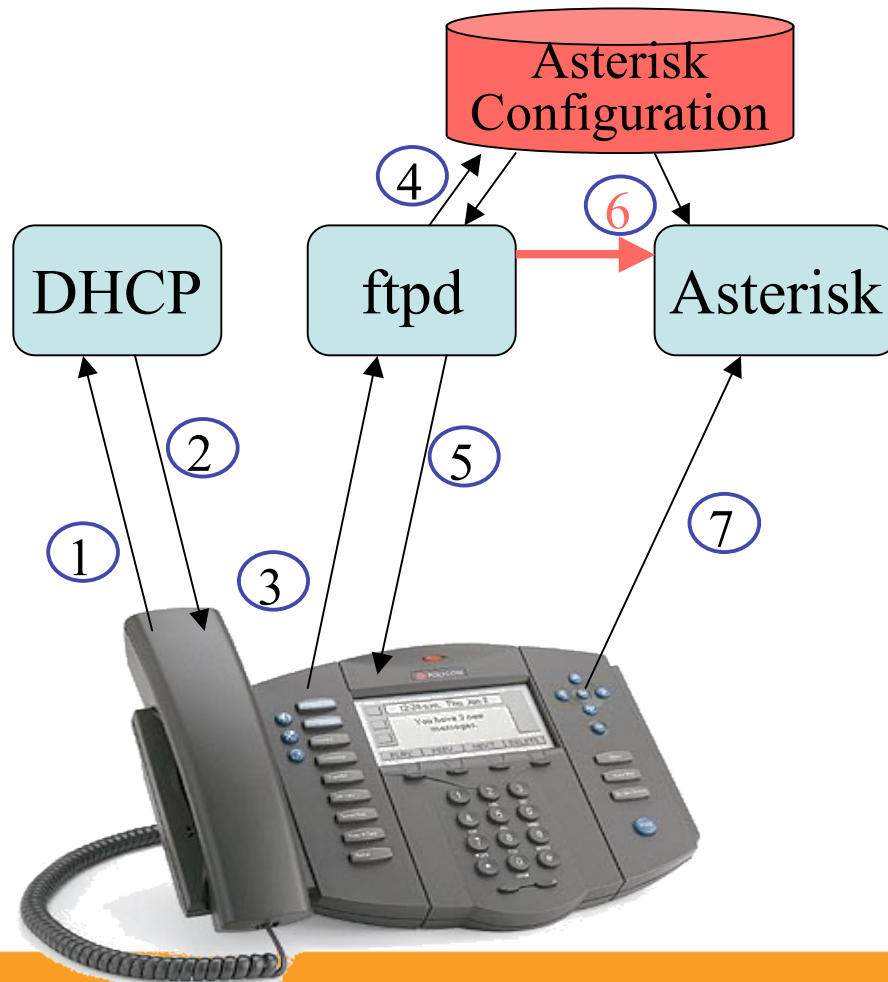
5) 0004f20116c9.cfg

```
<APPLICATION APP_FILE_PATH="sip.ld"
CONFIG_FILES="phone-6c9.cfg, sip.cfg"
LOG_FILE_DIRECTORY="logs"/>
```

5) phone-6c9.cfg

```
<phone_6c9>
<reg reg.1.userId="poly_6c9" reg.1.secret="*****" />...
</phone_6c9>
```

Pre Provisioning – asterisk notification

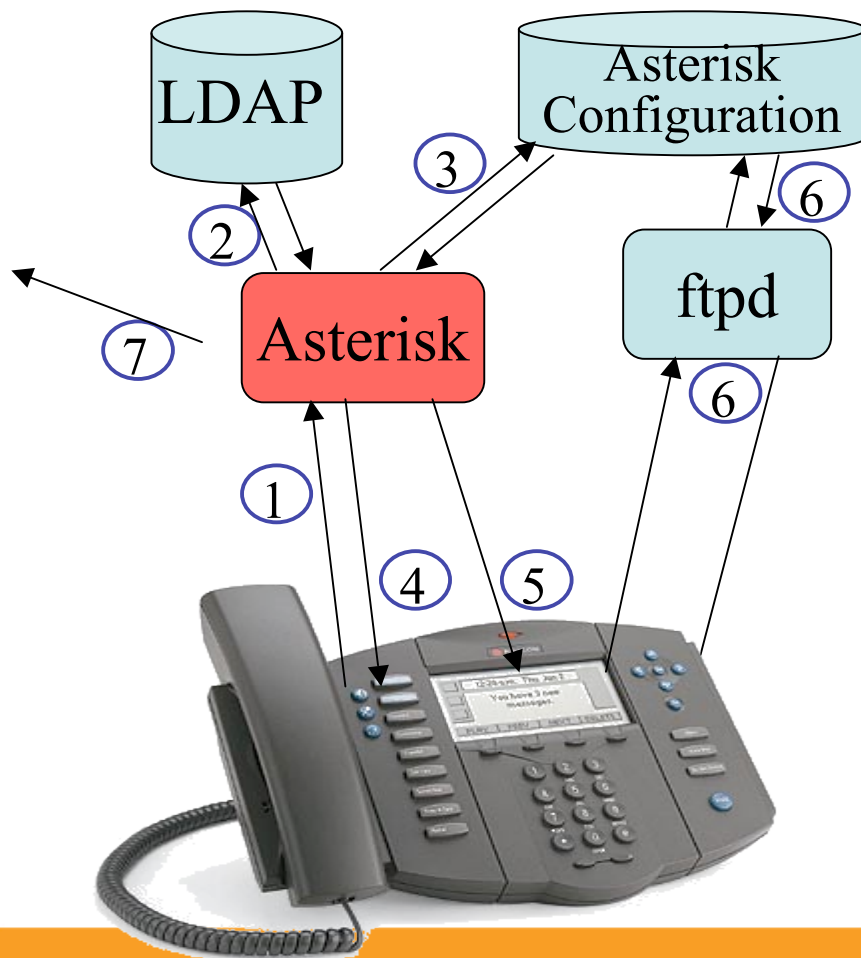


6) asterisk -rx "sip reload"

extensions.conf

```
[unprovisioned]
    include=test_numbers
    include=allow_provisioning
    ...
```

Post Provisioning – extensions.conf



[provision]

```
exten => s,1,Playback(provision/welcome)
exten => s,2,background(provision/enter_id)
```

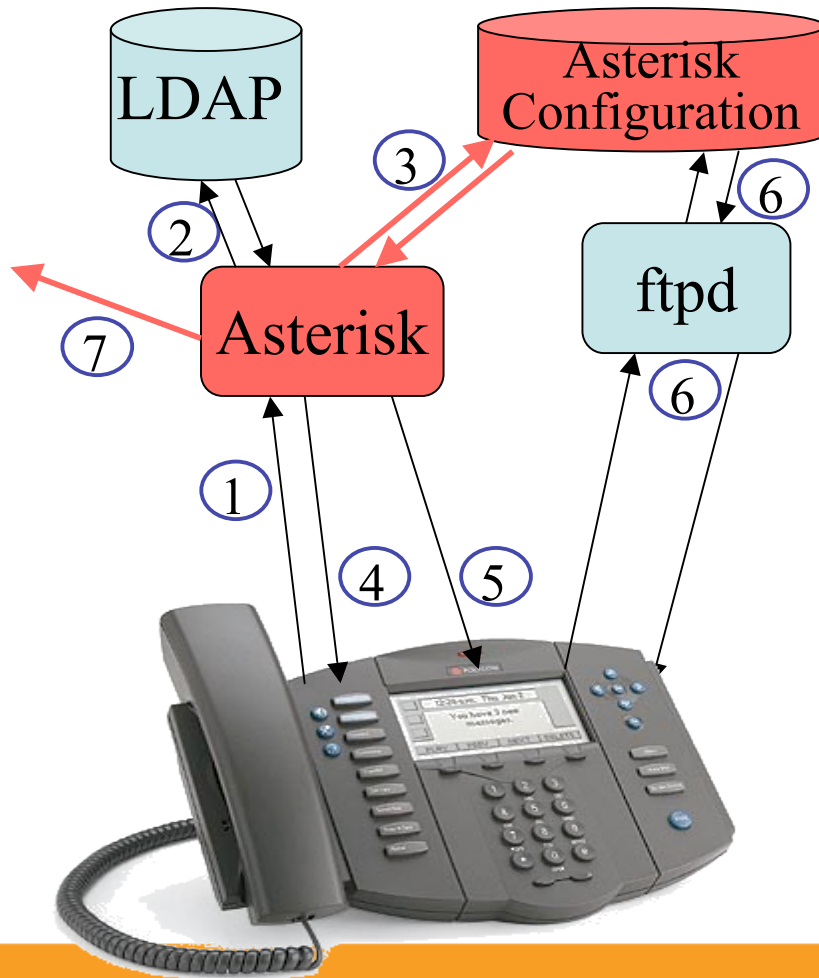
```
exten => _X,1,read(id); got 1st digit, get the rest
exten => _X,2,Setvar(id=${EXTEN}${id})
exten => _X,3,Playback(provision/entered_ok?)
exten => _X,4,saydigits(${id})
exten => _X,5,Goto(provision-2,s,1)
```

[provision-2]

```
exten => s,1,Playback(provision/ok_then_1)
exten => 1,1,agi(provision|${CHANNEL},${id})
exten => 1,2,Gotoif(${found} = 1)?20:30
```

```
exten => 20,1,Playback(provision/welcome2)
exten => 20,2,Saydigits(${extension})
exten => 20,3,Playback(provision/bye)
```

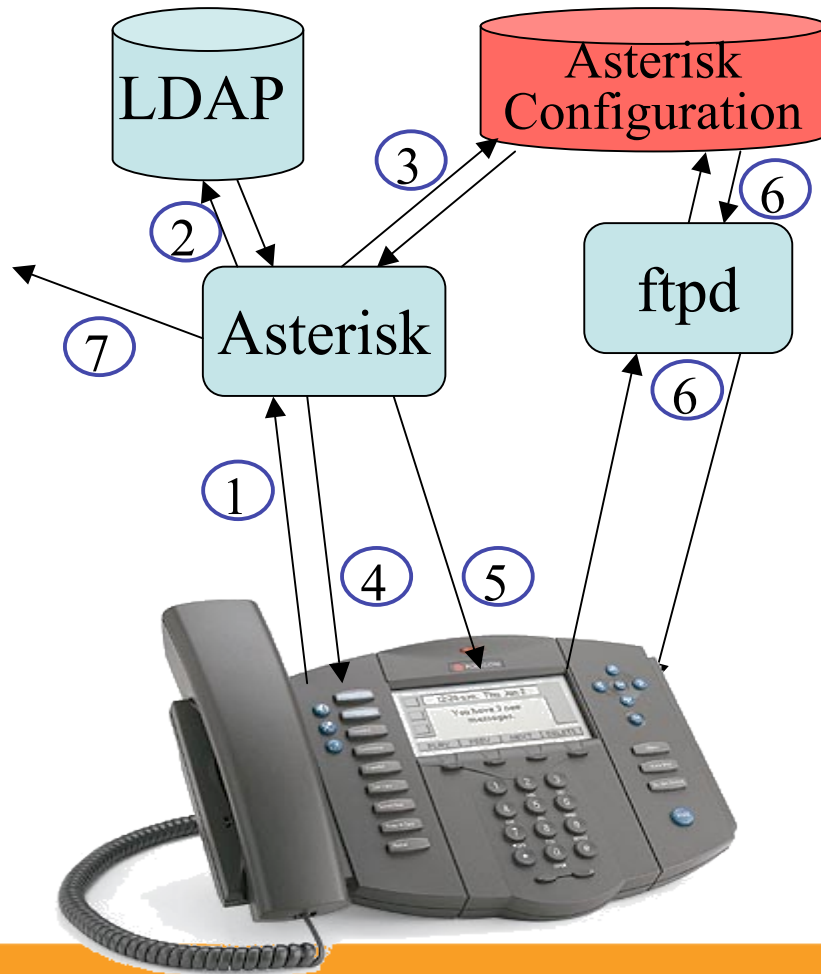
Post Provisioning – provision AGI



- Verify AGI arguments
 - employee-id
 - channel
- Retrieve employee data
 - email, name, manager
- Choose an extension
- Create config files from templates
 - sip.conf
 - extensions.conf
 - voicemail.conf
- Reload asterisk configurations
- Reboot the phone (via sip info)
- Send verification email
- Set employee data into channel variables

Post Provisioning – config templates

sip.conf template



```
; Modified by auto-provision: $date  
[poly_$id]  
type=friend  
secret=$pass  
host=dynamic  
mailbox=$ext  
callerid=$display <$ext>  
dtmfmode=rfc2833  
context=$context
```