VOVIDA.ORG

Vovida Open Communication Application Library
VOCAL

System Architecture

What Is in This Module

Module Title:

VOCAL System Architecture

Objectives:

At the end of this module, you will be able to:

- **Describe the VOCAL system architecture**
- **Describe the functionality offered by VOCAL**
- **Describe the components of the VOCAL system and how they interact**
- **Understand SIP call flows**

Module Length:

134 slides

VOVIDA.ORG

VOCAL Overview

VOCAL – What is it?

Vovida Open Communication Library (VOCAL):

- An open source, IP centric communication software, development platform and library.
- It runs on:
  - Linux and Solaris operating systems.
  - Intel (I86) based hardware.

VOCAL – What It Offers

VOCAL provides:

Feature and Application Creation	Operation System Support
SIP Based Call Control and Switching	

SIP Based Call Control & Switching

VOCAL offers SIP based call control and switching:

- **User registration.**
- **Call initiation.**
- **Call modification.**
- **Call termination.**

Operation Support Services

The operating support services within VOCAL provide the capabilities to:

- **Provision or configure the VOCAL system from a web GUI.**
- **Monitor network elements from an SNMP network manager.**
- **Add and manage subscribers and their feature subscriptions.**
- **Authenticate subscribers.**
- **Track billing information.**

Feature and Application Platform

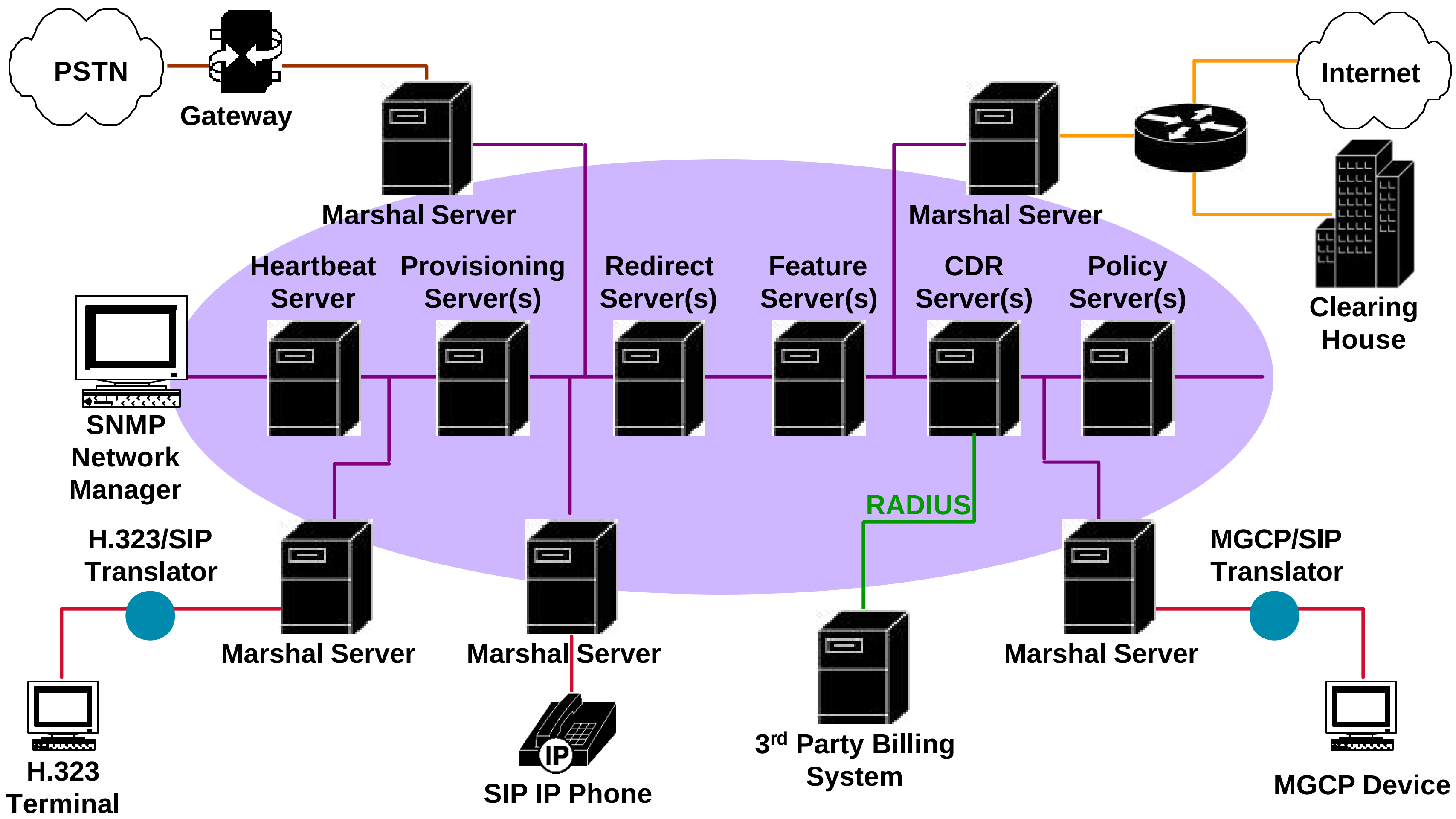
VOCAL provides:

- Basic features such as call forward, call blocking, call transfer, and call waiting.
- A software library for new feature and application creation in:
  - C++.
  - Call processing language (CPL).
  - Java telephony API (JTAPI).

VOVIDA.ORG

VOCAL System Architecture

VOCAL Architecture



A Basic SIP Call Using the VOCAL System (1)

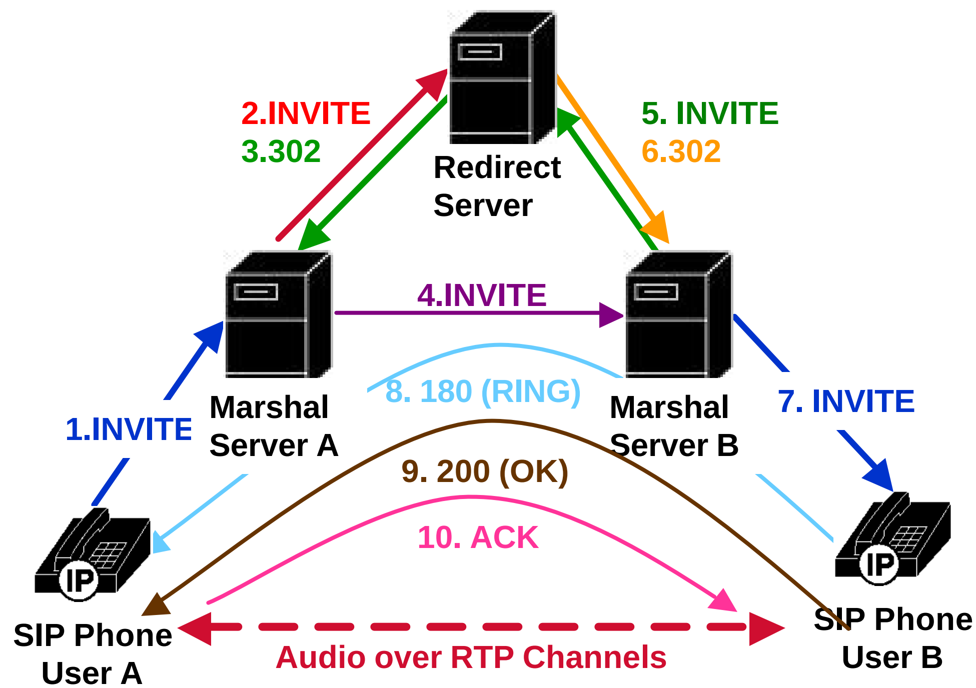
1.User A dials user B's number. User A's SIP phone sends an INVITE to marshal server.

2.Marshal server A forwards the INVITE to the redirect server.

3.The redirect server responds with a 302 containing information for marshal server A to contact marshal server B.

4.Marshal server A forwards an INVITE to marshal server B.

5.Marshal server B forwards the INVITE to the redirect server.



A Basic SIP Call Using the VOCAL System (2)

6. The redirect server responds with a 302 containing information for marshal server B to contact user B.

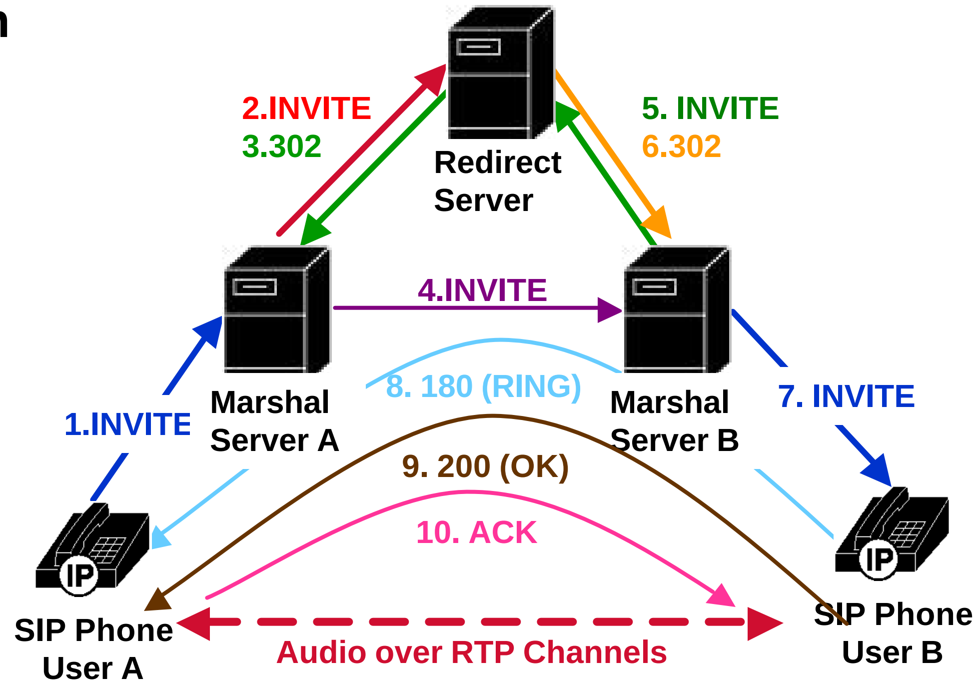
7. Marshal server B sends a INVITE to user B.

8. User B's SIP phone rings. The 180 message is sent back to user A's SIP phone.

9. When user B picks up the SIP phone, a 200 (OK) message is sent.

10. User A's SIP phone responds with an ACK message.

11. The RTP path is now established.



VOCAL Components

User Agent

User Agent

VOCAL supports SIP user agents including:

Cisco 7960 SIP IP Phone

Pingtel xpressa

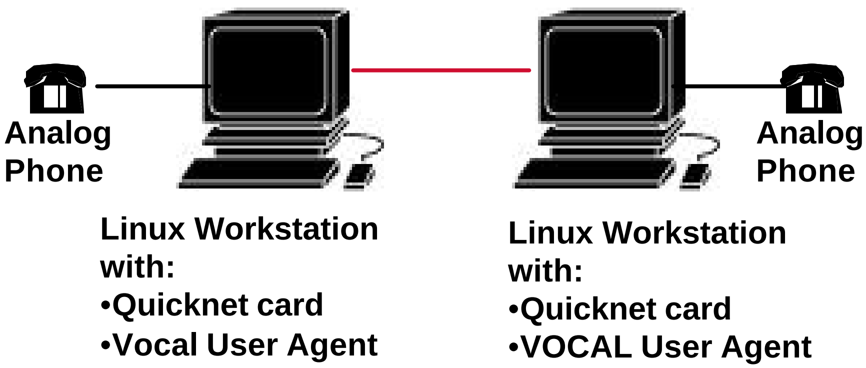
PC with softphone application

Komodo ATA 182/186

VOCAL User Agent

The VOCAL SIP user agent supports:

- **Call establishment.**
- **Call waiting.**
- **Transfer.**
- **Registration with a marshal server or SIP proxy server.**



VOVIDA.ORG

Marshal Server

Marshal Servers

The Marshal Servers:

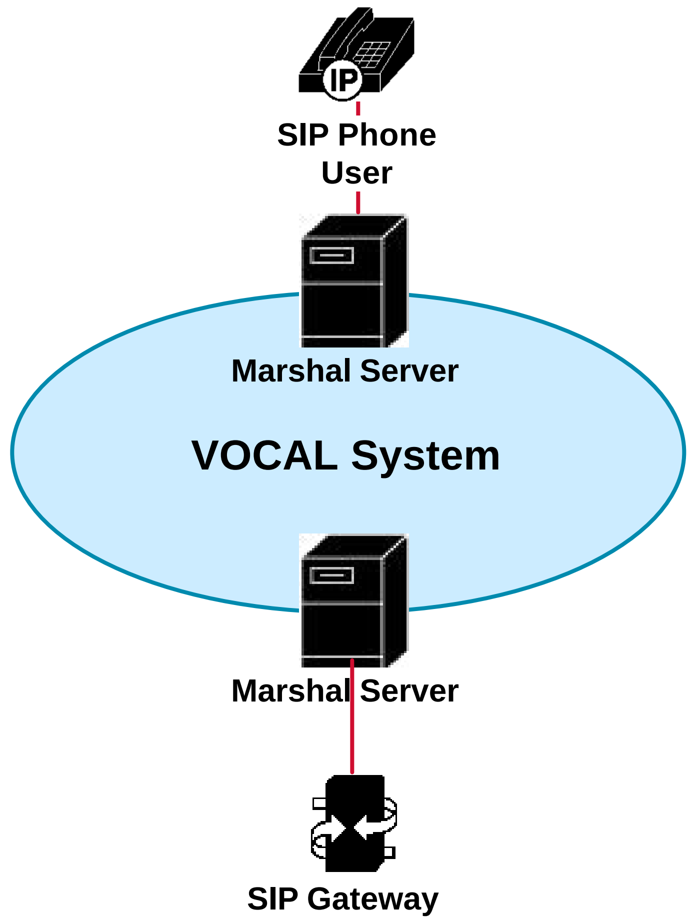
- Are the only point of contact for all external devices.
- Provides the logical function of the SIP proxy server and SIP registration server.
- Performs one or more of these functions:
  - SIP message translation.
  - Authentication and security.
  - Billing.

SIP Message Translation

The Marshal Server:

- Checks
- Translates
- Logs

all SIP messages it receives from external entities.

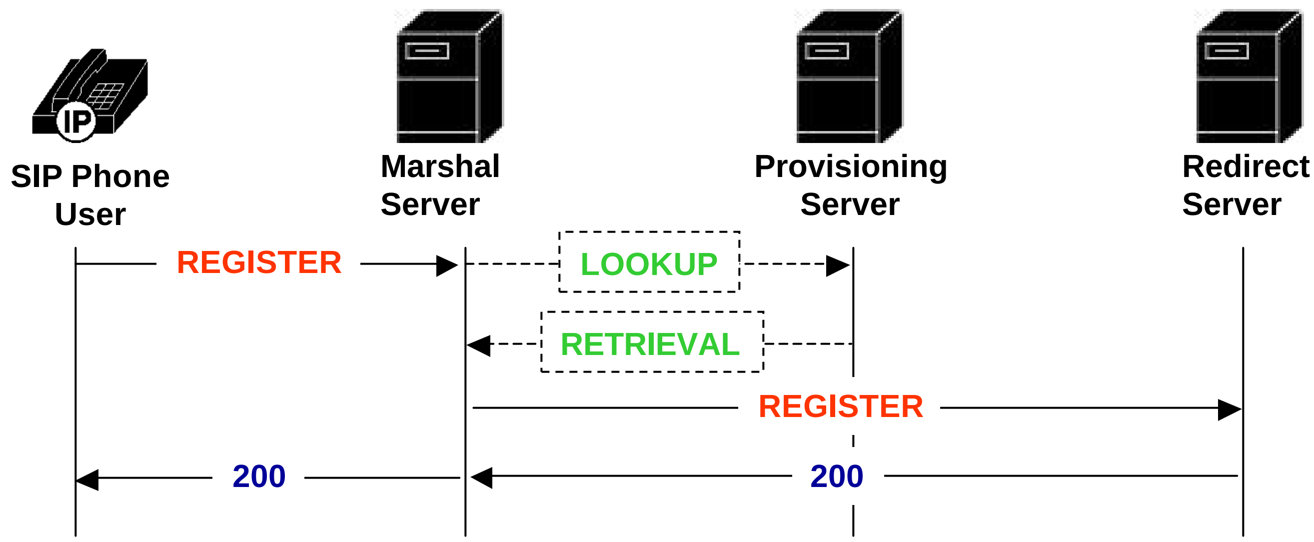


Marshal Functionality - Authentication

The Marshal Server supports these authentication methods:

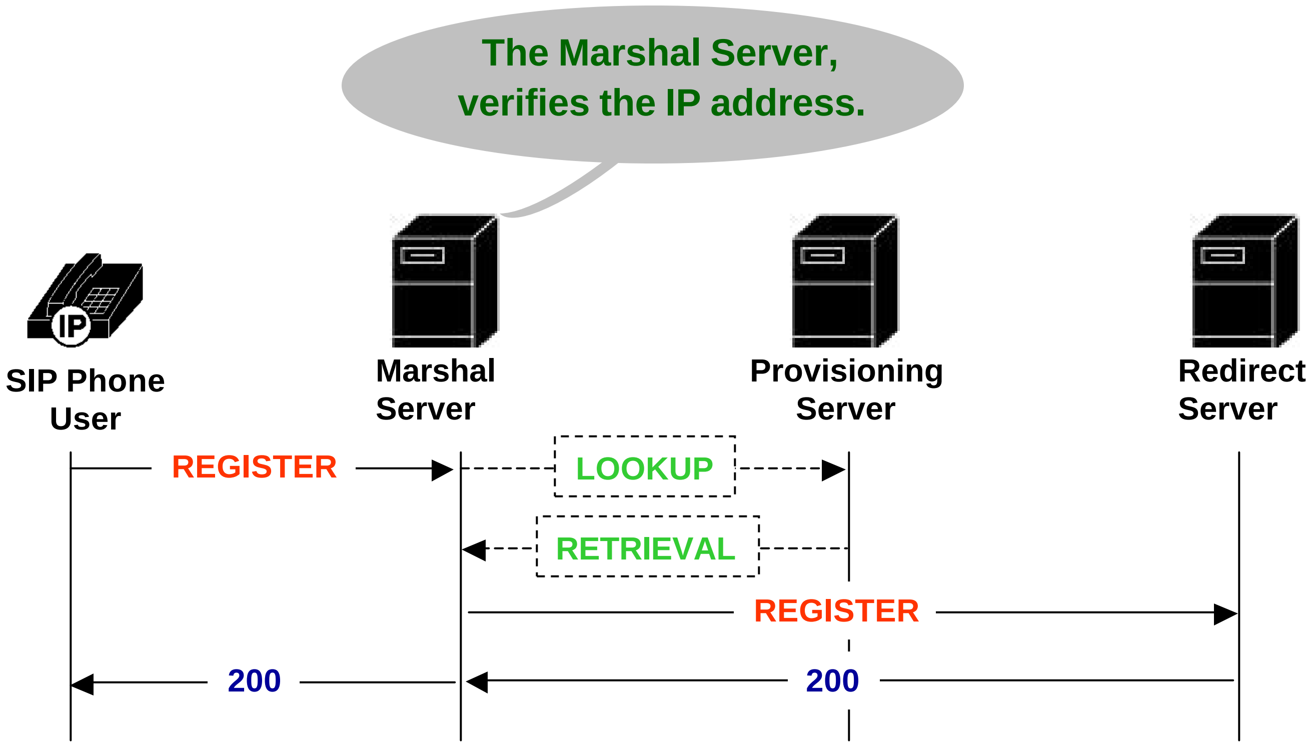
- **No authentication.**
- **Access control authentication – verification of IP address.**
- **HTTP Digest authentication – verification of username and password.**

No Authentication



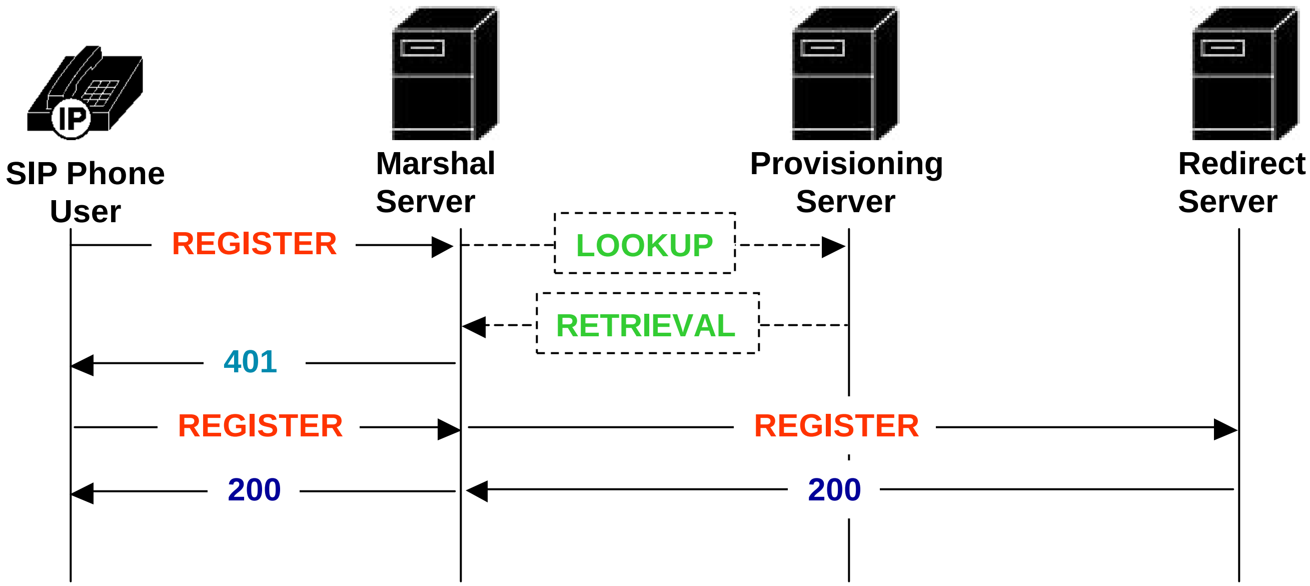
SIP Messages:
REGISTER – Registers the address listed in the To header field
200 – OK

Access List Authentication



SIP Messages:
REGISTER – Registers the address listed in the To header field
200 - OK

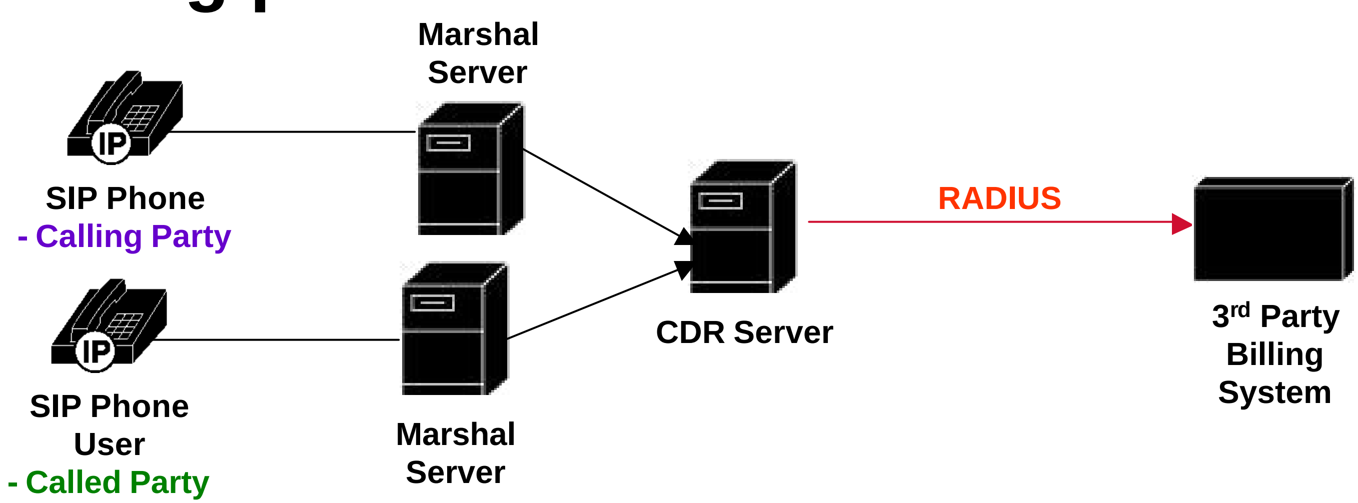
HTTP Digest Authentication



SIP Messages:
REGISTER – Registers the address listed in the To header field
200 – OK
401- Unauthorized

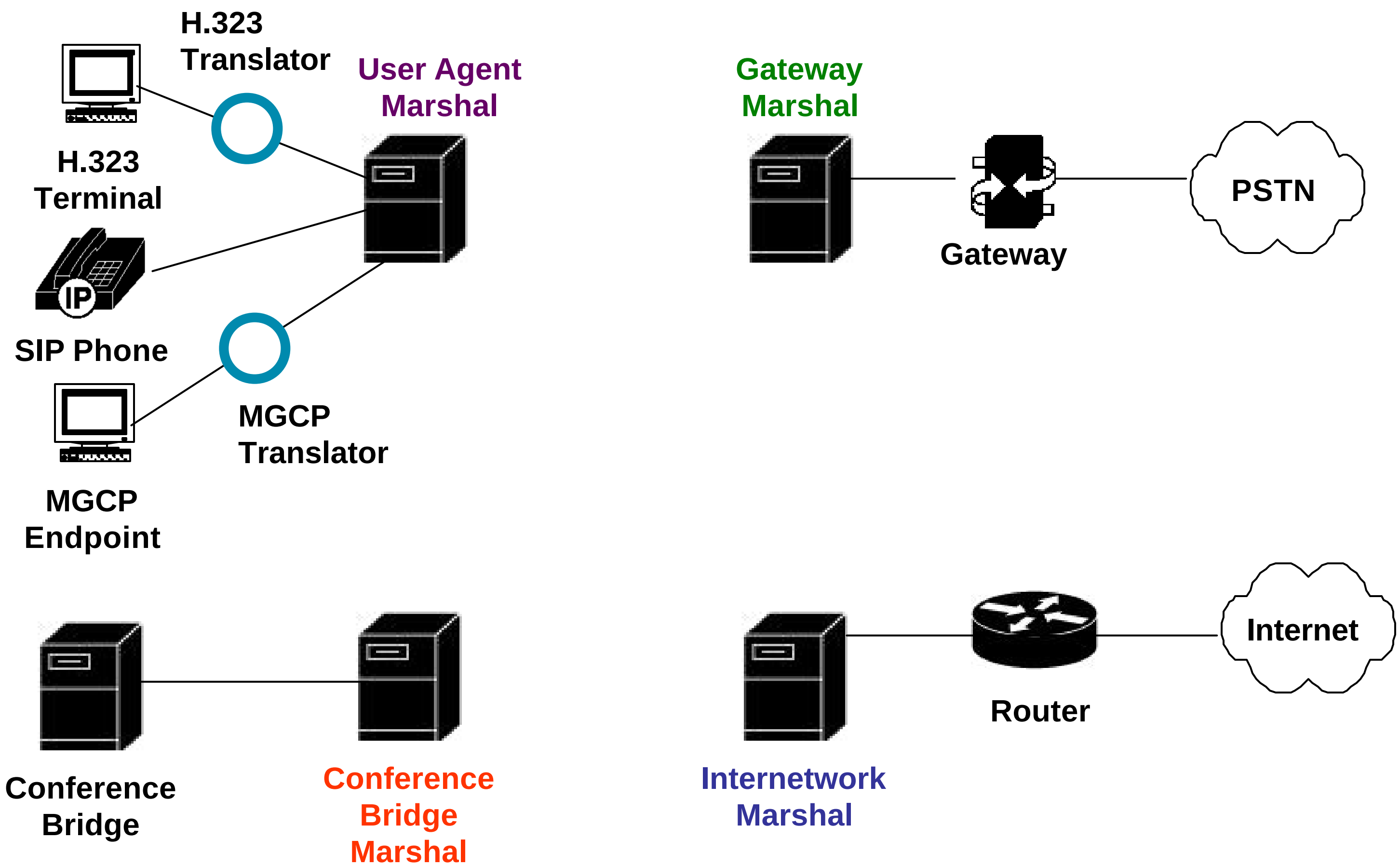
Marshal Functionality - Billing

- Each Marshal Server sends the start and stop time of a call to the CDR Server.
- The CDR Server forwards the data to 3^{rd} party billing systems using the RADIUS accounting protocol.



Types of Marshal Server

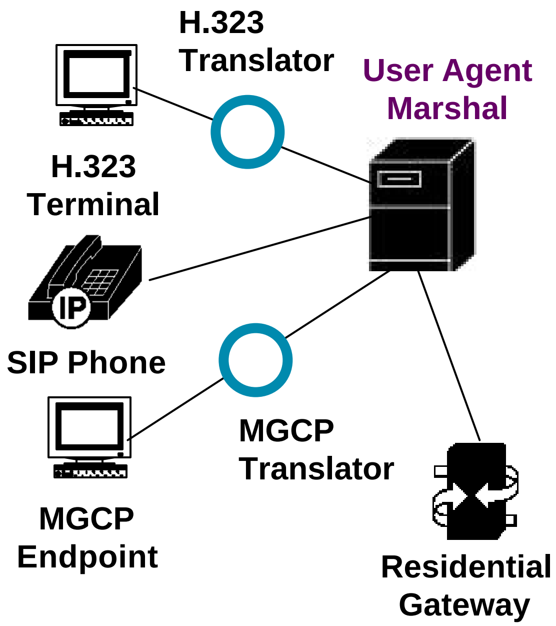
Types of Marshal Servers



User Agent Marshal

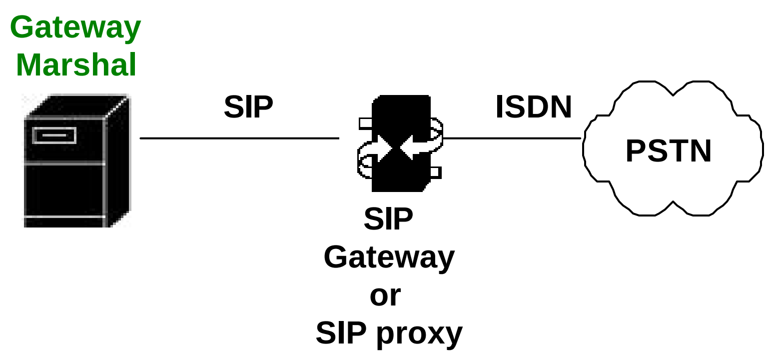
The User Agent Marshal Server:

- **Interact with User Agents.**
- **Receives INVITE messages from User Agent.**
- **Authenticates the user (against a user profile stored in a master file in the Redirect Server).**
- **Requests routing information from the Redirect Server.**



Gateway Marshal

- **Gateway Marshal Servers interact with SIP gateways or SIP proxy servers.**
- **Gateways provide translation or interconnection between the IP and the PSTN network.**



Conferencing

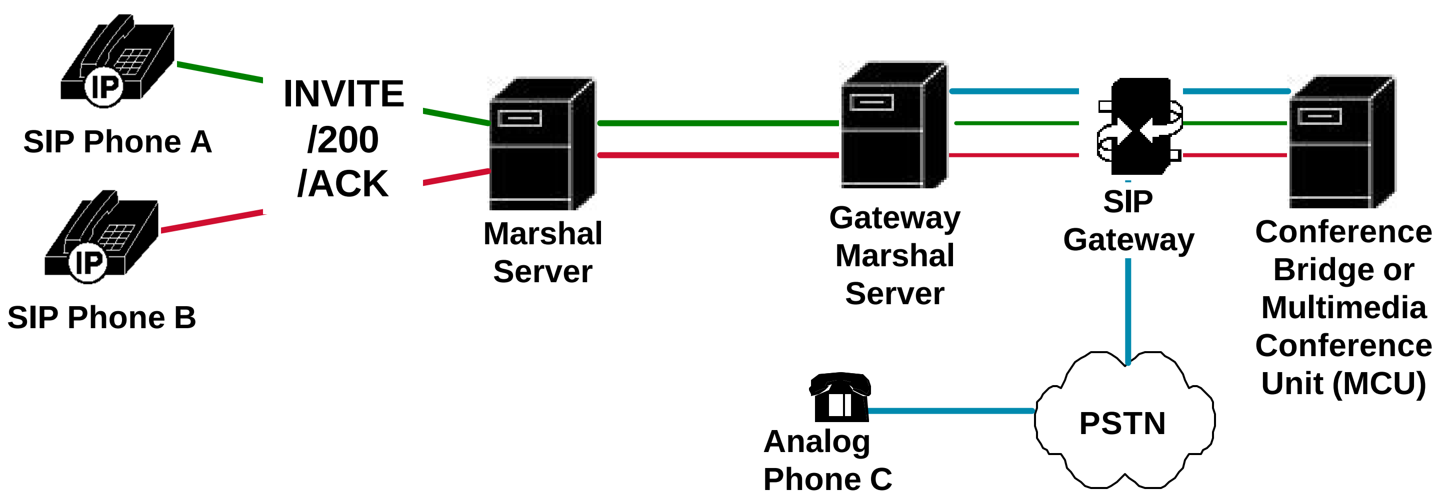
The VOCAL system supports two types of conferencing:

- Meet-Me – users call a predefined number at predefined time.
- Ad-Hoc – user adds multiple users to a call.

Ad-Hoc conferencing requires a Conference Bridge Marshal.

Meet-Me Conferencing

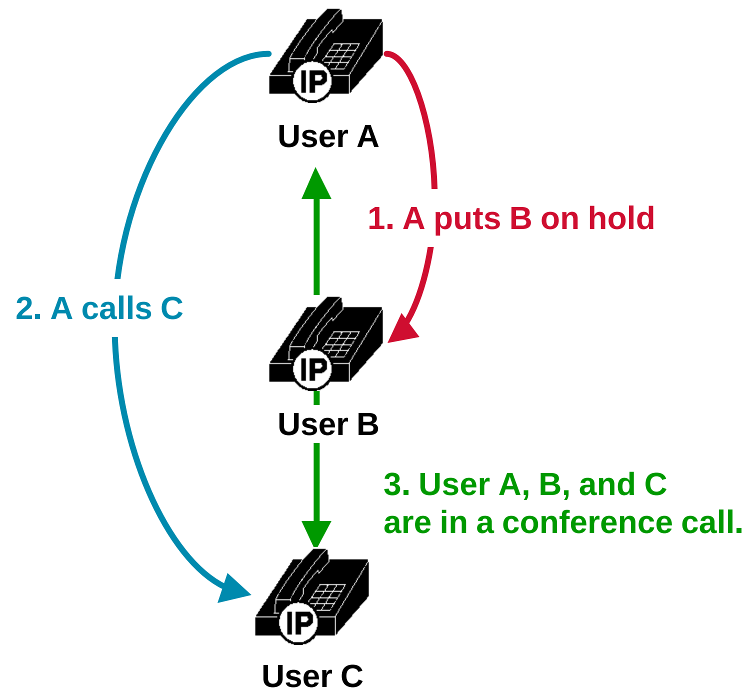
- Meet-Me conferencing allows any users to call a conference bridge number.
- RTP media channel is established for each user.
- The conference bridge mixes the audio streams.



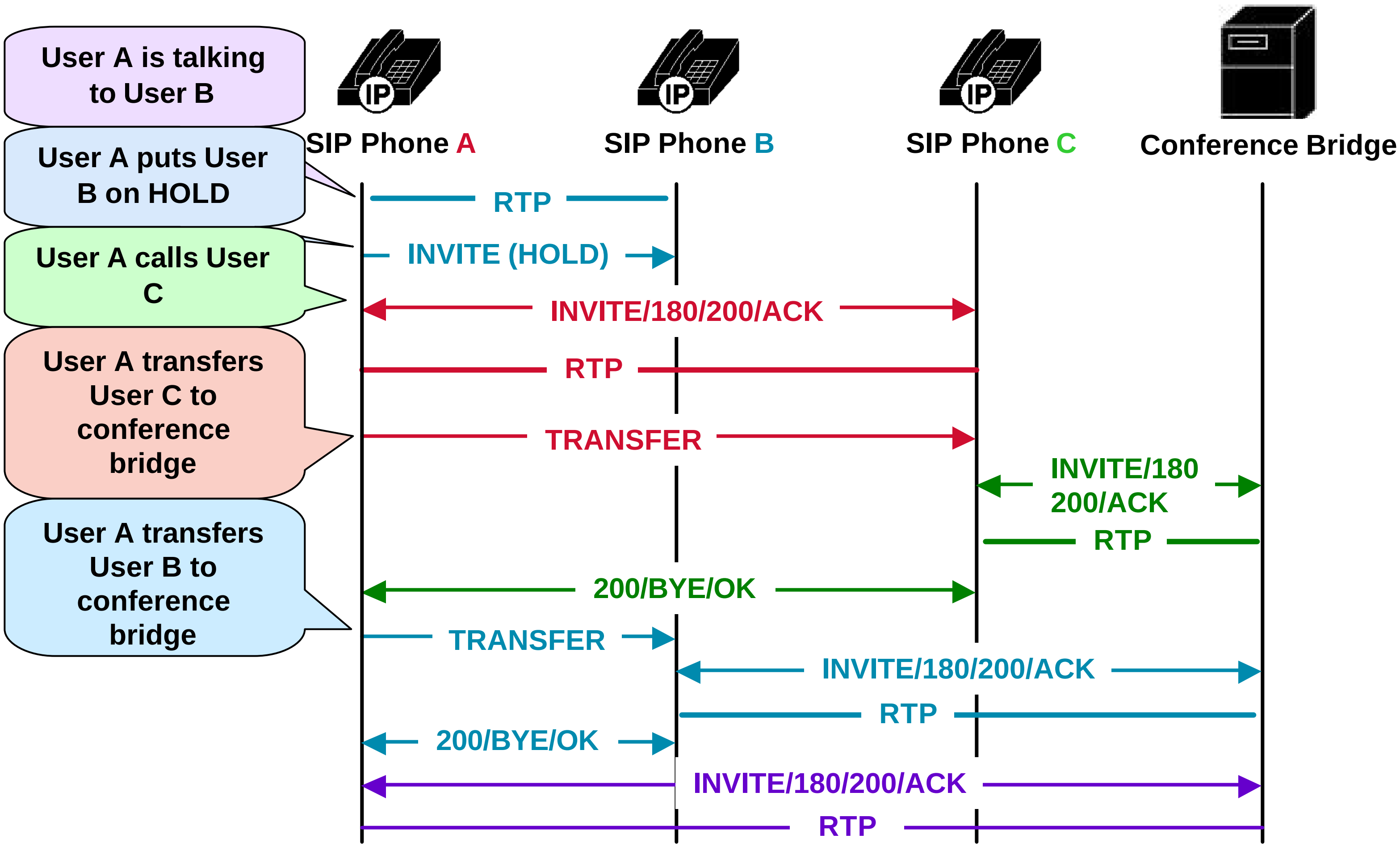
What is Ad-Hoc Conferencing?

With ad-hoc conferencing a user adds multiple participants to a call:

- **User A and User B are in a call. User A wishes to add User C to the call.**
- **User A places User B on hold and calls User C.**
- **User C answers.**
- **User A adds User C to the call with User B. The call is now a conference call.**



How would ad-hoc conferencing work with SIP?



Implementation Issues

The previous call flow diagram illustrates an ideal implementation. There are these implementation issues:

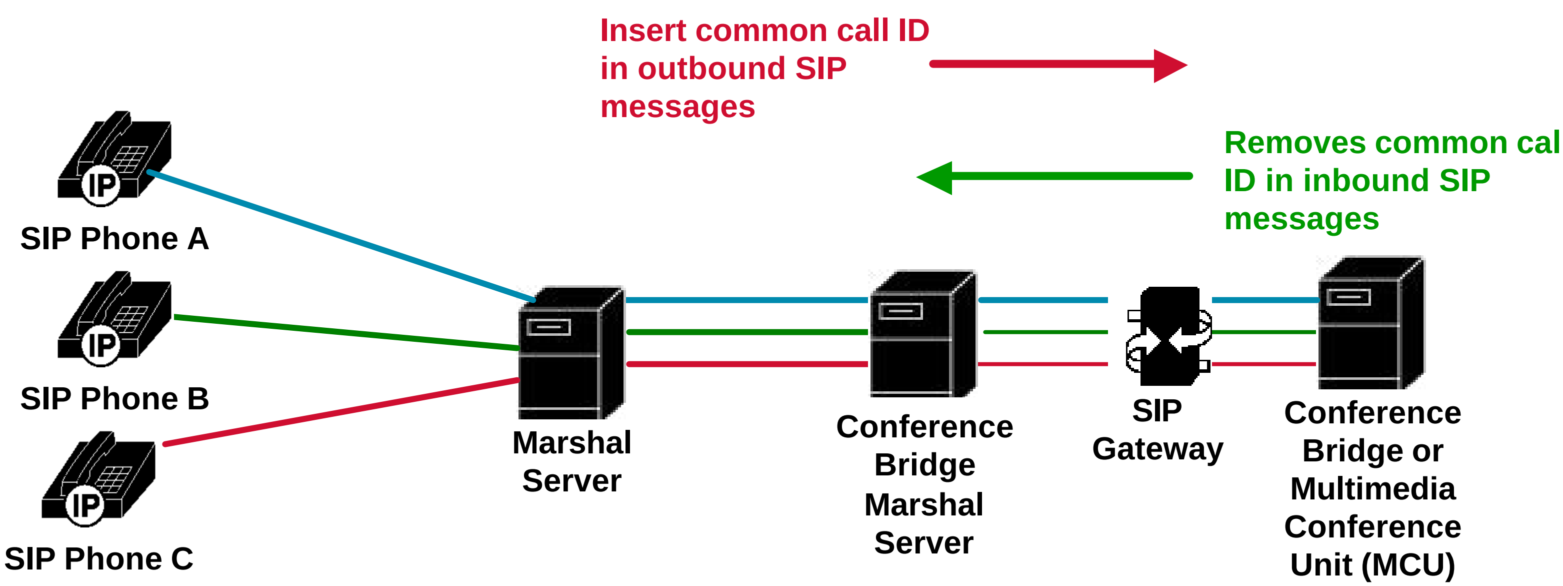
- Most conference bridges do not use SIP.
- Therefore a SIP gateway is required.
- However, most SIP gateway cannot handle multiple calls with the same call ID.
- All conference calls use the same call ID.

At the time of implementation, there was no SIP standard on conferencing.

VOCAL Solution – Conference Bridge Marshal Server

The Conference Bridge Marshal Server:

- Inserts a common call ID for outbound SIP messages to the SIP gateway.
- Removes the common call ID and inserts unique call IDs for inbound SIP messages from the SIP gateway.



Internetwork Marshal

The Internetwork Marshal Server is used to interconnect with:

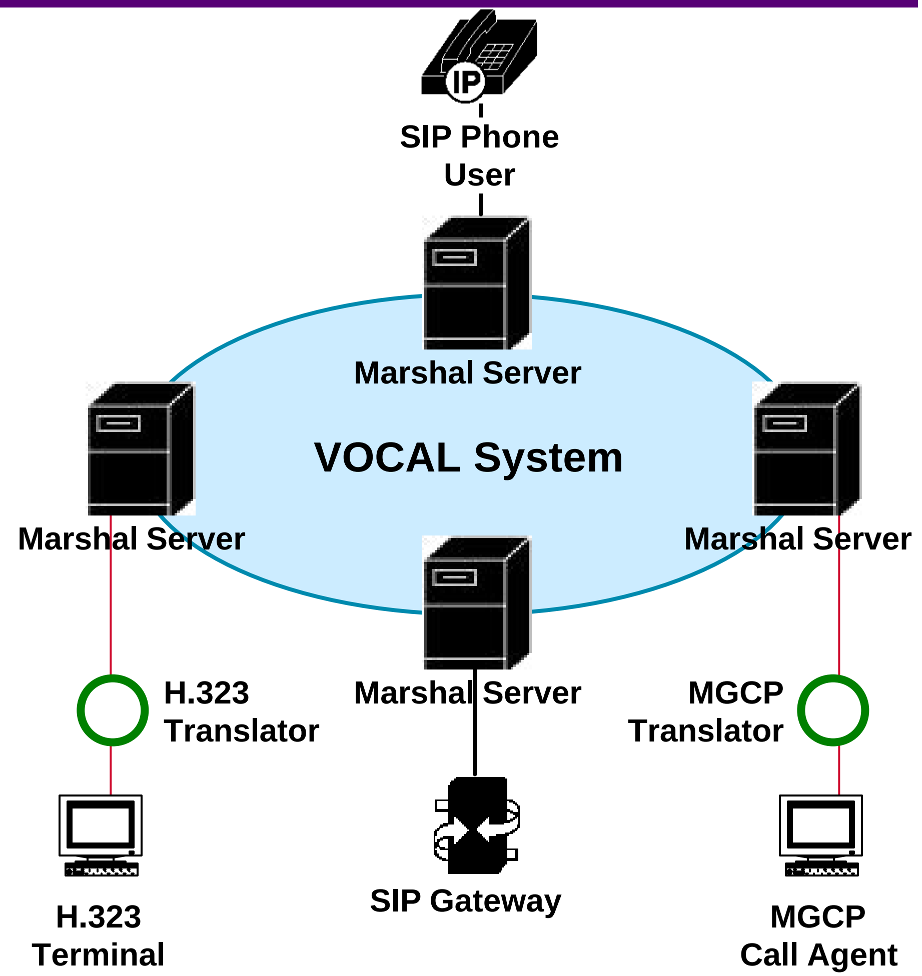
- **Other SIP systems that use the OSP protocol.**
- **Clearinghouses.**

VOVIDA.ORG

Translators

Translator Functionality

- **VOCAL is SIP based.**
- **Supports non-SIP endpoints using translators.**
- **Supports H.323 and MGCP endpoints.**



H.323 Translator

- Provides call signaling translation between H.323 endpoint and SIP server.
- H.323 endpoints appear as SIP user agents to the VOCAL system.
- The current implementation works with Microsoft NetMeeting 3.01 as the H.323 endpoint.

MGCP Translator

- **Provides call signaling translation between MGCP endpoint and SIP server.**
- **MCGP translator can act like a MGCP call agent that controls MGCP gateways.**

Provisioning Server

Provisioning Server

The Provisioning Server:

- Stores data on all users and servers within the VOCAL system.
- Accessible from a Java-based GUI via an Internet browser.

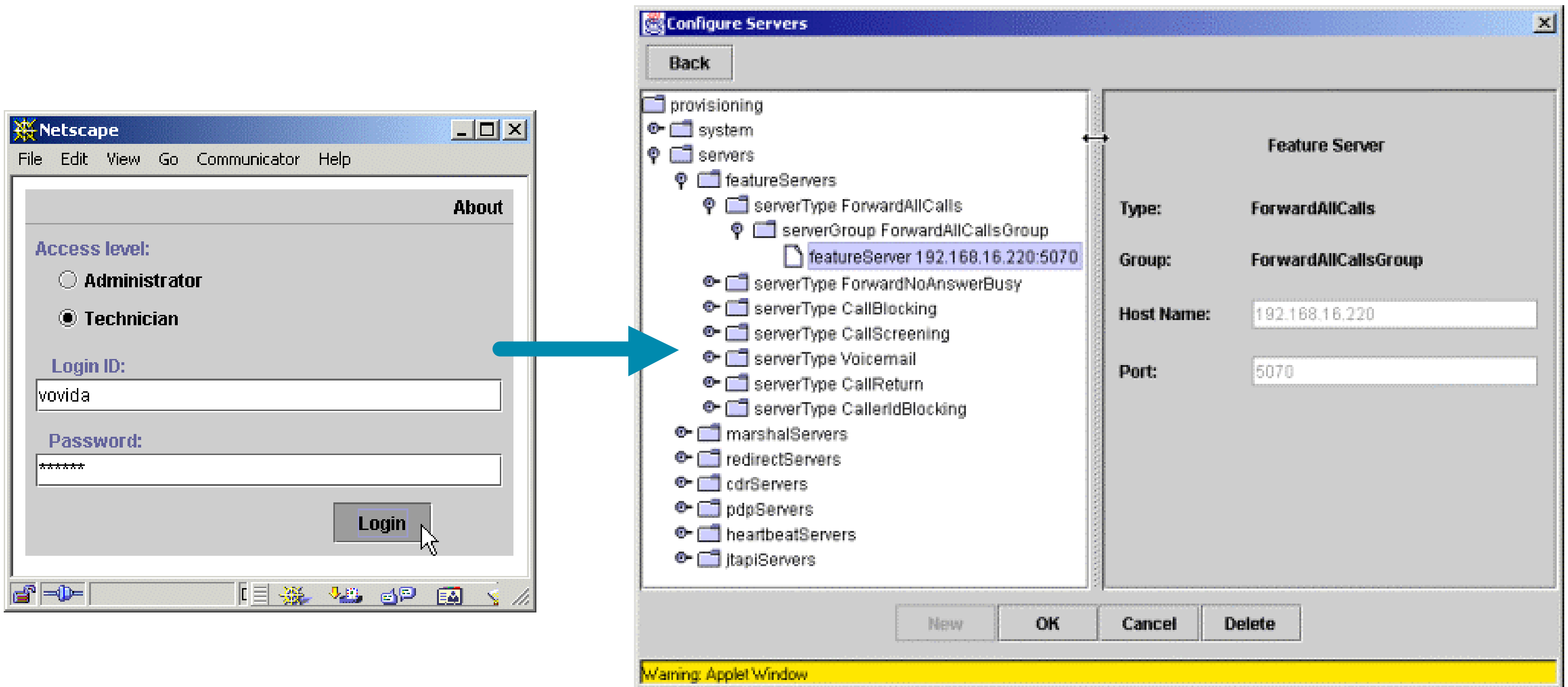
Provisioning GUI

The Provisioning GUI is used to:

- **Configure the VOCAL system.**
- **Administer users and enable user's features.**
- **Subscribe or unsubscribe user's features.**

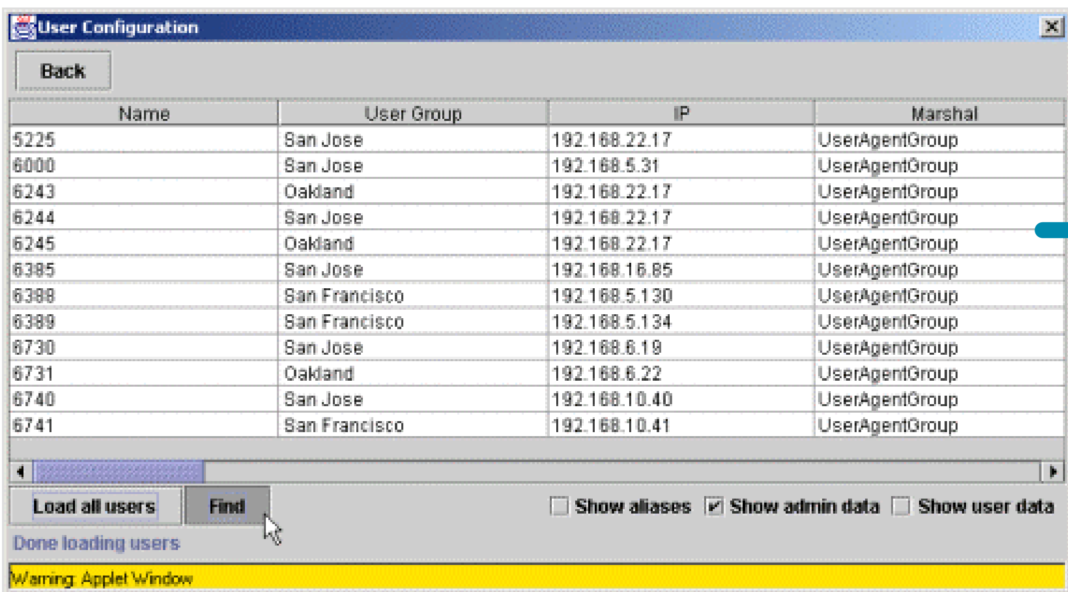
Technician Screen

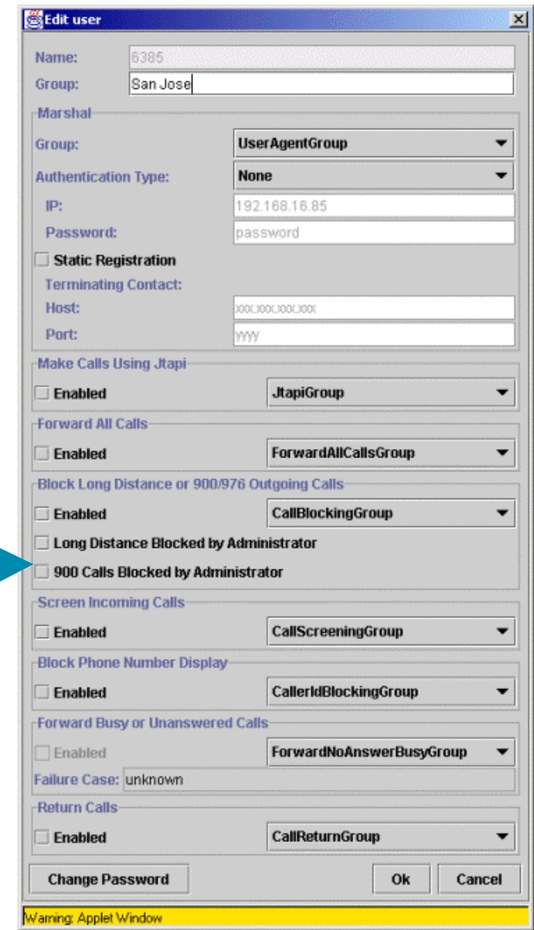
- The Technician screens allows you to configure or provisioning the VOCAL servers.



Administrator Screen

- **The Administrator screen allows you to add users and enable their features.**

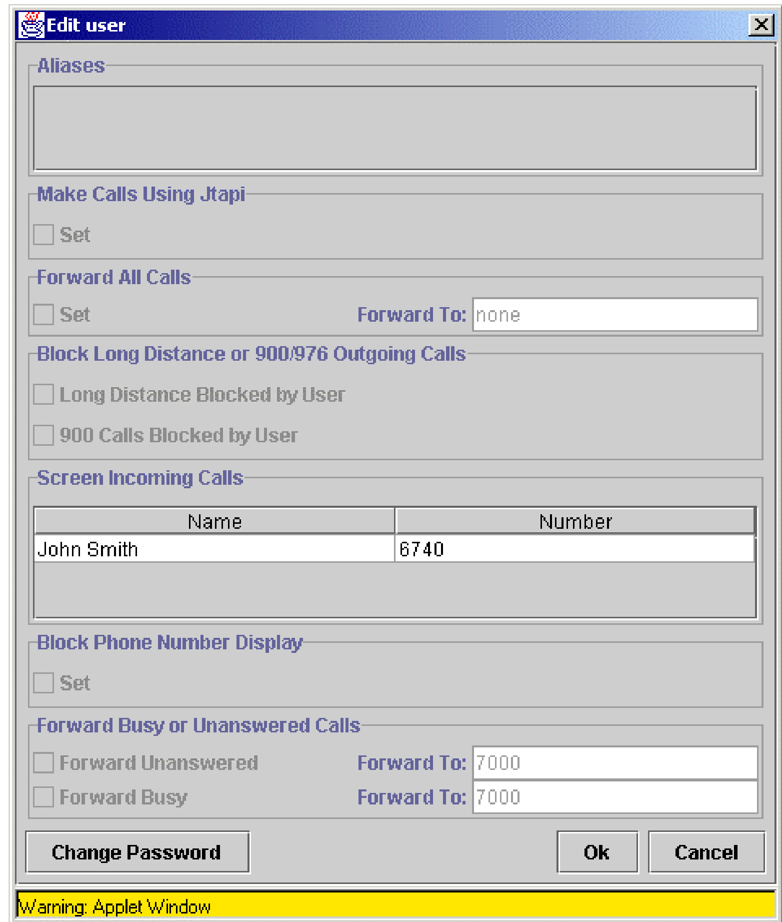




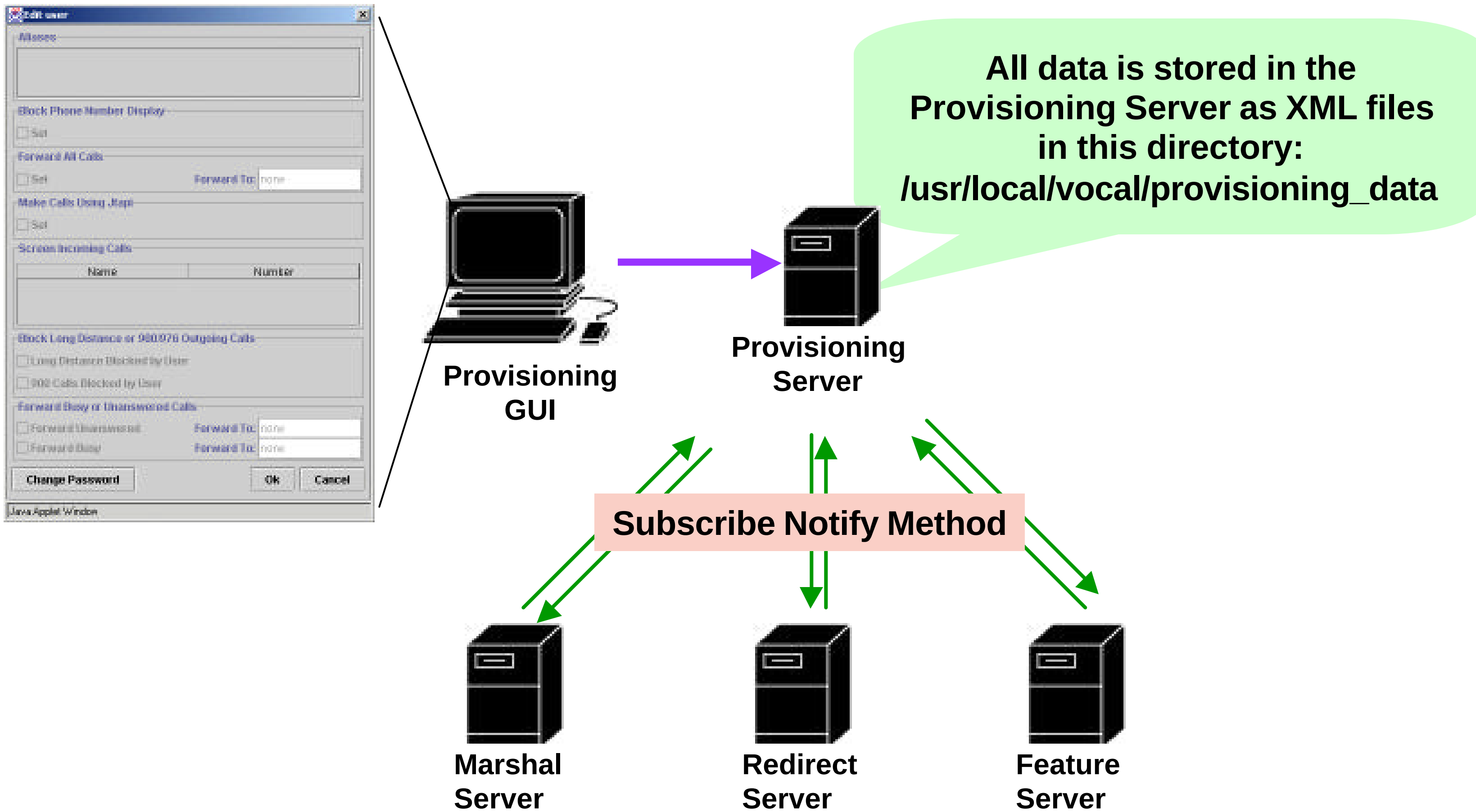
User Screen

The User's screen allows you set the user's features.

Note – the user's features must first be enabled by the administrator before a user can set it.



Provisioning Server - Data Storage



Redirect Server

Redirect Server

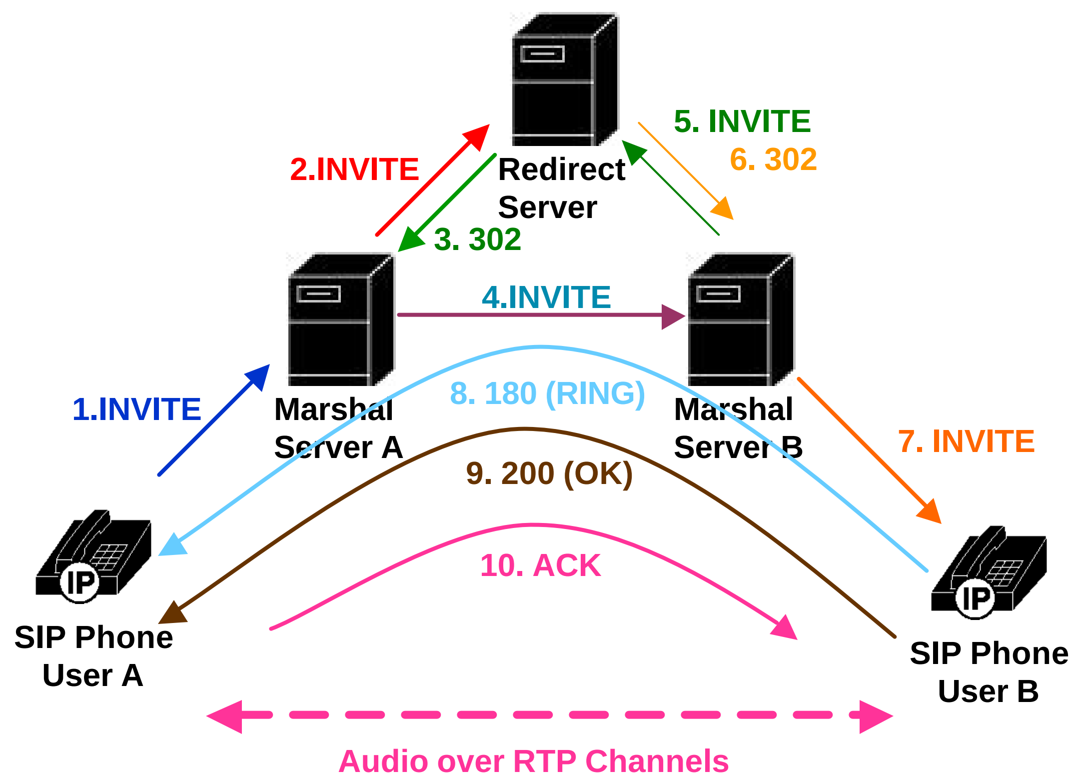
The Redirect Server provides these SIP services and functions:

- **Registration.**
- **Redirection.**
- **Location.**

The Redirect Server provides routing information to the Feature and Marshal Servers to route a call.

Review - A Basic Call involving Marshal and Redirect Servers

- Marshal Servers forwards INVITE messages to the Redirect Server to obtain routing information.
- The Redirect Server responds with a 302 message containing the routing information.



How the Redirect Server Determines Route

The Redirect Server determines route by:

1. **Retrieving a previously built subscriber list or the dial plan.**
2. **Building a contact list from information in the INVITE message.**
3. **Generating a 302 message with the routing information.**

When are the list built?

Subscriber List and Dial Plan:

- The subscriber list is built on startup and registration only.
- The dial plan is provisioned using the Provisioning GUI.

Contact List:

- The contact list is built on a per call basis, i.e. when the Redirect Server receives an INVITE message.

Redirect Server

Subscriber Lists

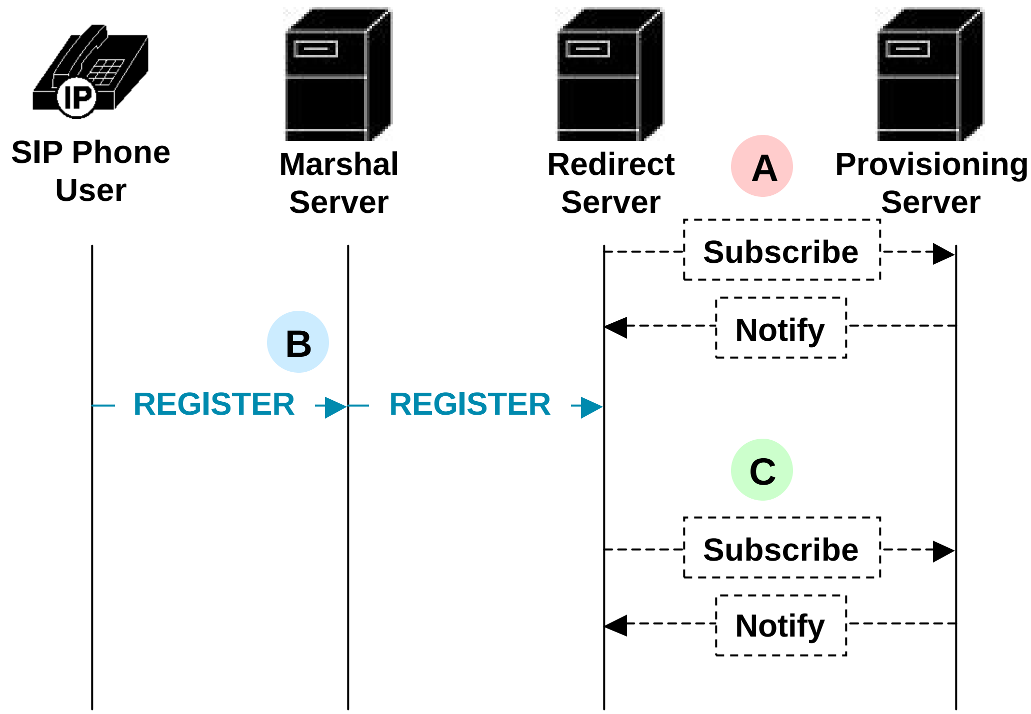
Building Subscriber Lists

To build a subscriber list the Redirect Server does three things:

A. On startup, collect user names from the Provisioning Server.

B. Looks at information in the REGISTER message.

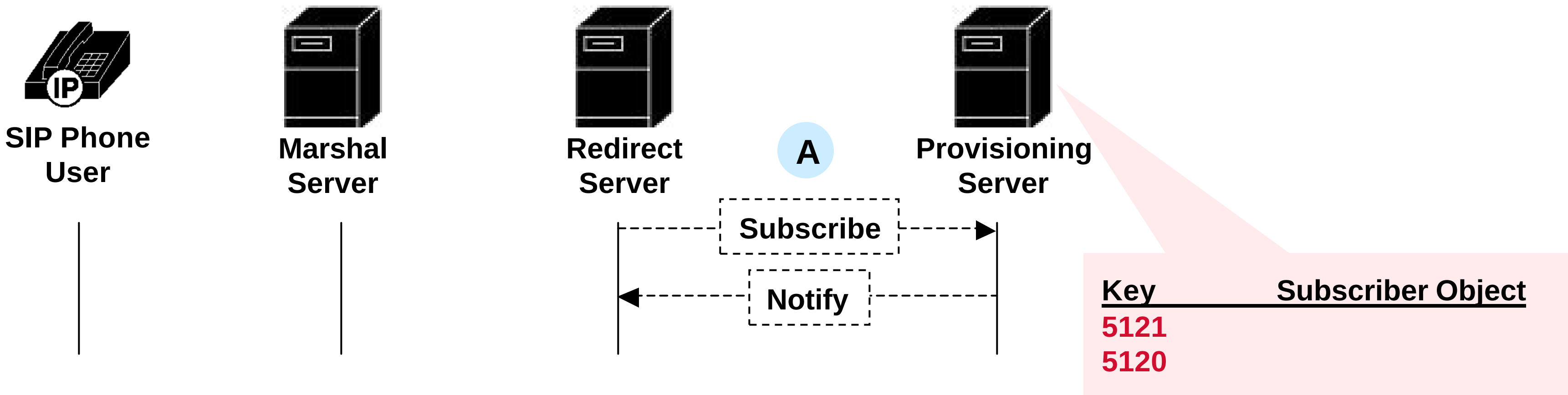
C. Collects feature and user data from the Provisioning Server.



Example: Subscriber List

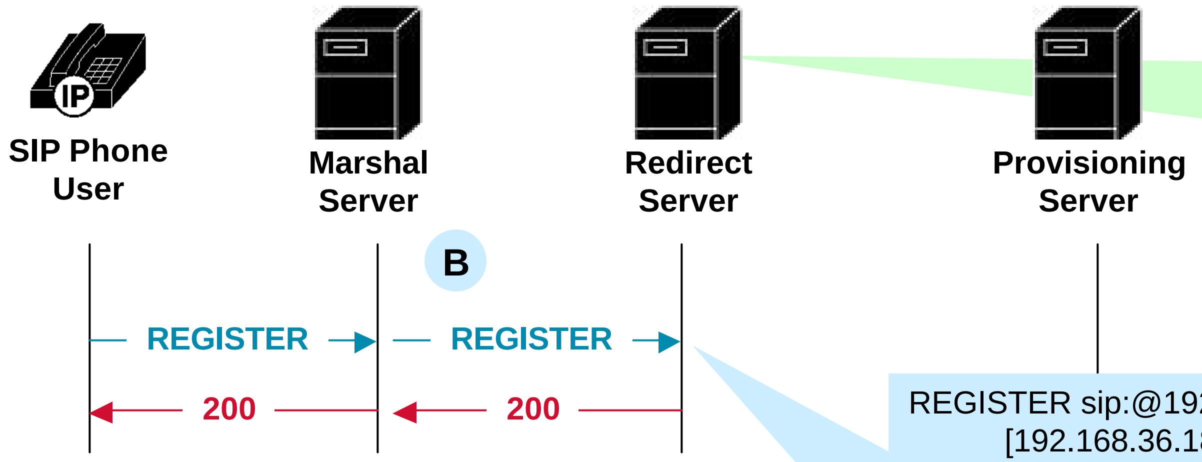
Key	Subscriber Object	
5121	Caller ID Blocking	← Called Contacts
	Call Forward No Answer	← Calling Contacts
	192.168.36.180	←Terminating Contacts
	192.168.36.21	←Terminating Contacts
	3600 milliseconds	← Expiry Time
5120	Call Blocking	←Called Contacts
	192.168.36.181	←Terminating Contacts
	192.168.36.20	← Terminating Contacts
	3600 milliseconds	← Expiry Time

Step A: On Startup – Collect User Names



1. On startup, the Redirect Server contacts the Provisioning Server for a list of user names (Subscribe).
2. The Provisioning Server sends user names (Notify).
3. The Redirect Server builds a subscriber list where:
   - User names are saved as keys.
   - Subscriber objects are left blank.

Step B – Getting Information from the REGISTER message



Key	Subscriber Object
5121	Terminating Contacts
	192.168.36.180
	192.168.36.21
	Expiry Time
	3600

REGISTER sip:@192.168.36.200:5060 SIP/2.0
[192.168.36.180:5060->192.168.36.200:5060]

Via: SIP/2.0/UDP 192.168.36.180:5060

Via: SIP/2.0/UDP 192.168.36.21:5060

From: <sip:5121@192.168.36.180:5060>

To: <sip:5121@192.168.36.180:5060>

Expires: 3600

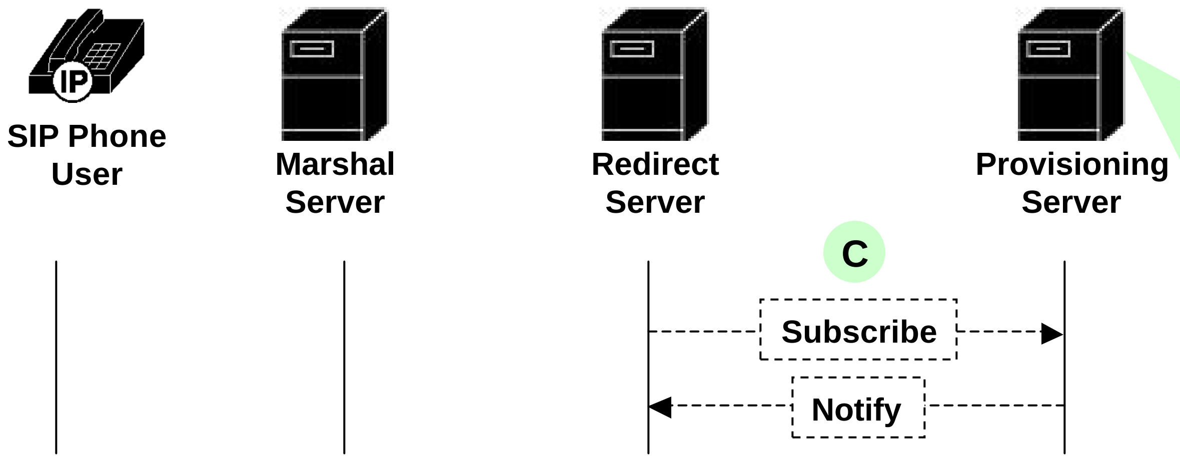
Contact: <sip:5121@192.168.36.180:5060>

Contact: <sip:5121@192.168.36.21:5060>

The Redirect Server:

- **Compares the From header field against keys in the subscriber list.**
- **Extracts Contact and Expiry header field.**
- **Updates the subscriber list.**

Step C- Getting User Information



Key	Subscriber Object
3001	**Called Contacts**
	Calling Contacts
	Terminating Contacts
	Expiry Time
3002	**Called Contacts**
	Calling Contacts
	Terminating Contacts
	Expiry Time

- **The Redirect Server contacts the Provisioning Server to obtain user's feature information.**
- **The user data is saved into the subscriber list as called contacts and calling contacts.**

VOVIDA.ORG

Redirect Server

Dial Plan

What is a Dial Plan?

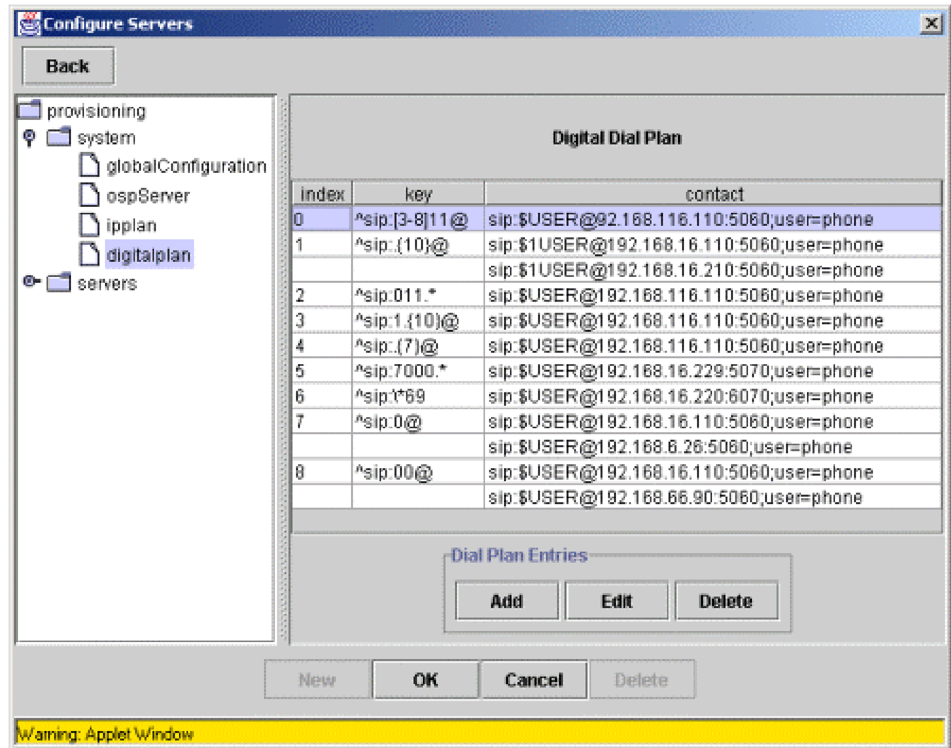
- **The dial plan consists of entries of keys and contacts.**
- **The dial plan is used if the caller's SIP URI does not match a key in the subscriber list.**

Key	Contact
^sip:1.{10}@	sip:$USER@192.168.116.110:5060;user=phone
^sip:*69	sip:$USER@192.168.116.220:5074;user=phone
	sip:$USER@server.com;user=ip

How do I create a Dial Plan?

From the Provisioning GUI you can build:

- **Digital Dial Plan – phone numbers (user=phone).**
- **IP Dial Plan – SIP URI addresses (user=IP).**



VOVIDA.ORG

Redirect Server

Contact List

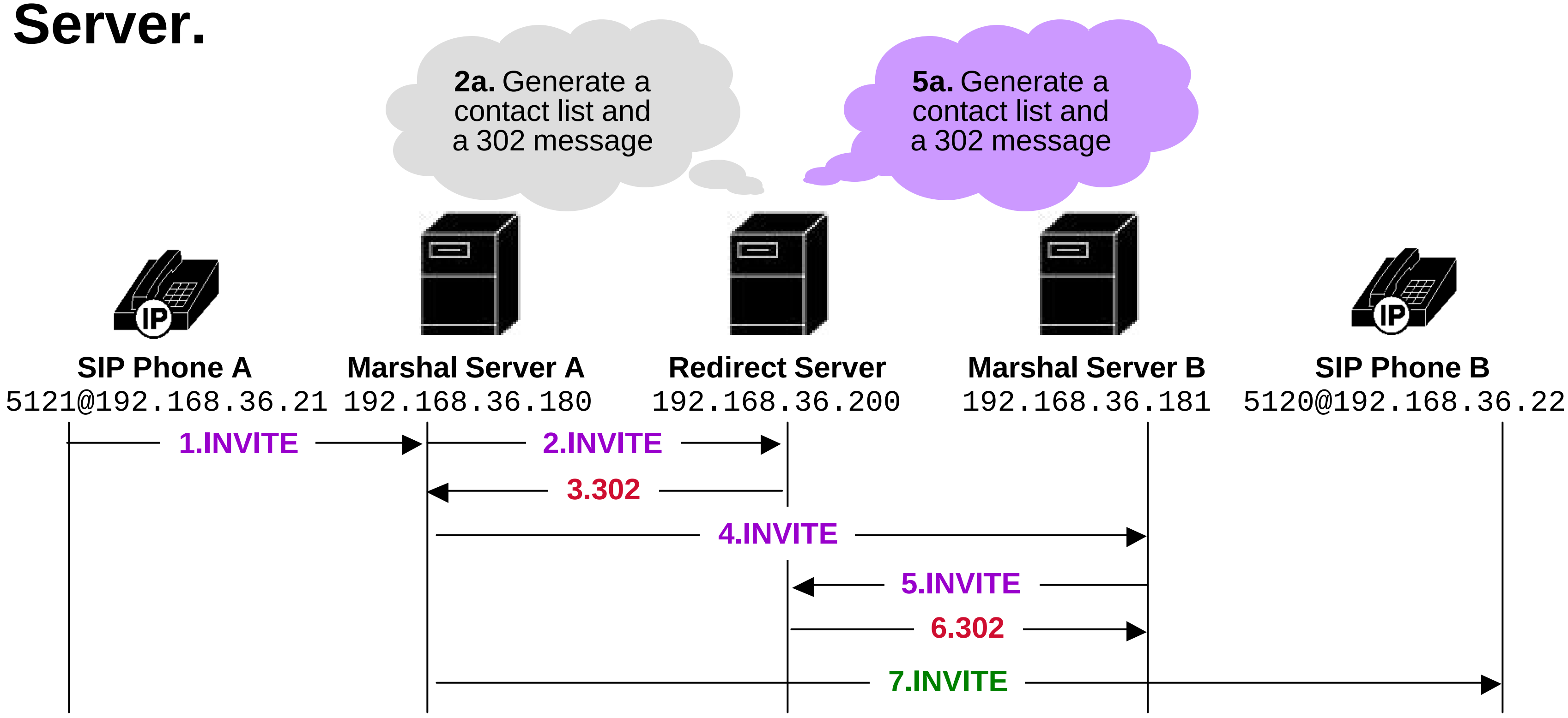
What is a Contact List?

The Contact List:

- **Is generated by the Redirect Server when it receives a INVITE message from a Marshal Server or Feature Server.**
- **Provides a list of called contacts, calling contacts, and terminating contacts.**
- **Is used to generate a 302 message in response to the INVITE.**
- **Is not saved and is valid for the duration that the Redirect Server needs to generate routing information.**

When does the Redirect Server generate a contact list?

Generated by the Redirect Server when it receives a INVITE message from a Marshal Server or Feature Server.



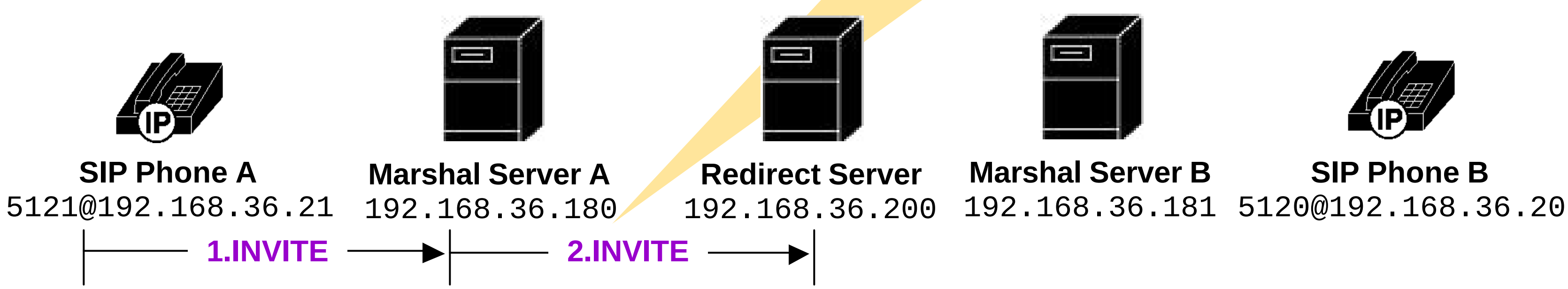
Extracting Information from the INVITE – FROM and REQUEST URI field

When the Redirect Server receives an INVITE message it extracts information from:

- **REQUEST URI field.**
- **FROM field.**

Using this information, the Redirect Server searches the subscriber list and builds a contact list.

INVITE sip:5120@192.168.36.200:5060
Via: SIP/2.0/UDP 192.168.36.180:5060;branch=1
Via: SIP/2.0/UDP 192.168.36.21:5060
From: sip:5121@192.168.36.21:5060
To: sip:5120@192.168.36.180
Expires: 180
Contact: sip:5121@192.168.36.21:5060



Building the Contact List

The Redirect Server builds a contact list:

- **FROM Calling Contacts.**
- **REQ URI Called Contacts.**
- **REQ URI Terminating Contacts.**

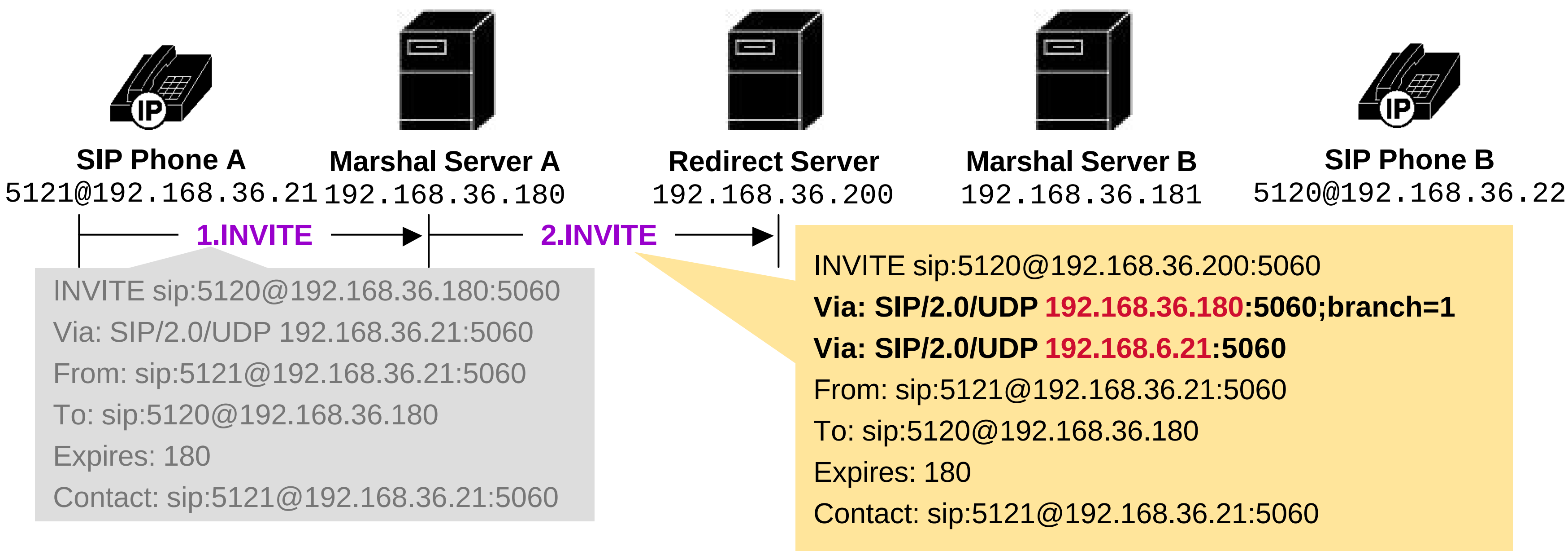
OR

- **FROM Calling Contacts.**
- **REQ URI Dial Plan Contact.**

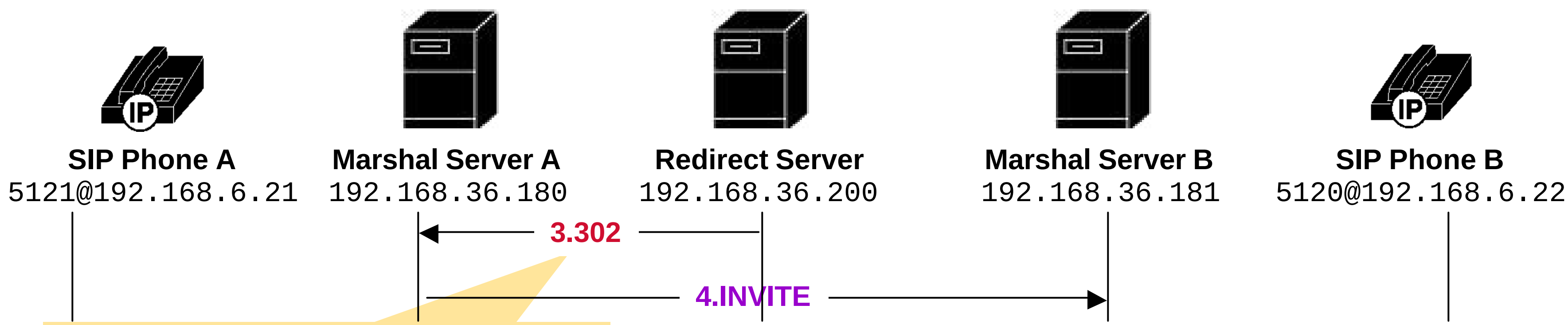
From this contact list, the Redirect Server determines a single contact to include within the 302 message.

INVITE Message – VIA Field

- **The Redirect Server looks at VIA field to determine where the call has been.**
- **The Redirect Server has a algorithm to determine which contact in the contact list to use.**



302 Message

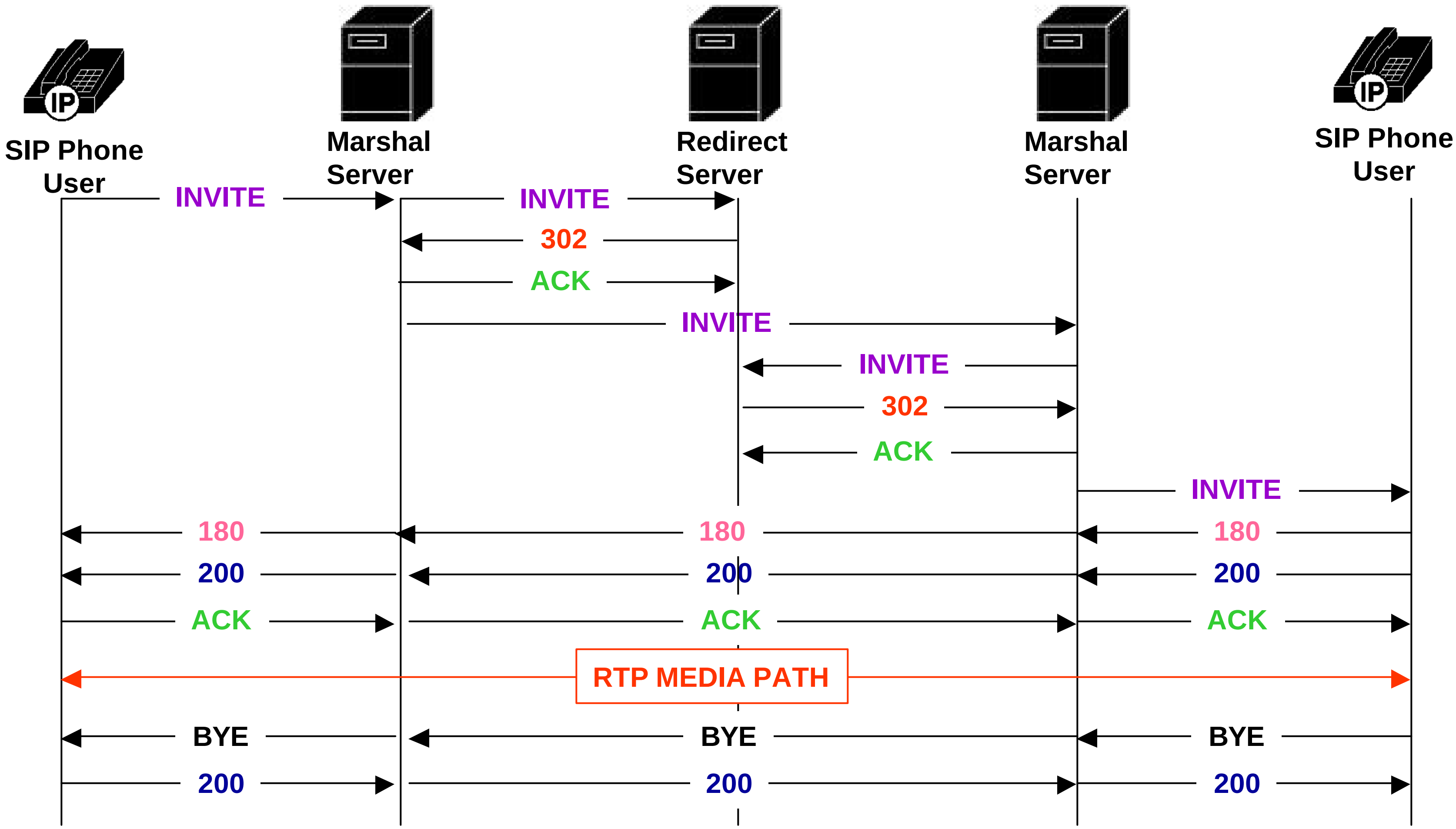


302 Moved Temporarily
Via: SIP/2.0/UDP 192.168.36.180:5060
Via: SIP/2.0/UDP 192.168.6.21:5060
From: sip:5121@192.168.6.21:5060
To: sip:5120@192.168.36.180:5060
Contact: sip:5120@192.168.36.181:5060

CONTACT: provides a new SIP URL where the user (5120) can be reached. In this case, user 5120 can be reached at 192.168.36.181.

INVITE sip:**5120@192.168.36.181**:5060;
Via: SIP/2.0/UDP 192.168.36.180:5060;
Via: SIP/2.0/UDP 192.168.6.21:5060
From: sip:5121@192.168.6.21:5060
To: sip:5120@192.168.36.180:5060
Expires: 180
Contact: sip:5121@192.168.6.21:5060

A Complete Call



Redirect Server

Redundancy

Redirect Server - Redundancy

- For redundancy multiple Redirect Servers are supported in the VOCAL system.
- The Redirect Server listens for and exchanges heartbeat messages with other Redirect Servers, Marshal Servers and Feature Servers.
- All Redirect Servers contain the same information.
- The Redirect Server shares registration information.

Process for Synchronizing a new Redirect Server

A new Redirect Server is synchronized in these steps:

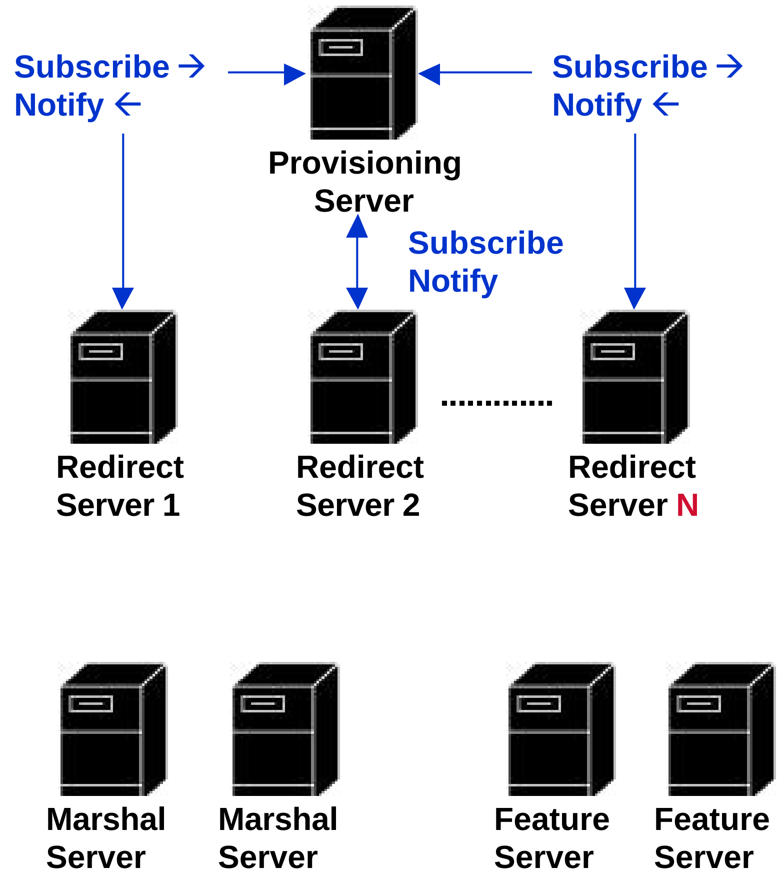
1. **Query the Provisioning Server.**
2. **Listen for heartbeat.**
3. **Synchronize with an active Redirect Server.**
4. **Send heartbeat.**

Obtaining Provisioning Information

On startup, a new Redirect Server queries the Provisioning Server for:

- Provisioned user to generate a subscriber list.
- List of all other Redirect, Marshal and Feature Servers.

When a user or server is added or deleted from the Provisioning Server, this information is sent to all Redirect Servers.



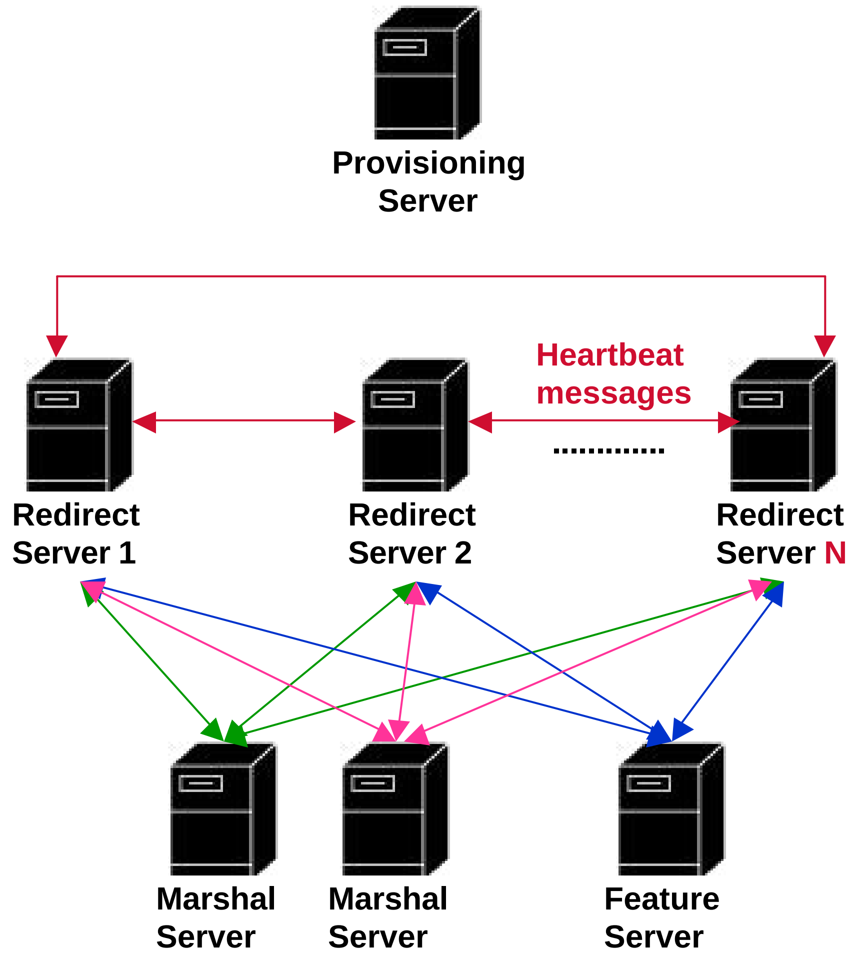
Listening for Heartbeat

The new Redirect Server then listens on the multicast address/port for heartbeats from:

- Other Redirect Servers.
- Marshal Servers.
- Feature Servers.

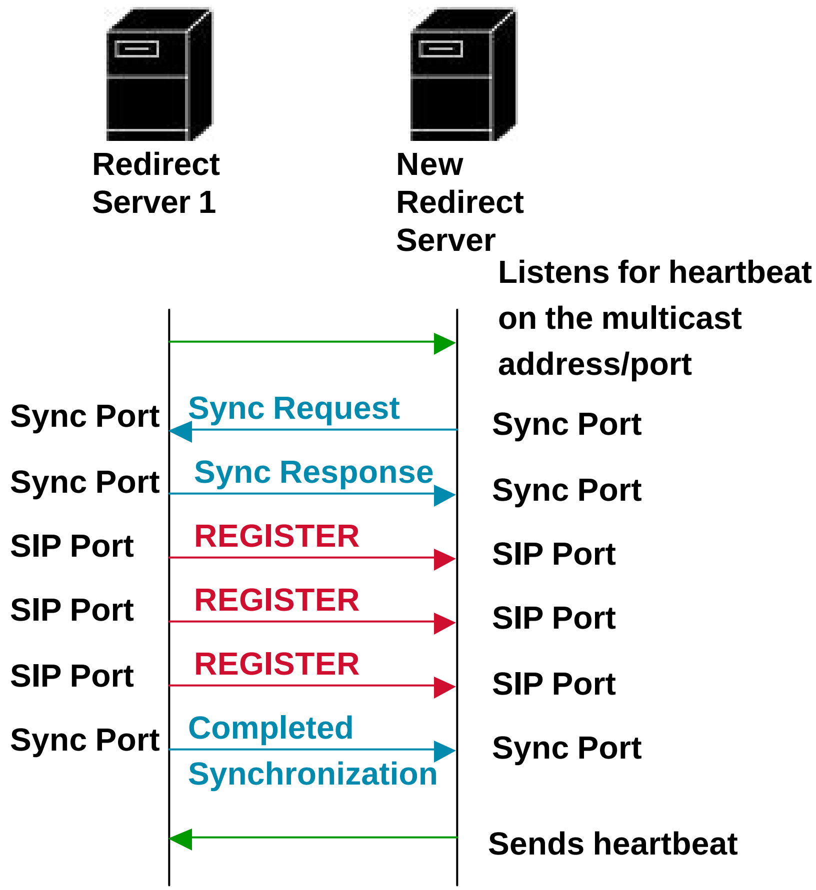
Each Redirect Server keeps track of:

- Status of the server (inactive/active).
- Number of heartbeat received.
- Number of missed heartbeats.



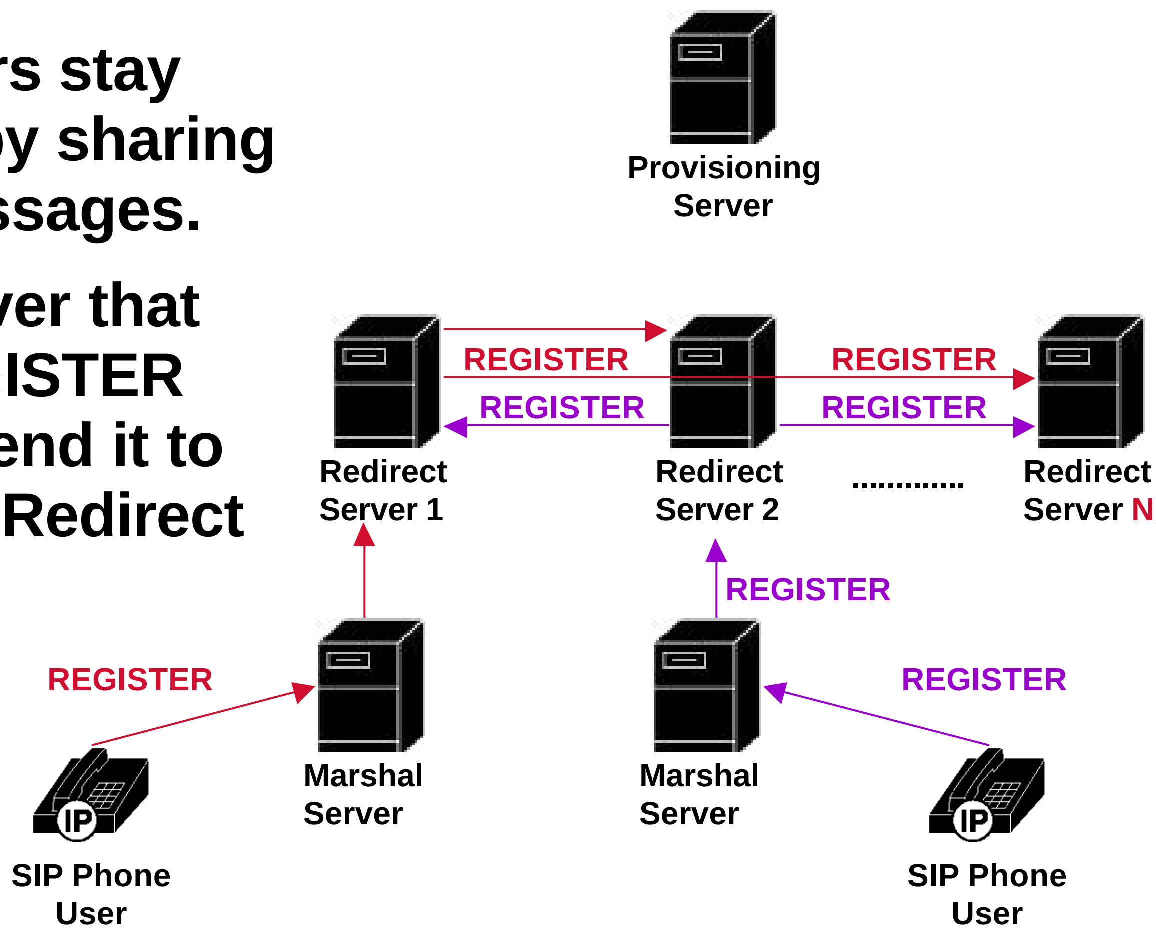
Synchronizing with an Active Redirect Server

- The new Redirect Server selects an active Redirect Server and sends a Sync Request on the Sync Port.
- The active Redirect Server responds with a Sync Response.
- The active Redirect Server generates REGISTER messages for each registered user in its subscriber list.
- The active Redirect Server notifies the new Redirect Server when it completes synchronization.
- The new Redirect Server begins sending heartbeats.



Mirroring the REGISTER Message

- Redirect Servers stay synchronized by sharing REGISTER messages.
- A Redirect Server that receives a REGISTER message will send it to all other active Redirect Servers.



Ports Used by the Redirect Server

Redirect Servers:

- Listen for heartbeats on the multicast address and port.
- Send sync request and sync response on the sync port.
- Forward REGISTER messages on the SIP port.

VOVIDA.ORG

Feature Server

Features

The VOCAL system supports these features:

Core network features provided by the Feature Server:

- Call Forward All.
- Call Forward No Answer.
- Call Forward Busy.
- Call Blocking.
- Call Return.
- Call Screen.
- Caller ID Blocking.

Set based features provided by the phone or device:

- Transfer.
- Calling Name Delivery.
- Calling Number Delivery.
- Call Waiting.
- Conferencing.

Calling and Called Features

Features can also grouped in calling and called features:

Calling Features.

- Calling Number Delivery.
- Calling Name Delivery.
- Caller ID Blocking.
- Call Blocking.

Called Features.

- Call Forward All Calls.
- Call Forward No Answer.
- Call Forward Busy.
- Call Screening.

Provisioning Features

- Features are enabled and set using the Provisioning GUI.
- The Provisioning Server generates a CPL script for each user’s features.
- The CPL scripts are saved into the /usr/local/vocal/provisioning_data directory.

How are Features Implemented?

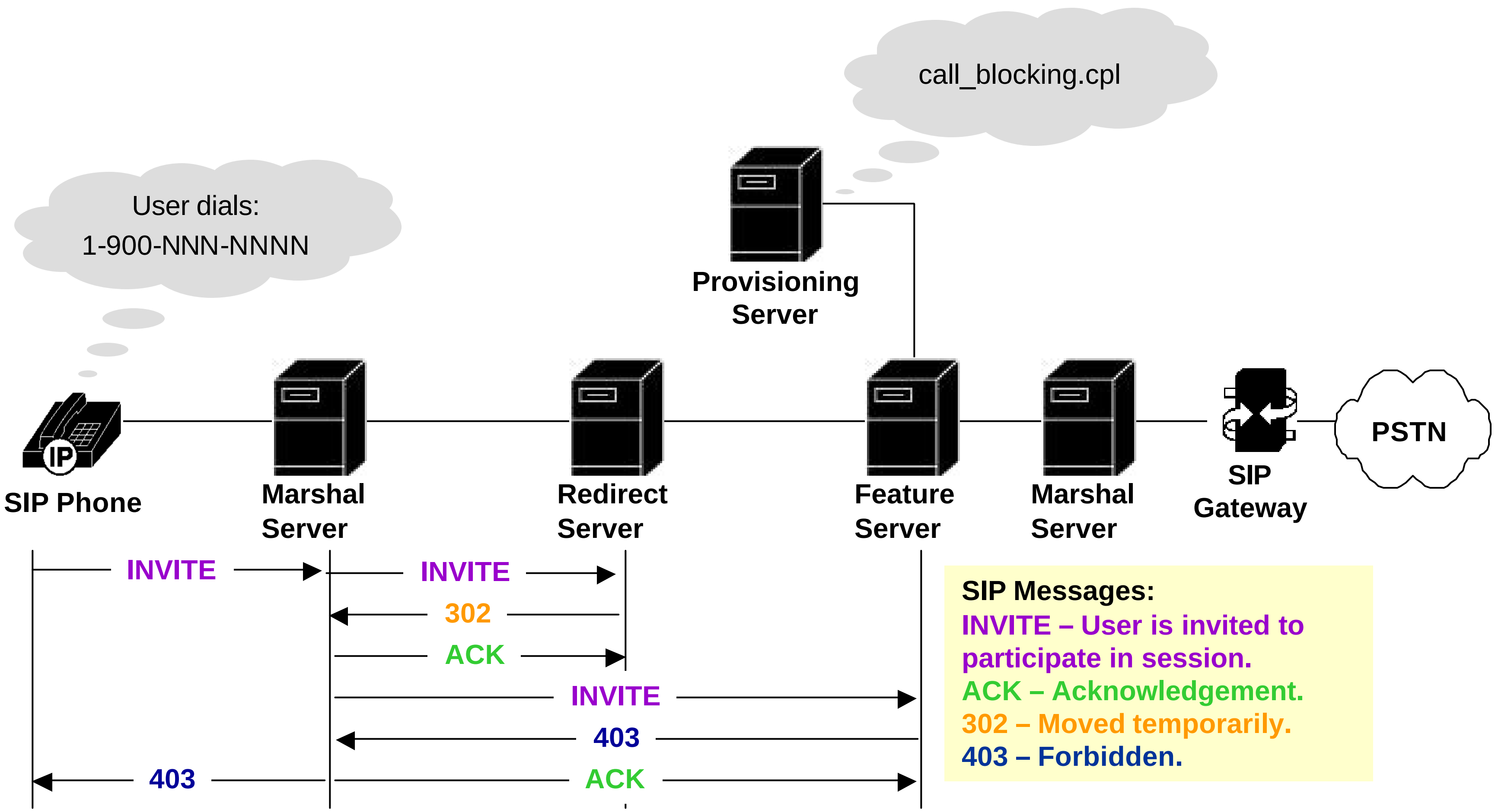
On startup, a Feature Server:

- **Queries the Provisioning Server.**
- **Interprets the CPL scripts and processes the CPL scripts into an executable.**

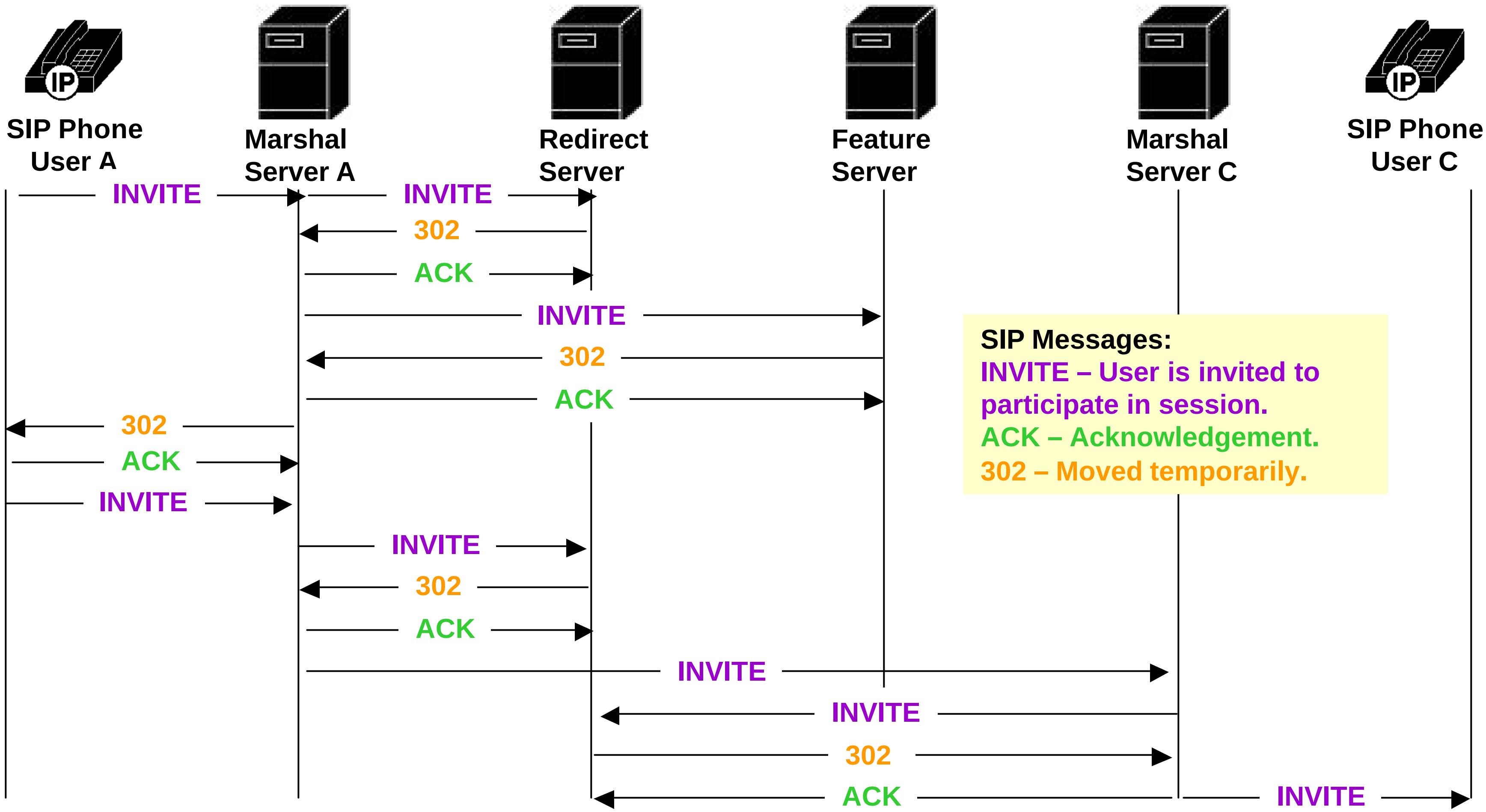
The executable is:

- **Saved in cache until needed.**
- **Triggered by a SIP event, such as an INVITE message, arriving at the Feature Server.**

Basic Call with Features – Call Blocking

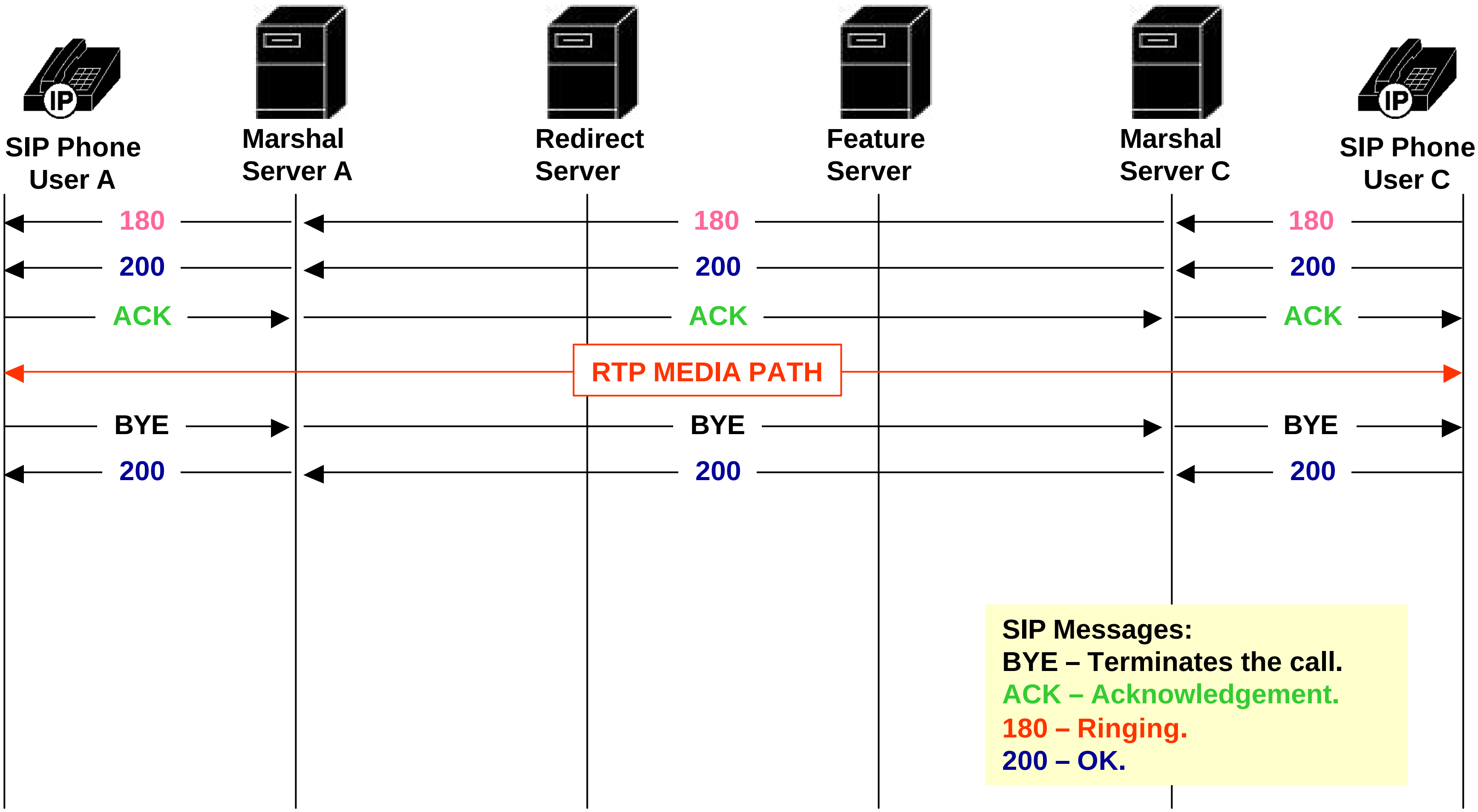


Basic Call with Features – Forward All Calls



SIP Messages:
INVITE – User is invited to participate in session.
ACK – Acknowledgement.
302 – Moved temporarily.

Basic Call with Features – Forward All Calls (continued)



SIP Messages:
BYE – Terminates the call.
ACK – Acknowledgement.
180 – Ringing.
200 – OK.

Call Processing Language-CPL

Call Processing Language (CPL) is:

- **XML based.**
- **Describes Internet telephony services and creating end-user service features.**
- **A lightweight scripting language - it has no variables, loops or ability to run external programs.**
- **Makes decisions based on call properties such as time of day, calling party, called party and priority and then apply an action such as, forwarding a call, blocking a call, redirecting a call, sending emails.**
- **Currently an IETF draft.**

VOVIDA.ORG

JTAPI Feature Server

JTAPI

Java Telephony API (JTAPI) is:

- **Used for telephony call control, physical device control, media services, and administrative services.**
- **http://java.sun.com/products/jtapi/**

JTAPI Packages

The JTAPI specifications defines fives packages:

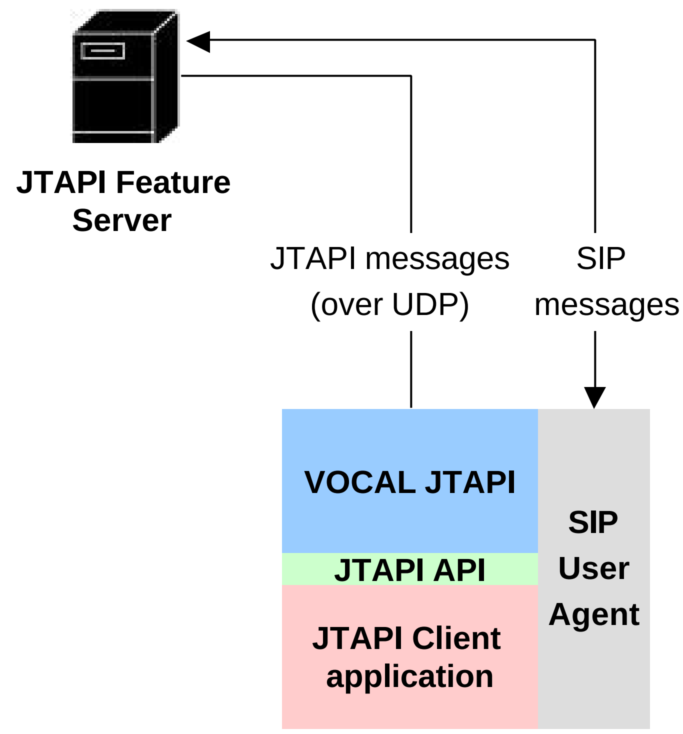
- **Core – support call setup and termination.**
- **Call Control – supports call transfer, conferencing, and hold.**
- **Call Center – supports call center applications.**
- **Media – supports applications that access the media channel of a call (i.e.. DTMF tones).**
- **Phone – supports applications that control physical features of a hardware telephone set.**

VOCAL implements the Core package only.

VOCAL JTAPI Implementation

The current VOCAL JTAPI implementation requires:

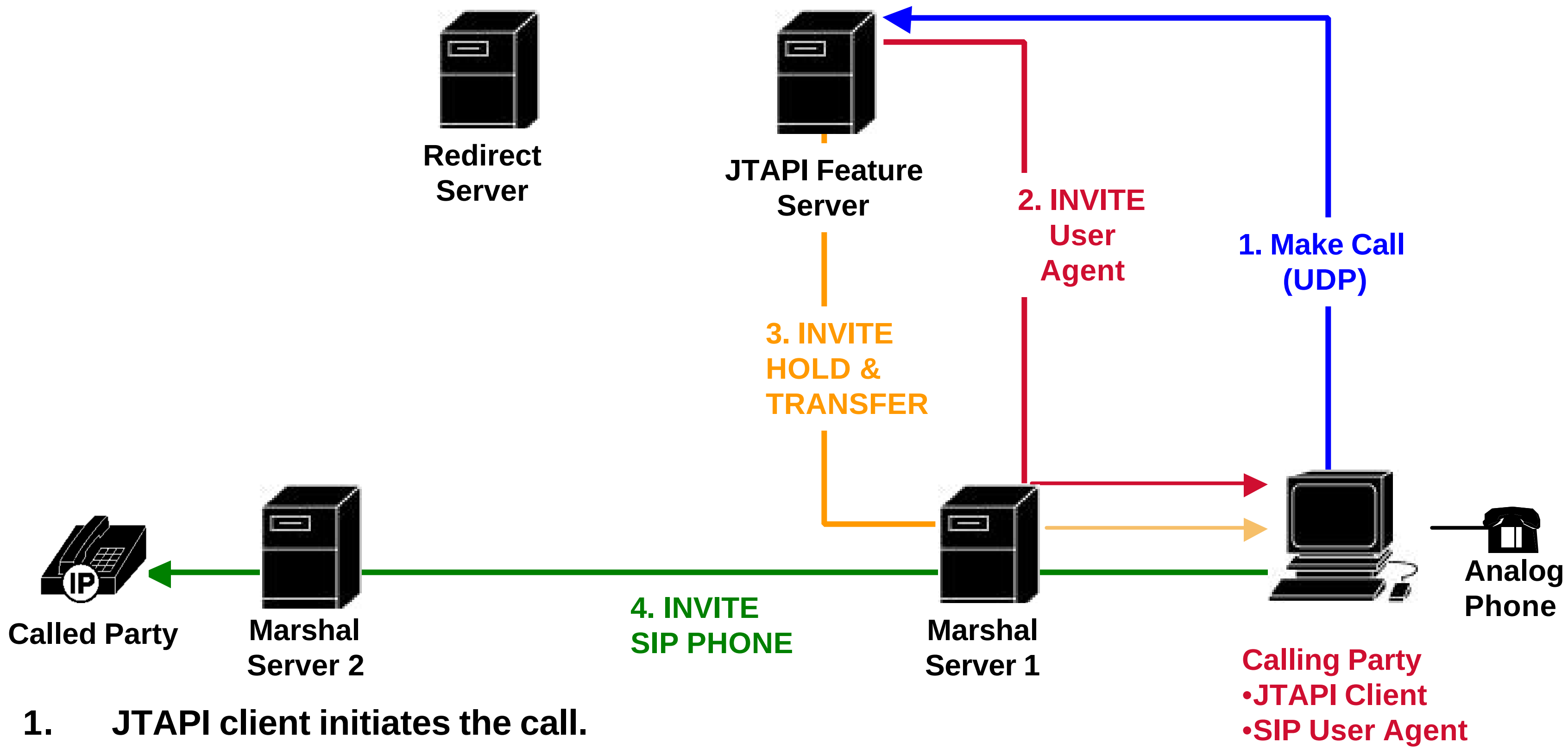
- **JTAPI client.**
- **SIP User Agent.**
- **JTAPI Feature Server.**



Implementation Issues

- **ALL endpoints must support SIP TRANSFER / REFER message.**
  - **This message is defined in a SIP draft proposal.**
- **The current VOCAL JTAPI implementation does not support redundancy.**

Simplified JTAPI and SIP Call Flow



1. JTAPI client initiates the call.
2. JTAPI Feature Server calls the SIP User Agent. (INVITE).
3. JTAPI Feature Server places the SIP User Agent on HOLD and sends a TRANSFER.
4. SIP user agent calls the SIP phone 2.

Detailed Call Flows and Call Scenarios

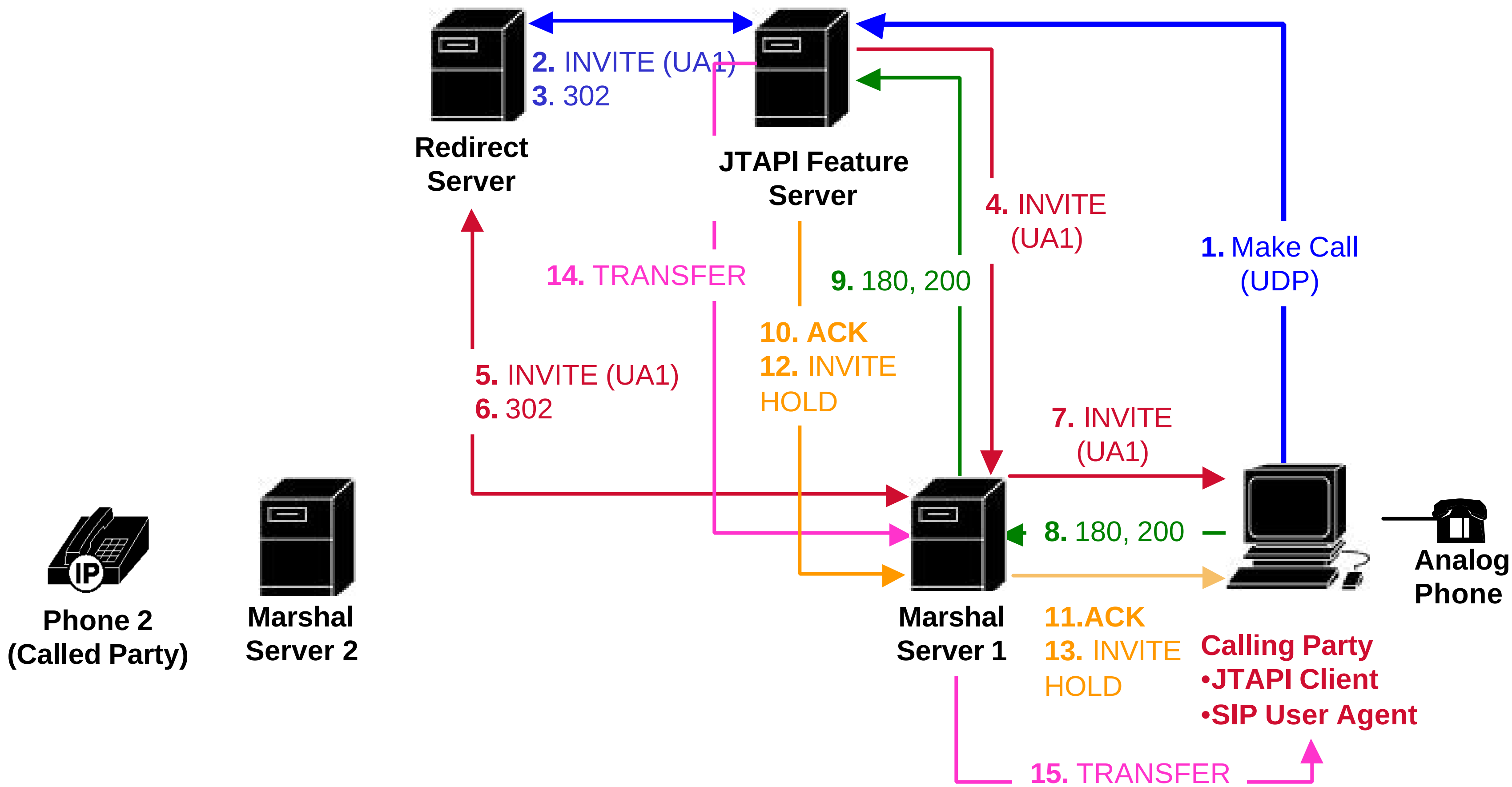
The next few slides will describe detailed call flows in two parts:

1. **Call initiation from a JTAPI client.**
2. **Call setup:**
   a. **To a SIP Phone.**

      OR.

   b. **To a JTAPI Client.**

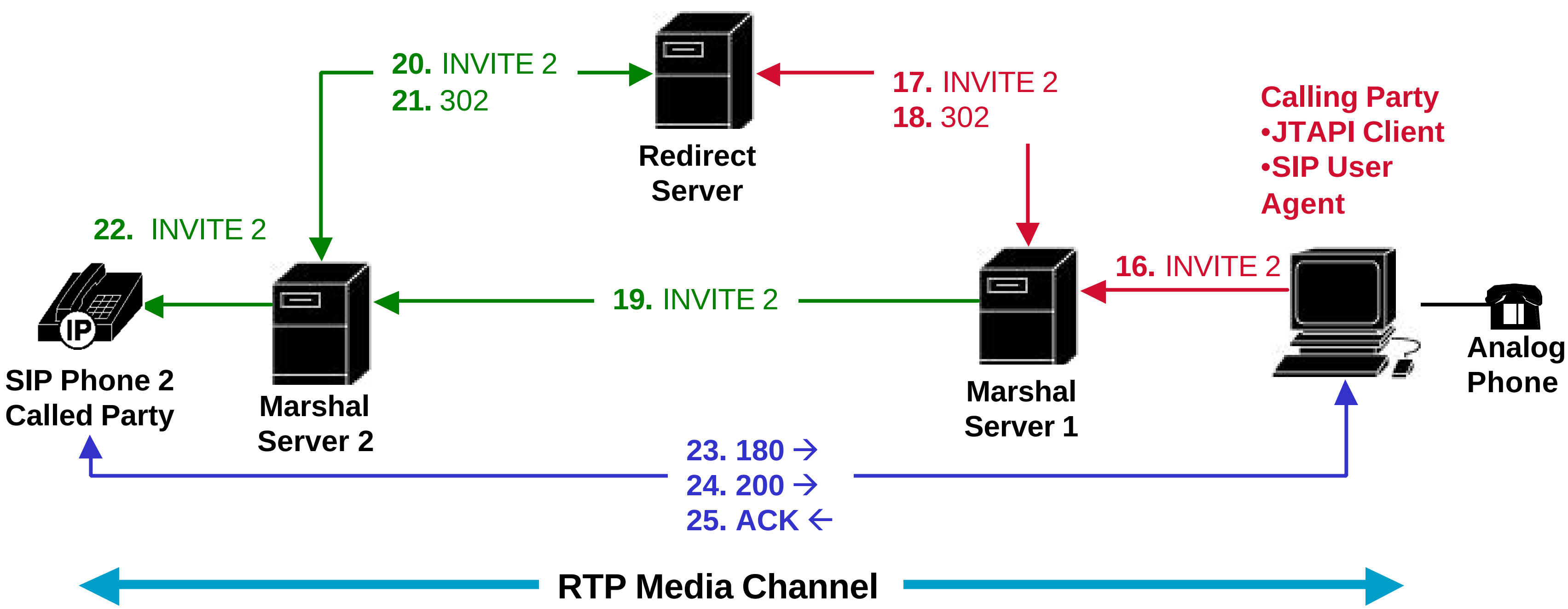
Detailed Call Flow
JTAPI Client Call Initiation (Part 1)



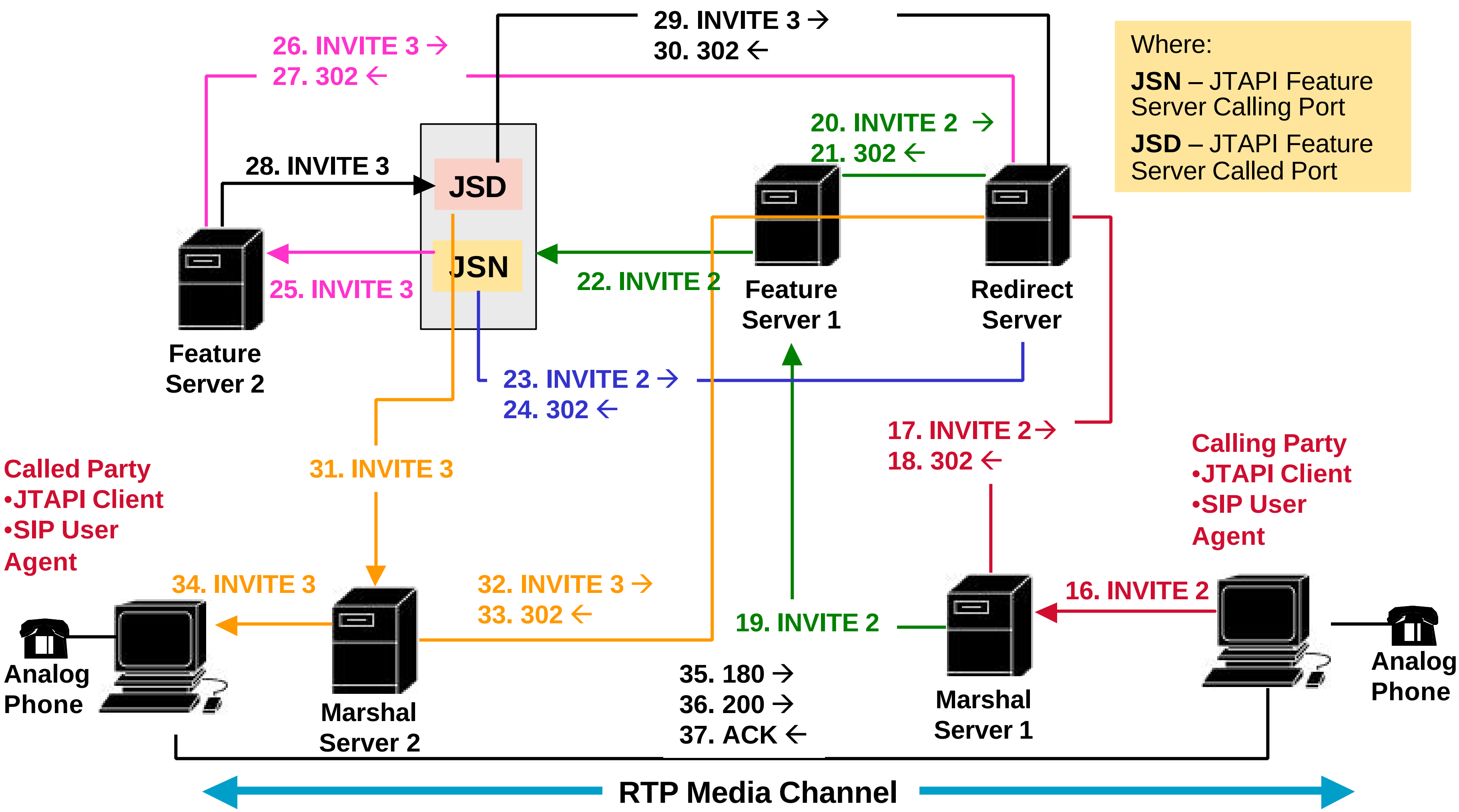
Special Messages – Media Channel on Hold

- 180 and 200 messages have been exchanged as in a normal call flow. JTAPI Server sends a ACK.
- JTAPI Server sends special INVITE to Called Party to place the media channel on hold.
- JTAPI Server sends a TRANSFER/REFER message to indicate the User Agent to call the called party's number.
- The User Agent sends a new INVITE to the called party.

Detailed Call Flow
JTAPI Client to SIP Phone (Part 2a)



Detailed Call Flow
JTAPI Client to JTAPI Client (Part 2b)

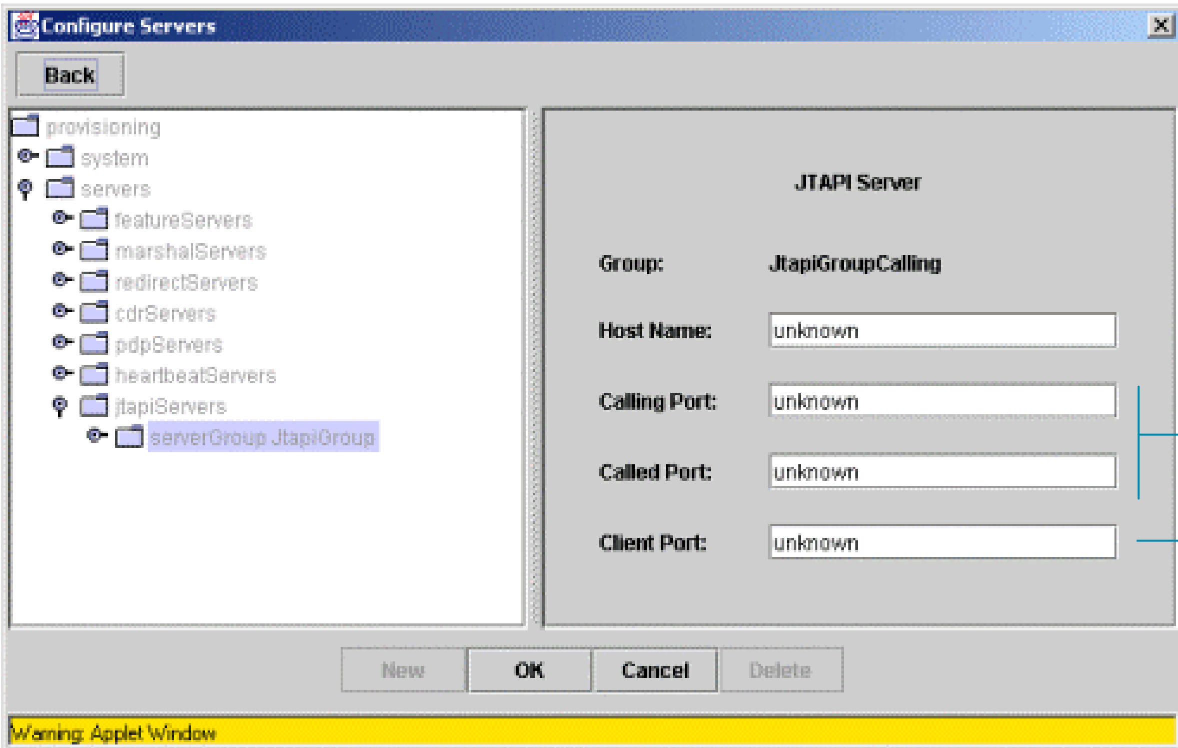


Where:

JSN – JTAPI Feature Server Calling Port

JSD – JTAPI Feature Server Called Port

Provisioning the JTAPI Server



Ports on which the JTAPI server receives and sends SIP messages

UDP port used to communicate with the JTAPI client.

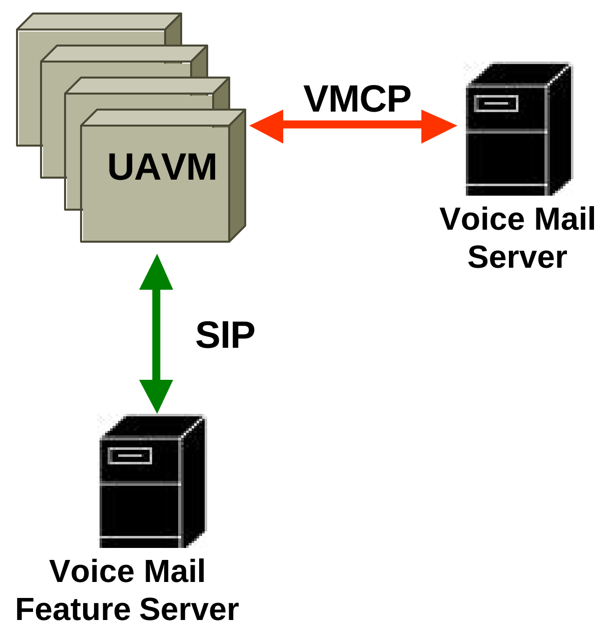
VOVIDA.ORG

Voice Mail Server

Voice Mail

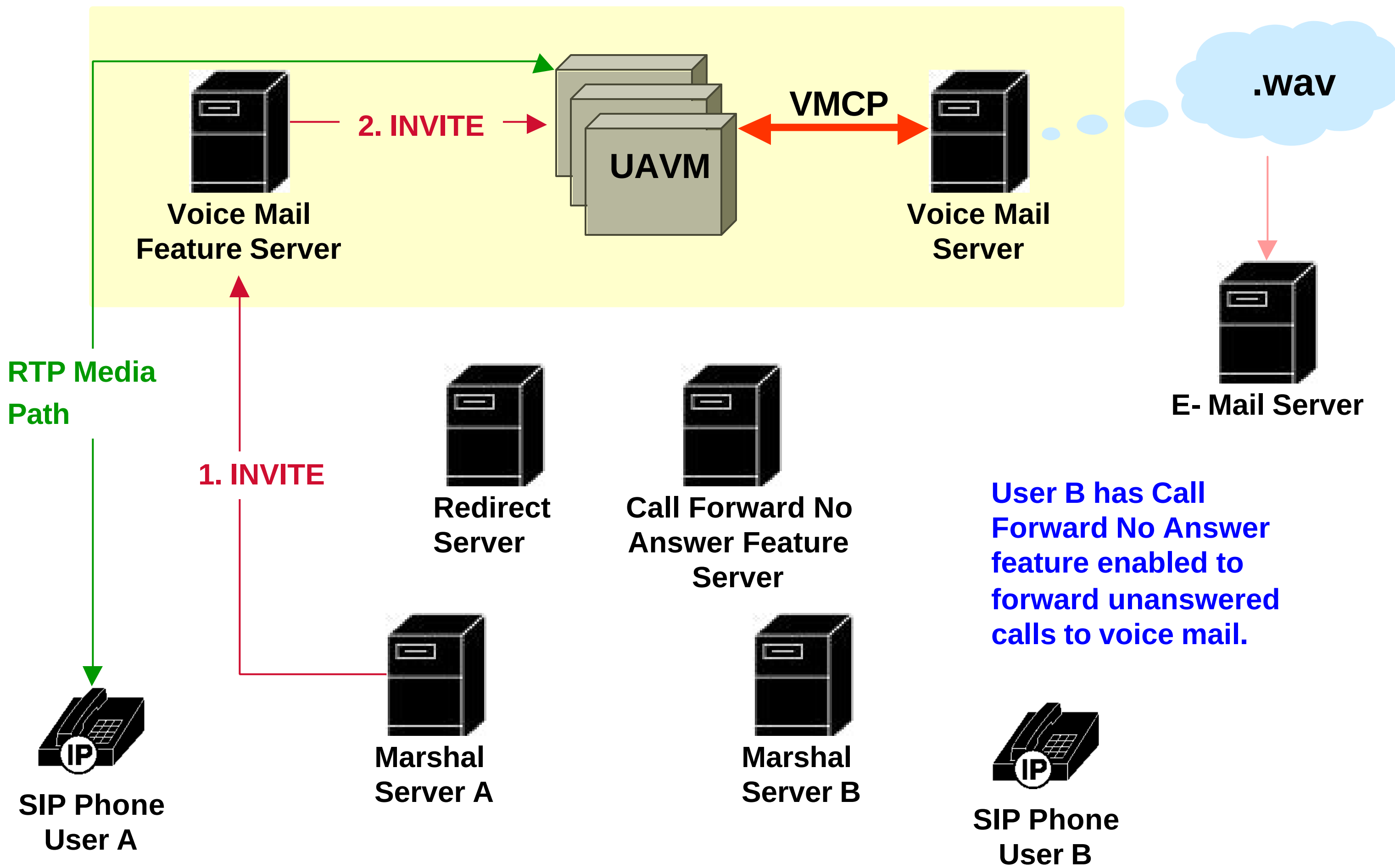
The VOCAL Voice Mail feature provides unified messaging using these logical functions:

- Voice Mail Feature Server.
- Voice Mail User Agent.
- Voice Mail Server.



VMCP- Voice Mail Control Protocol

Voice Mail Interactions



Voice Mail Feature Server

The Voice Mail Feature Server:

- **Distributes calls to available Voice Mail User Agents (UAVM).**
- **Listens for heartbeats from UAVM to know which UAVM is active and available.**
- **Forwards INVITE message to first available UAVM.**

Voice Mail User Agent

The Voice Mail User Agent (called UAVM):

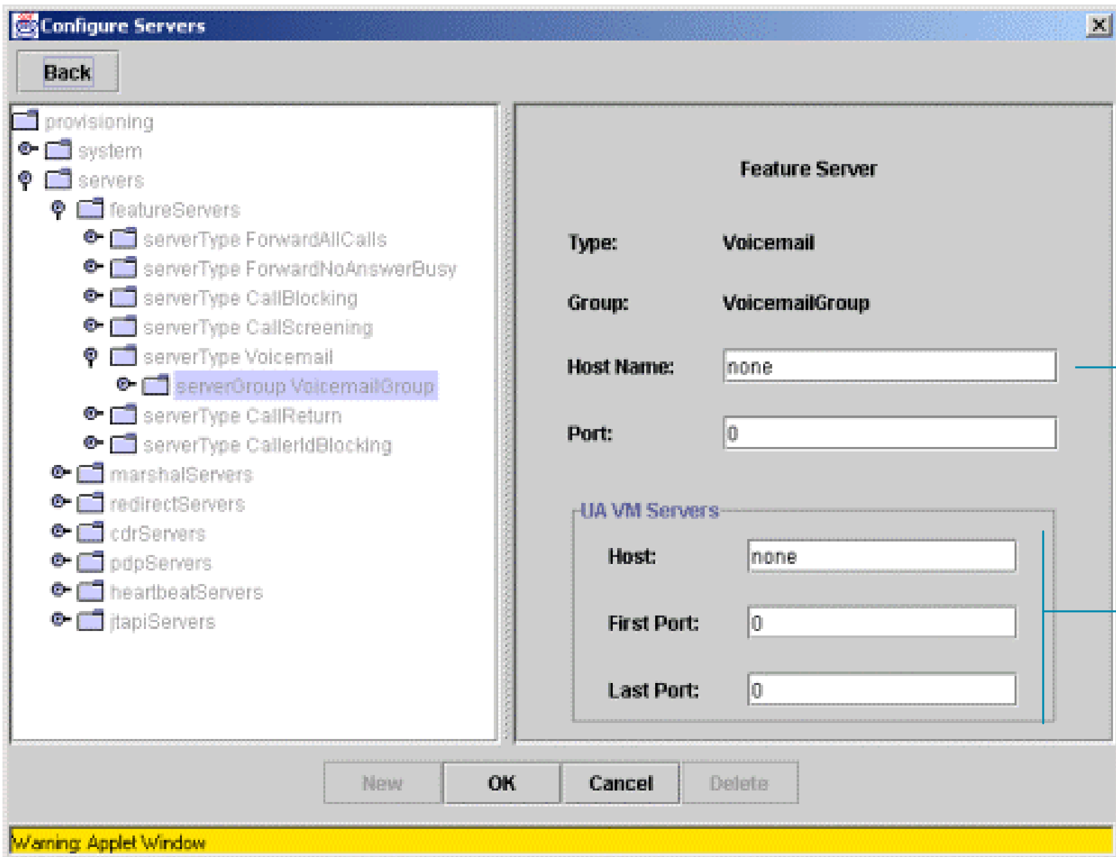
- **Acts as a gateway – it translates SIP and VMCP messages.**
- **Communicates with the Voice Mail Server using Voice Mail Control Protocol (VMCP) – a proprietary protocol.**
- **Plays greeting messages to caller over RTP path.**
- **When a Voice Mail Feature Server receives an INVITE message it forwards the message to the first available UAVM.**
- **The number of UAVM can be configured using the Provisioning GUI – you specify a range of SIP ports.**
- **The UAVMs sends heartbeat messages to indicate status.**
- **Each UAVM supports one call at a time.**

Voice Mail Server

The Voice Mail Server is used to:

- **Play recorded messages.**
- **Save voice messages as .wav files into a temp directory.**
- **Send .wav files as email attachments to a pre-configured email address. The email address is specified in the configuration file for each user.**
- **The UAVMs act as front end into the Voice Mail Server.**

Provisioning Voice Mail Feature Server



IP address of the host machine running the Voice Mail Feature Server

Port on which the Voice Mail Feature Server receives SIP messages

IP address of host machine running UAVMs. SIP port numbers on which SIP messages are sent and received.

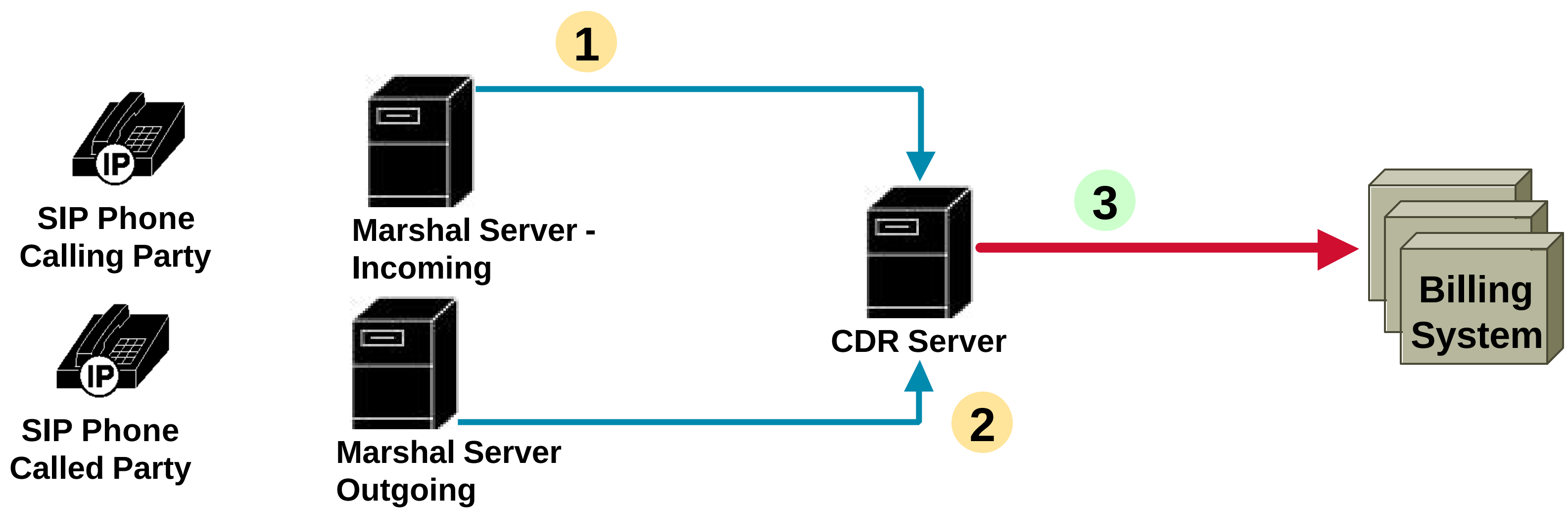
VOVIDA.ORG

Call Detail Record Server

Call Detail Record (CDR) Server

The Call Detail Record (CDR) Server:

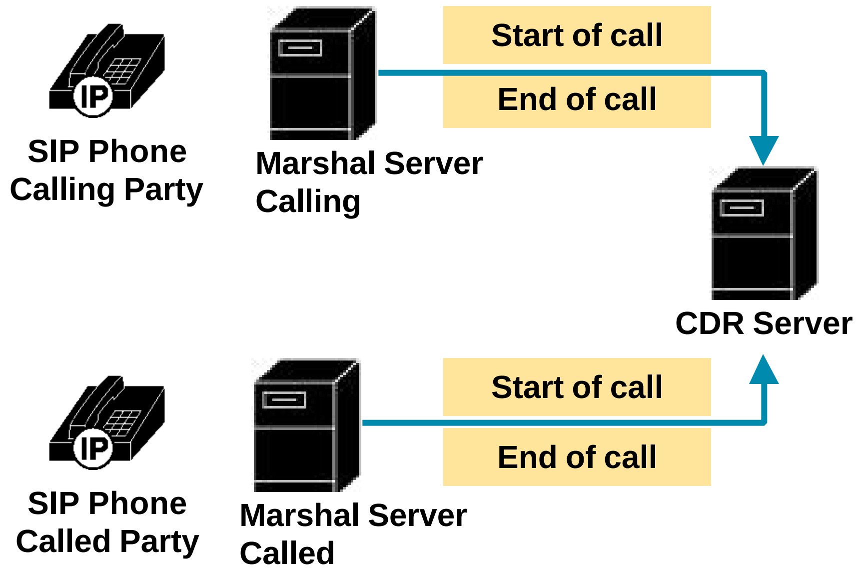
1. **Receives start and end times from Marshal Servers.**
2. **Formats data into CDR data for each call.**
3. **Forwards CDR data to 3rd party billing system using RADIUS accounting protocol.**



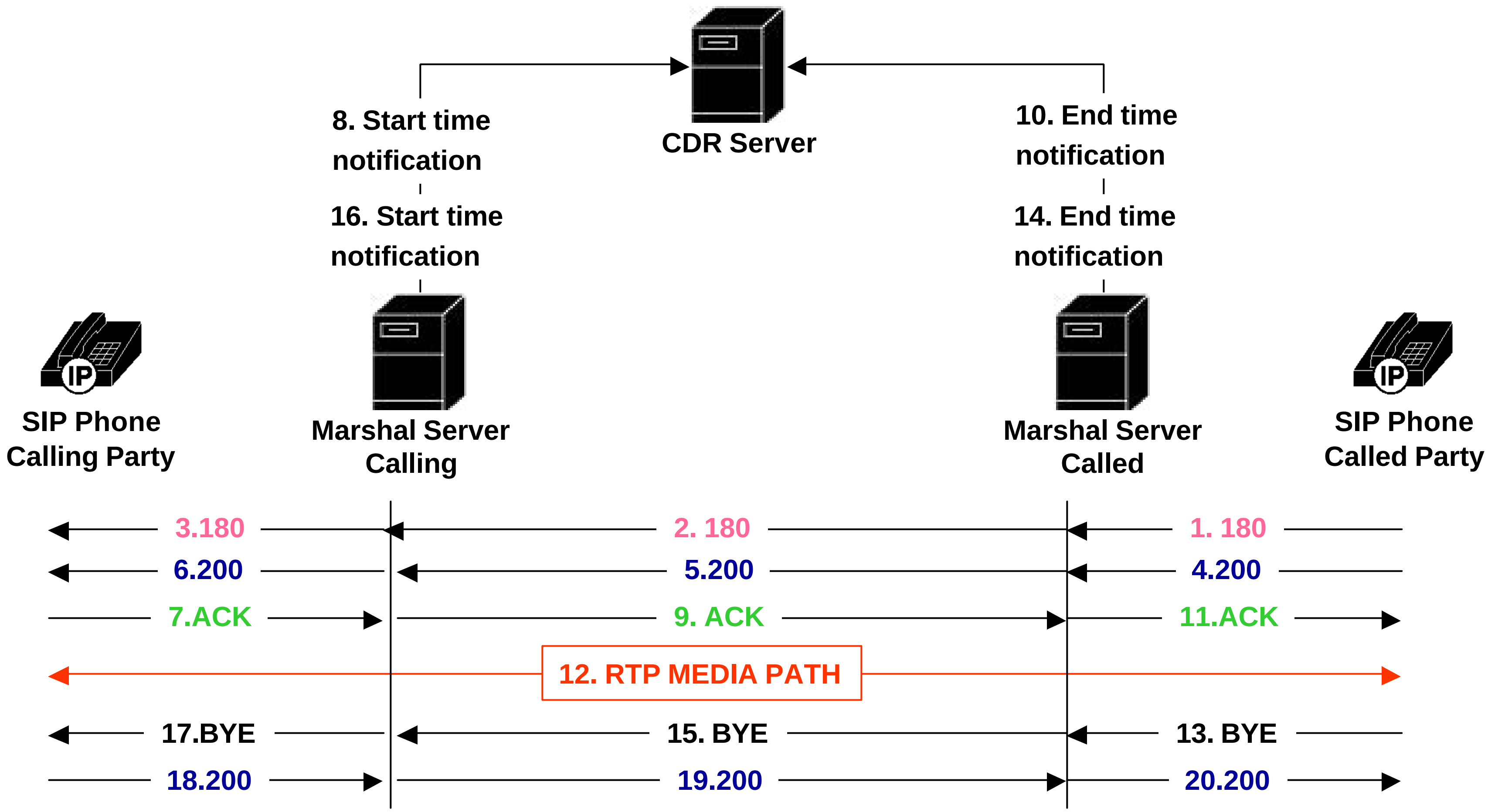
Step 1 –Data from Marshal Server

Both Marshal Servers send start and end times to the CDR Server:

- **Start - when the Marshal Servers receives an ACK message.**
- **Ring time (optional) – when the Marshal Servers receives a 180 message.**
- **End – when the Marshal Servers receive a BYE message.**



Start and Stop Time



Step 2 – Creating the Billing.dat File

The CDR Server saves records into the billing.dat file:

- Two start records.
- Two end records.
- Computed call duration record.

The CDR Server maintains a directory containing:

- billing.dat.
- billing.dat.timestamp.unsent.
- billing.dat.timestamp.

Example Billing.Dat File

The billing.dat file contains comma delimited fields that provide information including:

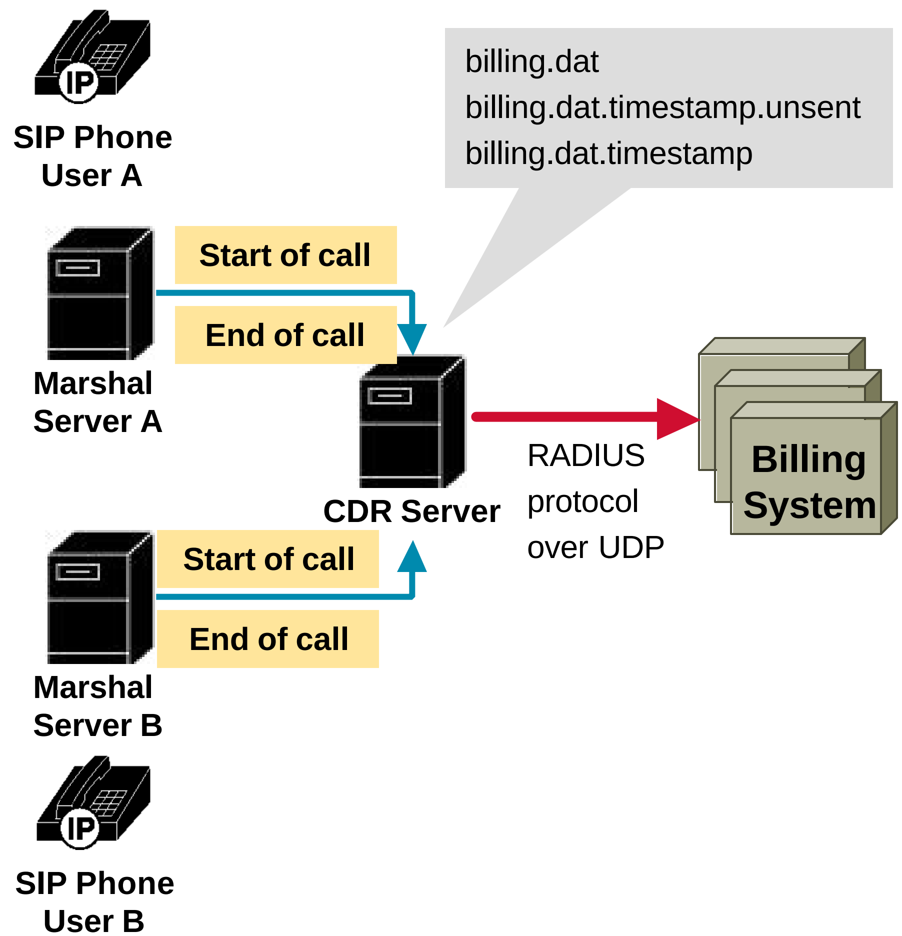
- **Start and stop of call.**
- **Call duration.**
- **Originator IP address.**
- **Call Type.**

```
CALL_RING,1,6383,,,6388,01/01/1970,00:00:00,0,0,01/01/1970,00:00:00,0,0,968972268,169,000:00:00,0,0,192.168.5.25,0,192.168.5.25,0,V,1,E,I

CALL_START,1,6383,,,6388,09/14/2000,22:57:49,968972269,174,01/01/1970,00:00:00,0,0,0,0,000:00:00,0,0,192.168.5.25,0,192.168.5.25,0,V,1,E,I

CALL_END,1,6383,,,6388,01/01/1970,00:00:00,0,0,09/14/2000,22:57:51,968972271,182,0,0,000:00:00,0,0,,0,,0,

CALL_BILL,1,6383,6383,,6388,09/14/2000,22:57:49,968972269,174,09/14/2000,22:57:51,968972271,182,0,0,000:02:08,2,8,192.168.5.25,0,192.168.5.25,0,V,1,N,I
```

Step 3 – CDR Server Forwards CDR Data to Billing System

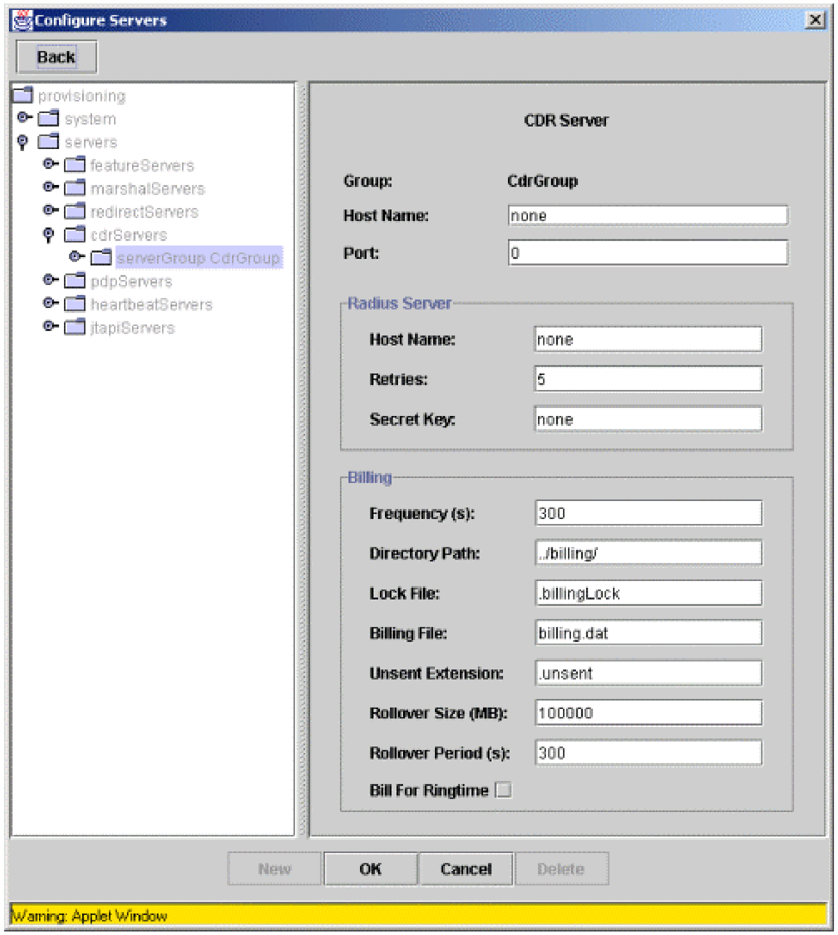
The CDR Server:

- Reads the record from the billing.dat. timestamp.unsent file.
- Sends the record to the billing system at a defined time interval.
- Communicates with the billing system using the RADIUS accounting protocol.



Provisioning the CDR Server

- **Frequency (seconds) – The frequency in seconds that the CDR Server sends records to the billing system.**
- **Rollover Size (MB) – Maximum file size of a billing file before it is rolled over.**
- **Rollover Period (seconds) – maximum age of a billing file before it is rolled over.**
- **Bill for Ring time – option to collecting billing information when 180 (Ringing) message is received at a Marshal Server.**



VOVIDA.ORG

Policy Server

Policy Server

The Policy Server:

- Administers admission request for bandwidth or quality of service (QoS).
- Interacts with Internetwork Marshal Servers that enforce QoS.
- Interfaces with a clearinghouse to authorize the use of a network for internetworking calls.

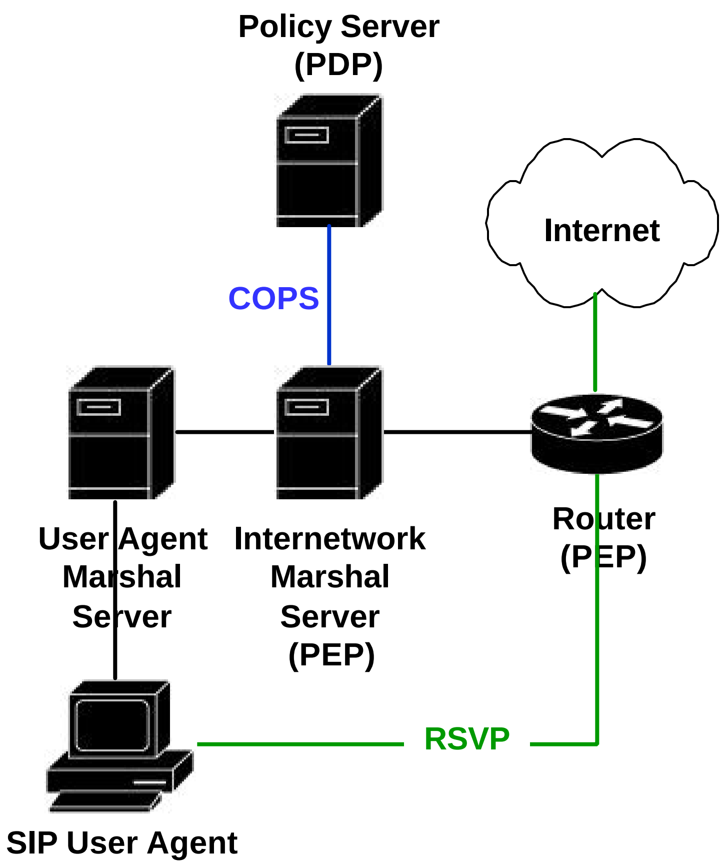
Function of the Policy Server as a COPS Server

The Policy Server:

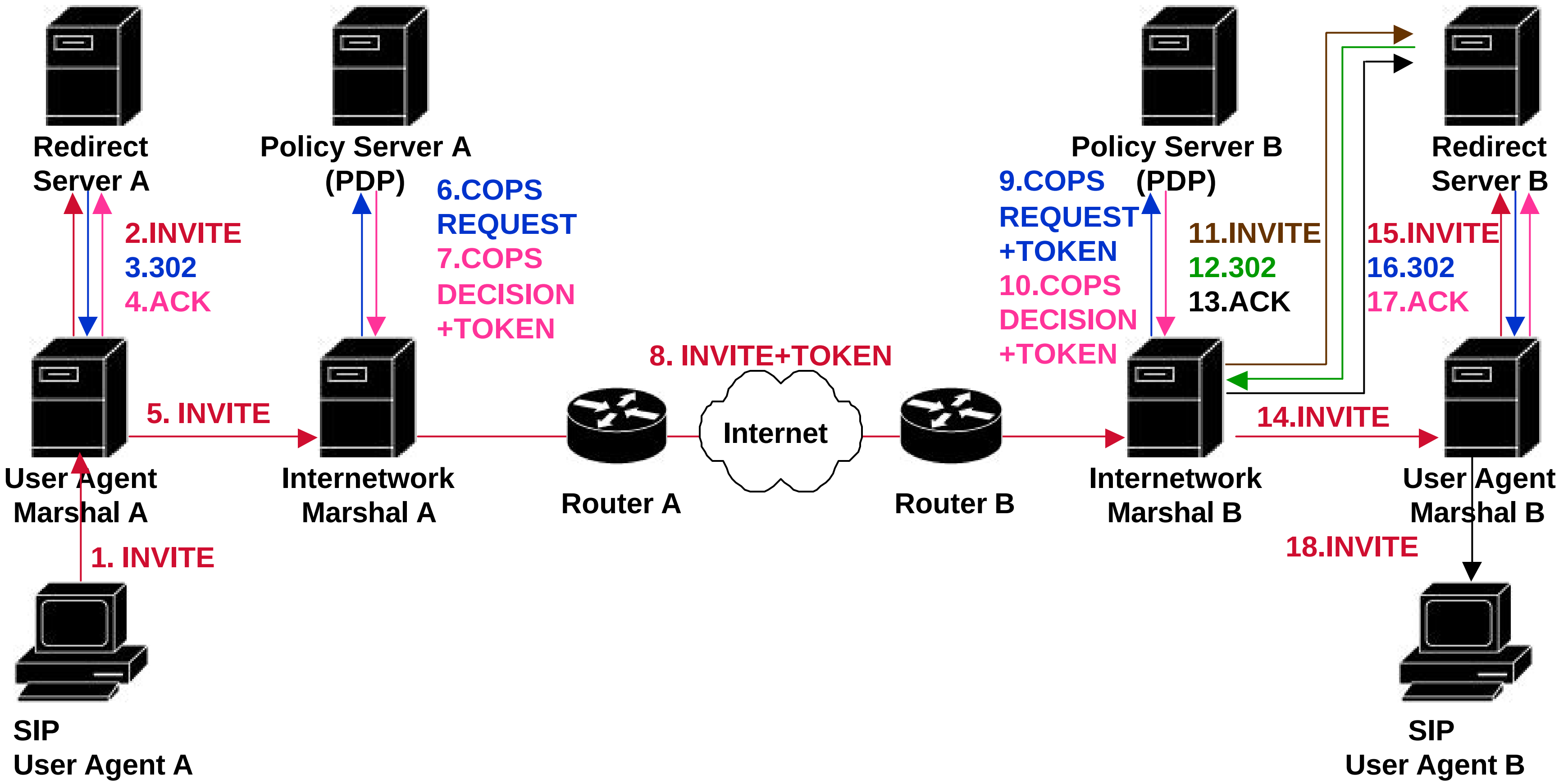
- Acts as a policy decision point (PDP) or COPS server.
- Makes policy decisions to accept or reject authorization requests from policy enforcement points (PEP).

COPS (Common Open Policy Service Protocol) – is used administer authorization.

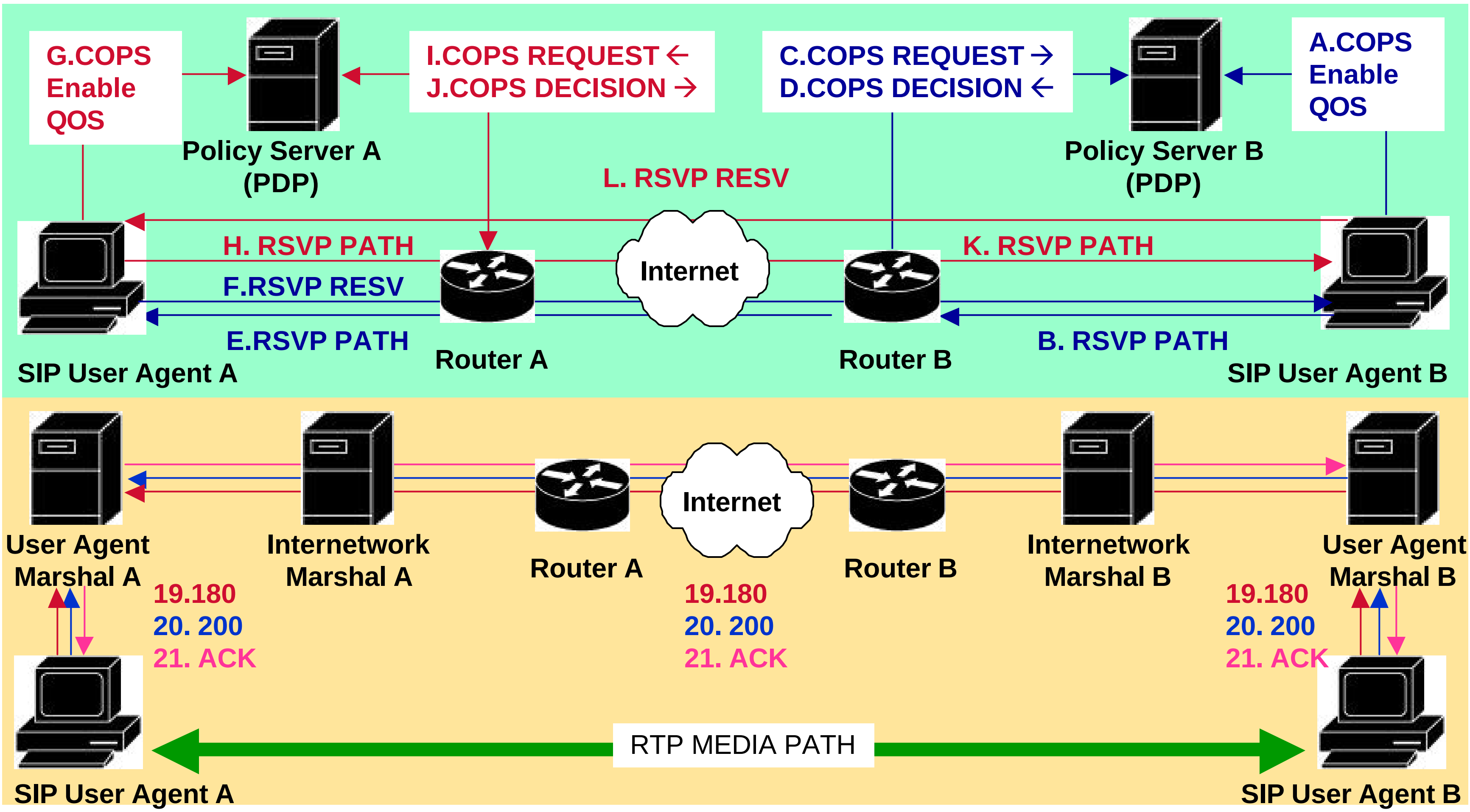
RSVP (Resource Reservation Protocol) – is used to allocate



Requesting Authorization using COPS



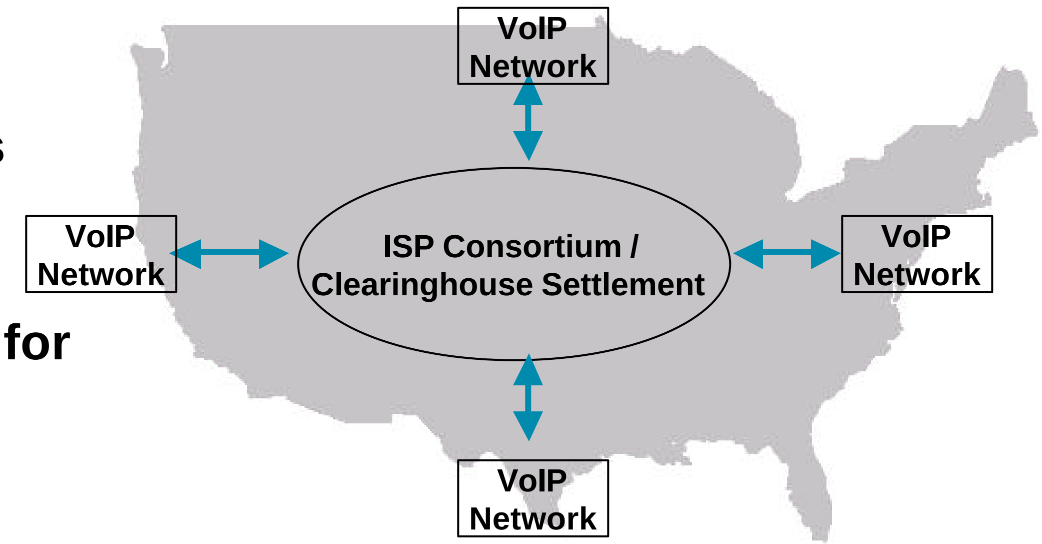
Establishing the Media Path and Requesting Bandwidth



What is a Clearinghouse?

A clearinghouse:

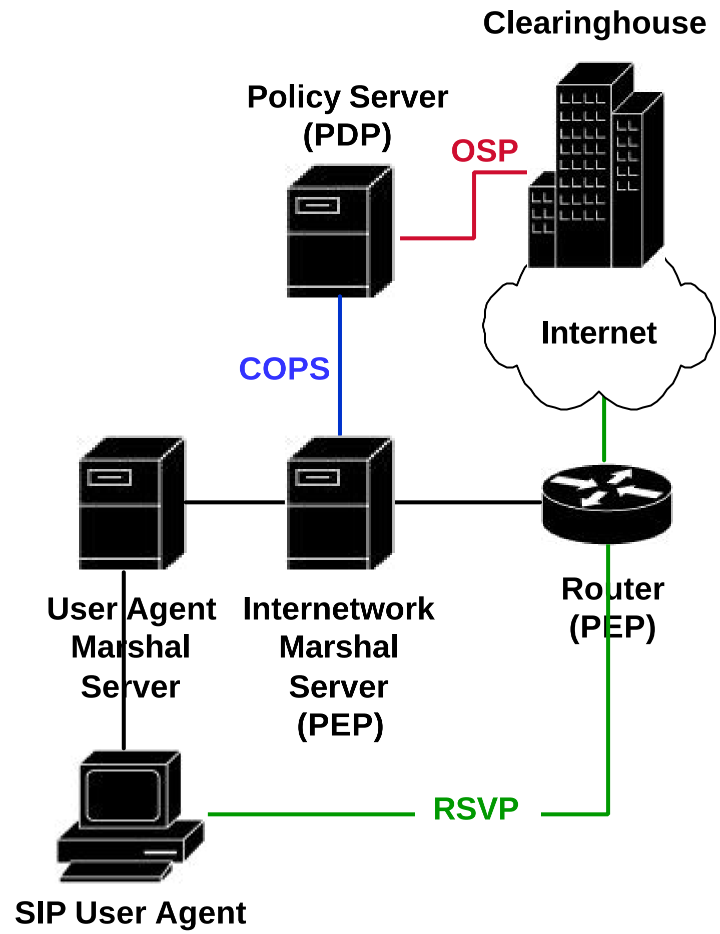
- **Enables the clearing and settlement for shared IP Telephony traffic.**
- **Determines how networks allocate shared traffic.**
- **Provides the essential authorization and routing for shared traffic.**
- **Facilitates the revenue sharing corresponding to the shared traffic.**



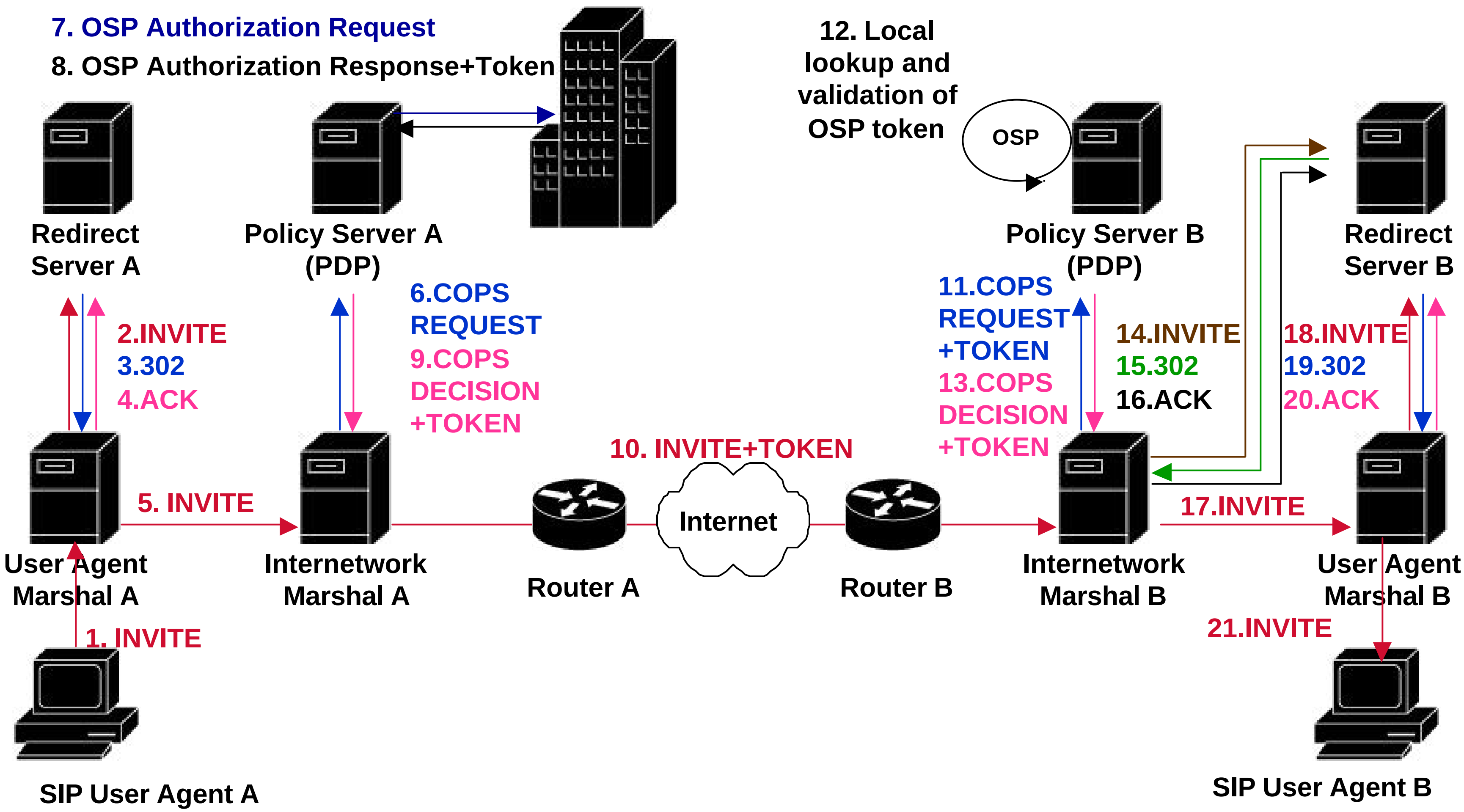
Policy Server – Interactions with a Clearinghouse

The Policy Server:

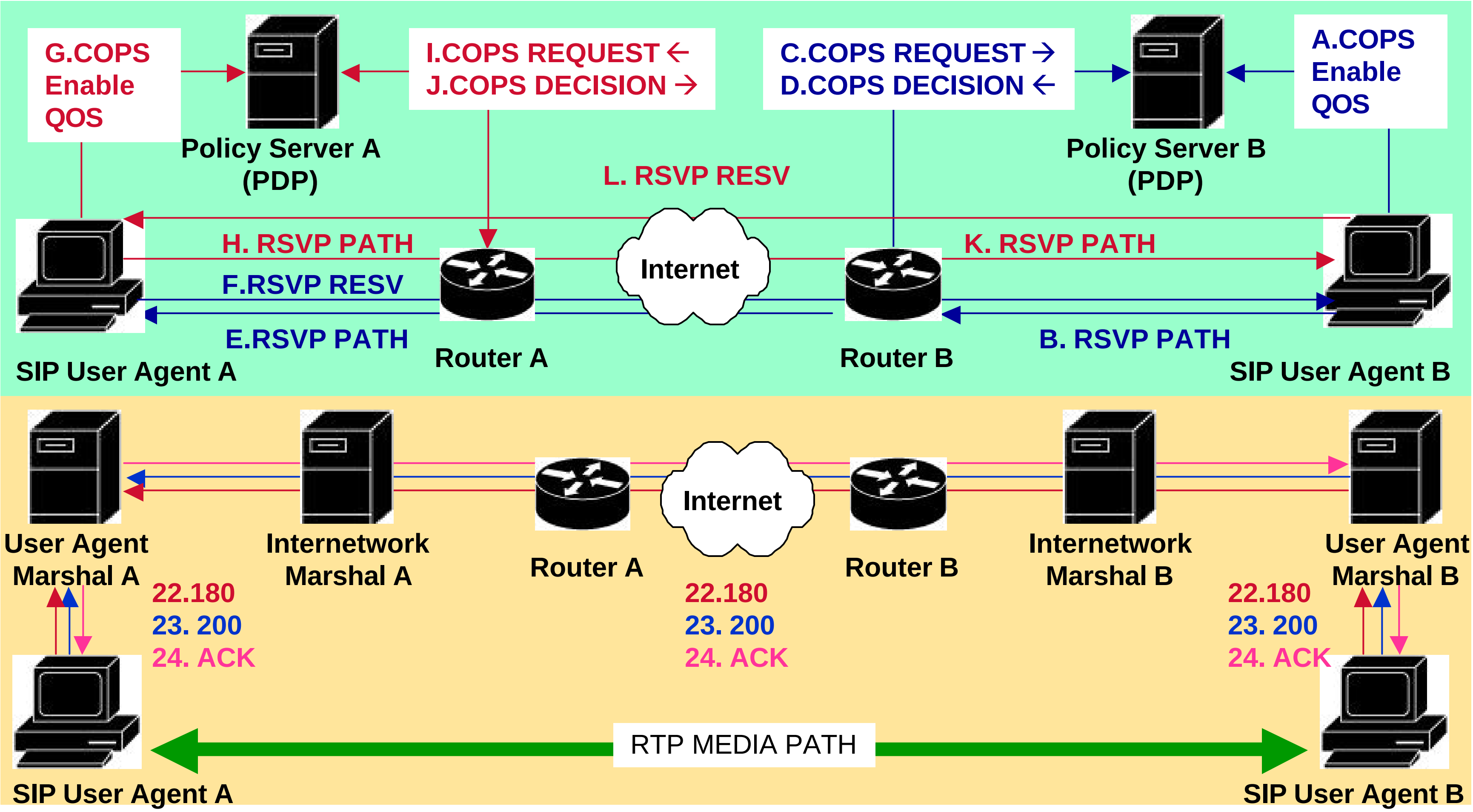
- Can also interact with clearinghouses to authorize the use of a network for internetworking calls.
- Uses the Open Settlement Protocol (OSP) to exchange authentication, authorization, and accounting information.



Requesting Authorization Using COPS and OSP



Establishing the Media Path and Requesting Bandwidth



VOVIDA.ORG

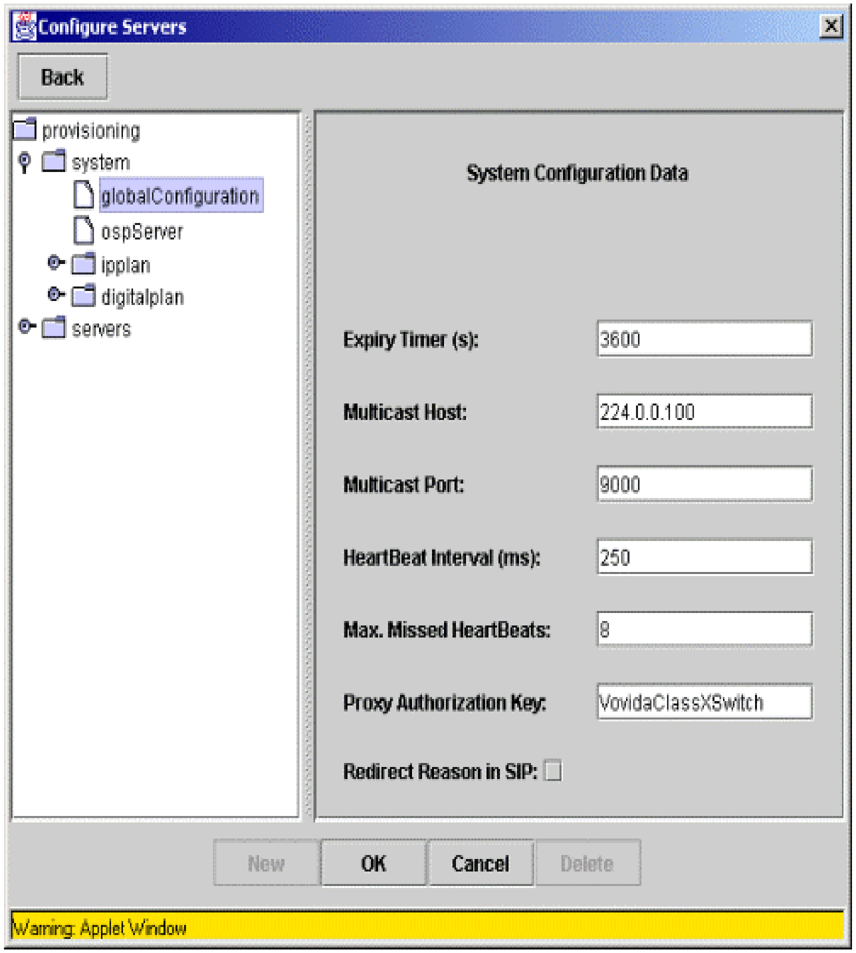
Heartbeat Server

Heartbeat Messages

- VOCAL servers sends and listens heartbeat packets on a multicast port.
- If a VOCAL server does not send a heartbeat packet after a certain time, the server is considered down. Messages intended for this server could be rerouted to its redundant server.
- Not all VOCAL servers send and listen for heartbeat packets.

Configuring the Heartbeat Parameters

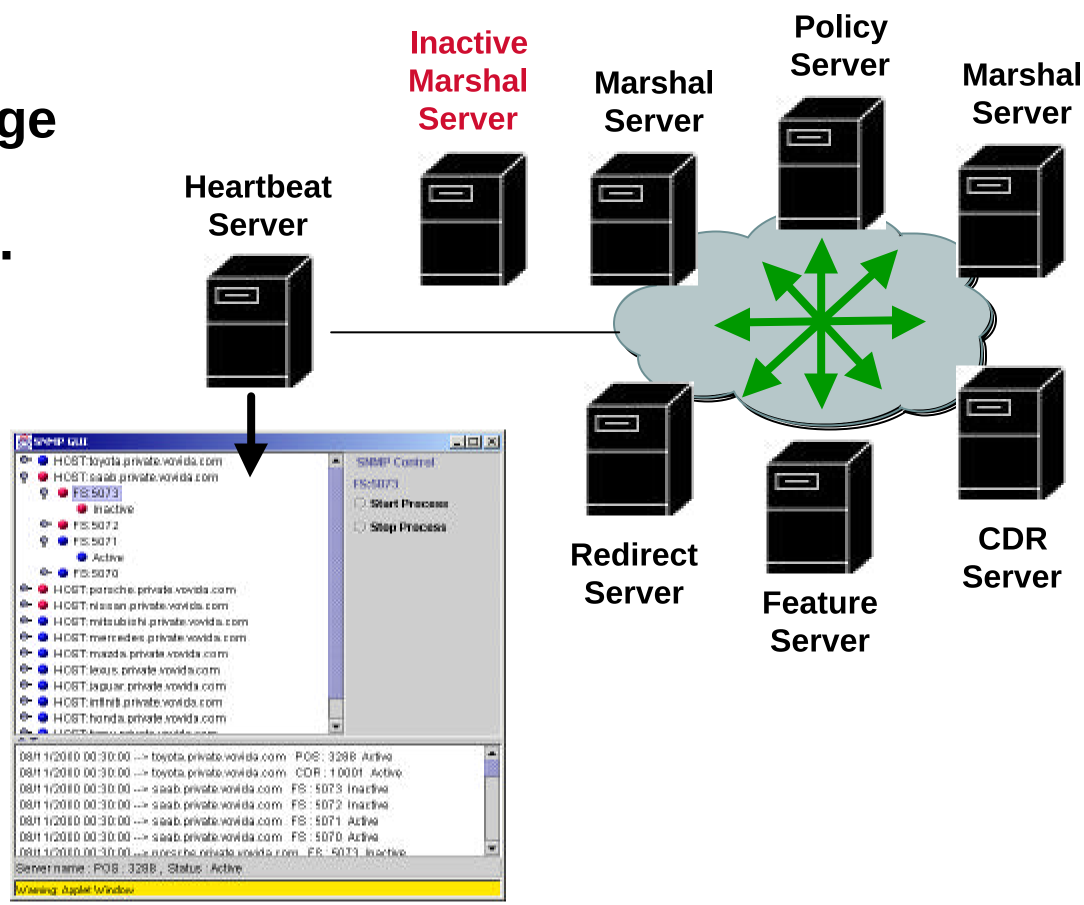
- **Multicast Host/Port:** used to send heartbeat broadcasts.
- **Heartbeat Interval:** time in milliseconds between transmission of heartbeat messages.
- **Maximum Missed Heartbeats:** the maximum number of heartbeat an application can miss before its status becomes inactive.



Heartbeat Server

The Heartbeat Server:

- **Monitors the exchange of heartbeat packets from VOCAL servers.**
- **Sends server status information to the SNMP Network Manager.**



VOVIDA.ORG

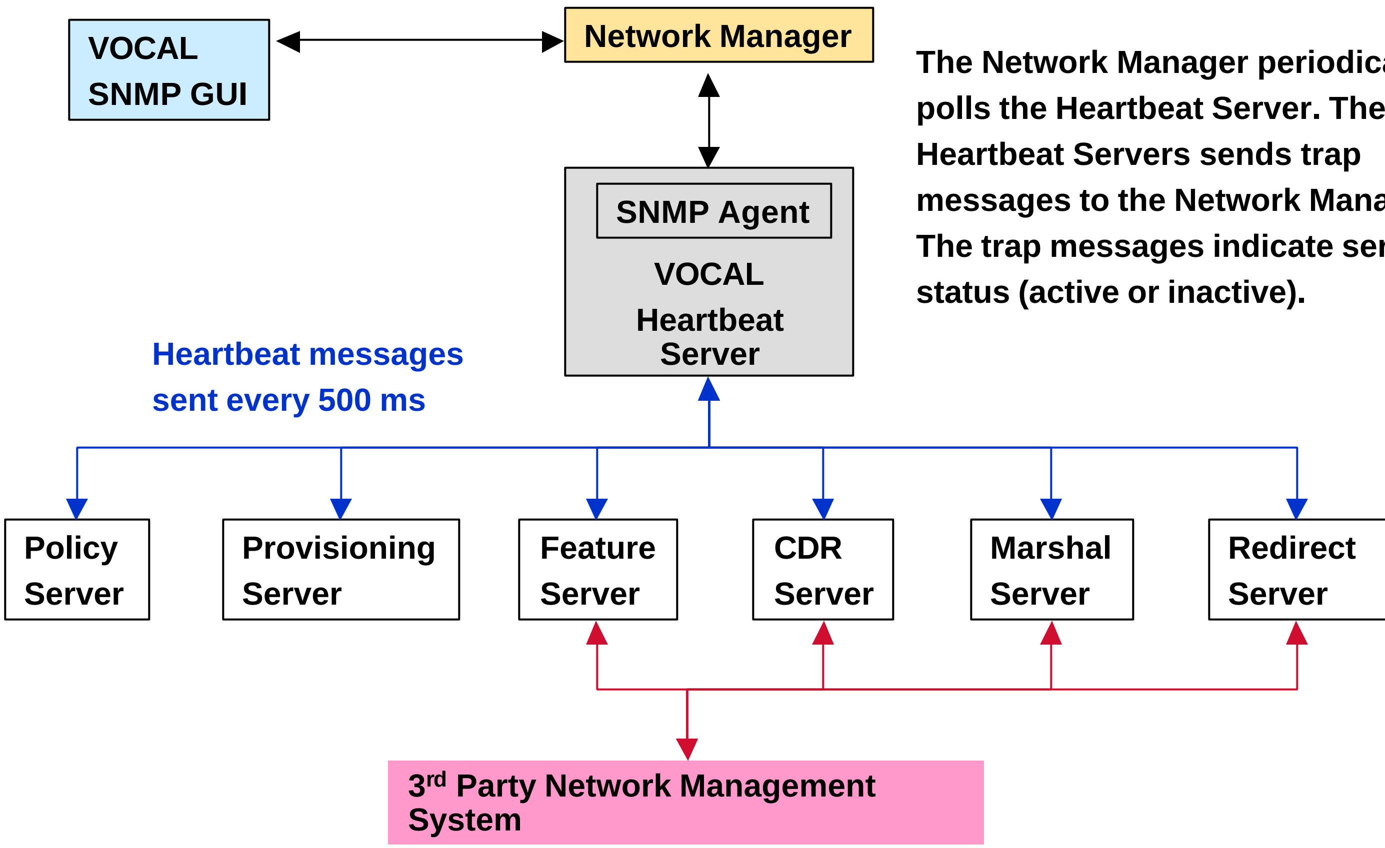
Network Management

VOCAL SNMP Support

VOCAL supports SNMP management and monitoring from:

- **The VOCAL SNMP GUI - this supports monitoring of VOCAL server status.**
- **A third party SNMP network manager, for example, HPOpenView.**

VOCAL SNMP Support

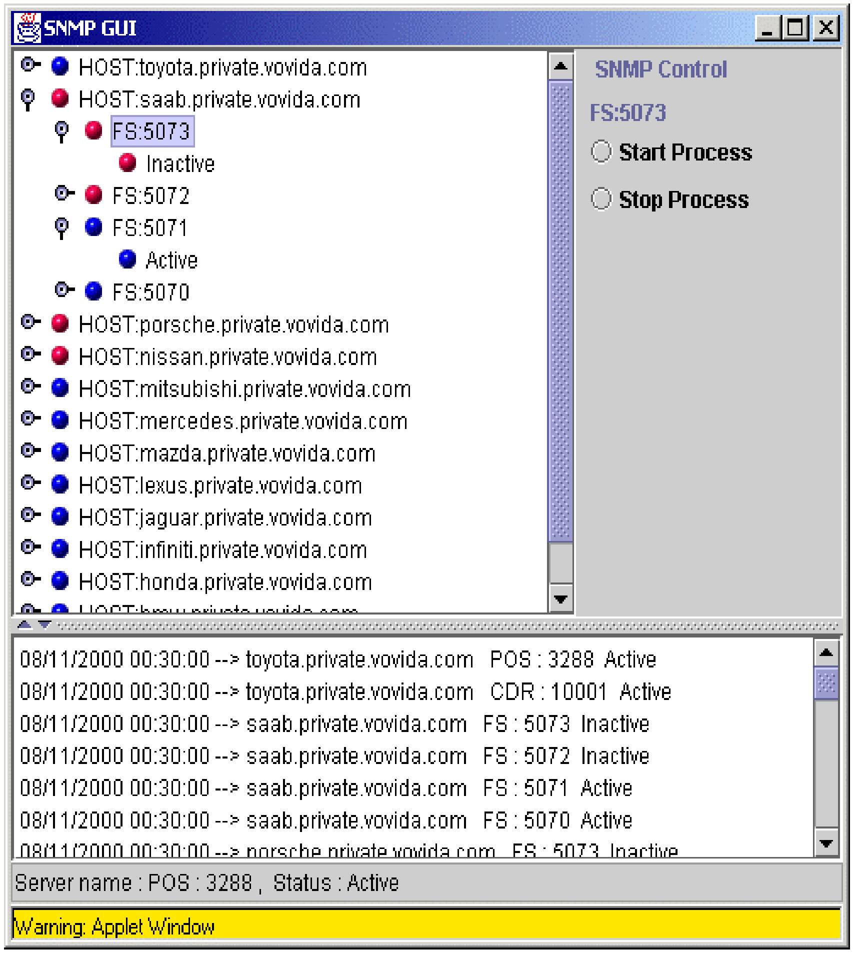


The Network Manager periodically polls the Heartbeat Server. The Heartbeat Servers sends trap messages to the Network Manager. The trap messages indicate server status (active or inactive).

VOCAL SNMP GUI

The VOCAL SNMP GUI provides:

- **SNMP Process Controller - allows you to start or stop the SNMP control process for a server.**
- **Trap message display.**



Supported Management Information Base (MIBs) (1)

VOCAL supports the following public MIBs:

- **RFC 1213 – MIB II.**
- **RFC 2788 - Network Services Monitoring MIB.**
- **SIP MIBS - Draft-ietf-sip-mib-01.txt (July 2000).**

Supported Management Information Base (MIBs) (2)

VOCAL supports the following private MIBs:

- **VOVIDA-LOCAL-GRP-MIB.**
- **VOVIDA-NOTIFICATIONS-MIB.**
- **VOVIDA-SERVERGRP-MIB.**
- **VOVIDA-SOFTSWITCHSTATS-MIB.**
- **VOVIDA-SUBSCRIBERSTATS-MIB.**

More information on each MIB is provided in the MIB file. After installing VOCAL, refer to this directory /usr/local/vocal/proxies/netMgnt.

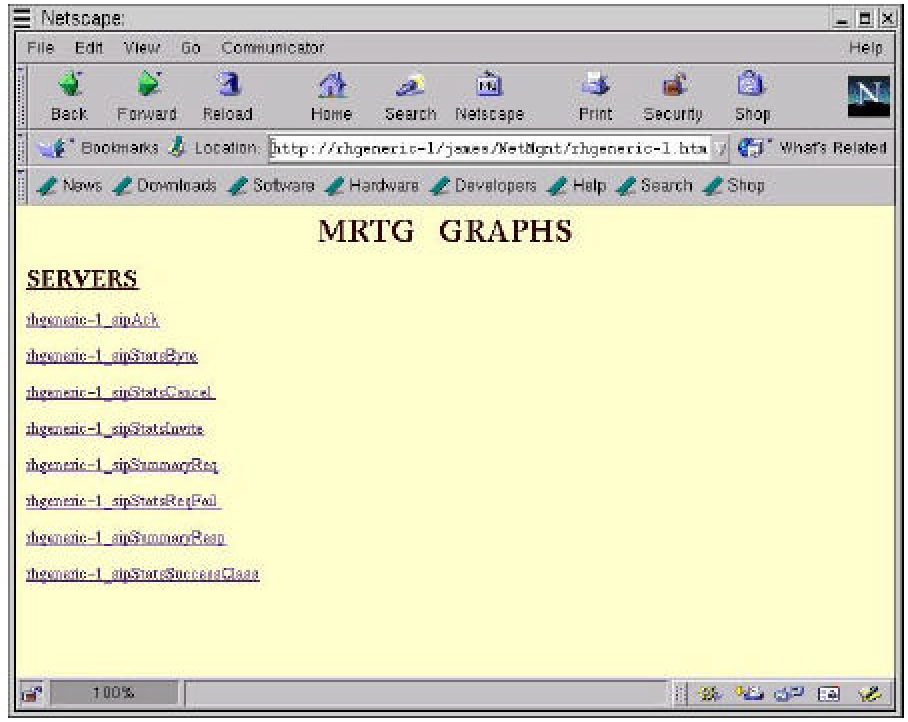
VOVIDA.ORG

Traffic Statistics

Multi-Router Traffic Grapher (MRTG)

MRTG is a 3rd party open source tool that:

- **Monitors traffic on a network.**
- **Generates HTML pages with graphs of network traffic.**



End of Module

This is the end of the VOCAL System Architecture training module.

For additional training and documentation visit us at www.vovida.org.