iPBX

Technical Overview

Version 1.2

Table of Contents

1. Introduction

iPBX is a hybrid TDM and packet voice PBX, and a IVR platform with ACD functionality. iPBX is possibly the most powerful, flexible, and extensible piece of integrated telecommunications software available.

iPBX is designed to interface between any piece of telephony hardware or software, and any telephony application, seamlessly and consistently.

Traditionally, telephony products are designed to meet a specific technical need in a network. However, many applications using telephony share a great deal of technology. iPBX takes advantage of this synergy to create a single environment that can be molded to fit any particular application, or collection of applications, as the user sees fit.

iPBX can, among other things, be used in any of these applications:

Heterogeneous Voice over IP gateway (MGCP, SIP, IAX, H.323)

- Heterogeneous Voice over IP gateway (MGCP, SIP, IAX, H.323)
- Private Branch eXchange (PBX)
- Custom Interactive Voice Response (IVR) server
- Softswitch
- Conferencing server
- Number translation
- Calling card application
- Predictive dialer
- Call queuing with remote agents
- Remote offices for existing PBX

More importantly, it can fill all of those roles, simultaneously and seamlessly, between interfaces. iPBX is designed to allow new interfaces and technologies to be added easily. Its lofty goal is to support every kind of telephony technology available.

In general, interfaces are divided into three categories, Zaptel hardware, non-Zaptel hardware, and packet voice. These interfaces provide integration with traditional and legacy digital and analog telephone interfaces (including connection to the public phone network itself).

In addition, Zaptel compatible interfaces support Pseudo-TDM switching between them, to keep latency nearly nonexistent on strictly TDM calls, conferences, etc.

Zaptel provides connectivity for a variety of network interfaces including PSTN, POTS, T1, E1, PRI, PRA, E&M, Wink, and Feature Group D interfaces among others.

Examples of available hardware:

- 0-24 FXS/FXO (in any configuration) analog PSTN connection
- Single span T1/E1 or PRI connection (mixed data/voice permitted)
- Dual span T1/E1 or PRI connection (mixed data/voice permitted)
- Quad span T1/E1 or PRI connection (mixed data/voice permitted)
- 8 ports T1/E1 or PRI connection (mixed data/voice permitted)
- T3/E3 connection

These are standard protocols, for communication over packet (IP and Frame Relay) networks, and are the only interfaces that do not require any kind of specialized hardware.

- Session Initiation Protocol (SIP)
- Inter-Asterisk eXchange (IAX)
- Media Gateway Control Protocol (MGCP)
- ITU H.323
- Voice over Frame Relay (VOFR)

2. iPBX Architecture Overview

iPBX architecture is simple, and different than most telephony products. iPBX acts as middleware, connecting bottom tier telephony technologies with top tier telephony applications, creating a consistent environment for deploying a mixed telephony environment. Telephony technologies can include VoIP services like SIP, H.323, IAX, and MGCP (both gateways and phones), as well as more traditional TDM technologies like T1, ISDN PRI, analog POTS and PSTN services, Basic Rate ISDN (BRI), and more. Telephone applications include services such as, call bridging, conferencing, voicemail, auto attendant, custom IVR scripting, call parking, intercom, and more.

2.1 Detailed iPBX Architecture

iPBX core contains several engines, that each play a critical role in the software's operation. When the iPBX starts, the *Dynamic Module Loader* loads and initializes each of the drivers which provide channel drivers, file formats, call detail record backends, codecs, applications and more, linking them with the appropriate internal APIs.

Then, the iPBX *Switching Core* begins accepting calls from interfaces and handling them according to the dialplan, using the *Application Launcher* for ringing phones, connecting to voicemail, dialing out outbound trunks, etc.

The core also provides a standard *Scheduler and I/O Manager* that applications and drivers can use. The iPBX *Codec Translator* permits channels which are compressed with different codecs to deamlessly communicate. Most of iPBX usefulness and flexibility come from the applications, codecs, channel drivers, file formats, and more, which plug into the iPBX's various programming interfaces.

SME with Remote Offices

One of iPBX most powerful features is its ability to link remote offices of a SME (Small to Medium Enterprise) together. The above diagram shows how you can build individual small PBXs for multiple offices using iPBX, and then link them together transparently into a single network.

2.1.1 High Density IVR and Conferencing

iPBX can be used as a high density IVR and conferencing platform, with traditional PRI/E1 interfaces, providing redundancy, scalability, and intercommunication using TDM over Ethernet, which permits iPBX to extend to TDM bus across the Ethernet network, while retaining minimal latency.

2.1.2 Naming Channels

Understanding iPBX channel naming convention is critical to using it effectively. Outgoing channel names used, for example, with the Dial application are named in the format:

<technology>/<dialstring>

The <technology> parameter represents which type of interface, you are trying to create or reference to (e.g. SIP, Zap, MGCP, IAX, etc). The<dialstring> is a driver-specific string, representing the required destination. This following section describes the naming convention for each channel type.

2.1.2.1 Zap: Zaptel TDM Channels

Outgoing

The basic formats of a Zap channel name are:

Zap/[g]<identifier>[c]r<cadence>]

Where <identifier> is a numerical identifier for the physical channel number of the required channel. If the identifier is prefixed by the letter g, then the number is interpreted as a group number instead of as a channel. The identifier may be followed by one or more options. If the letter r and a number follow, that number is used as a “distinctive ring” for this dial command (valid numbers are 1-4). If the letter c follows, then "Answer Confirmation" is requested, in which the call is not considered answered until the called user presses '#'.

- Zap/1 TDM Channel 1
- Zap/g1 First available channel in group 1
- Zap/3r2 TDM Channel 3 with 2nd distinctive ring
- Zap/g2c First available channel in group 2 with confirmation

Incoming

Incoming Zap channels are labeled simply:

Where <channel> is the channel number and <instance> is a number from 1 to 3 representing which of up to 3 logical channels associated with a single physical channel this is.

- Zap/1-1 First call appearance on TDM channel 1
- Zap/3-2 Second call appears on TDM channel 3

2.1.2.2 SIP: Session Initiation Protocol Channels

Outgoing

Outgoing channels typically take the form:

SIP/[<exten>@]<peer>[:<portno>]

Where <peer> is the name of the peer (or hostname/IP of the remote server), <portno> is an optional port number (the default is the SIP standard port 5060), and <exten> is an optional extension.

- SIP/ipphone SIP peer ipphone
- SIP/8500@sip.com:5060 Extension 8500 at sip.com, port 5060

Incoming

Incoming SIP channels are of the form:

SIP/<peer>-<id>

Where <peer> is the identified peer and <id> is a random identifier that uniquely identifies multiple calls from a single peer.

- SIP/192.168.0.1-01fb34d6 - A SIP call from 192.168.0.1
- SIP/sipphone-45ed721c - A SIP call from peer sipphone

2.1.3 Dialplan

The most important part of comprehending iPBX is understanding how the dialplan works. Dialplan routes every call in the system from its source through various applications, to its final destination. Everything from voicemail, to conferencing, to auto attendant voice menus is done through a consistent concept and logic.

2.1.3.1 Extension Context Uses

The dialplan is composed of one or more extension contexts. Each extension context is a collection of extensions. Each extension context in a dialplan has a unique name associated with it. The use of contexts can be used to implement a number of important features including:

- Security - Permit long distance calls from certain phones only
- Routing - Route calls based on extension
- Auto attendant - Greet callers and ask them to enter extensions
- Multilevel menus - Menus for sales, support, etc.
- Authentication - Ask for passwords for certain extensions
- Callback - Reduce long distance changes
- Privacy - Blacklist annoying callers from contacting you
- PBX Multihosting - Yes, you can have .virtual hosts. on your PBX
- Daytime/Nighttime - You can vary behavior after hours
- Macros - Create scripts for commonly used functions

2.1.3.2 Dialplan Concepts

The goal of this section is to familiarize you with the concepts behind the dialplan, show some examples, and empower you with the knowledge you need to perform neat tricks and impress friends, co-workers, and competitors with your iPBX-foo, like a pro.

2.1.3.2.1 Basic Extension Context

Extension context example 1:default

default	
Extension	Description
101	Mark Spencer
102	Will Meadows
103	Greg Vance
104	Check voicemail
105	Conference Room
0	Operator

In this context example, the first three extensions (101 to 103) all would be associated with ringing phones belonging to various employees. The fourth extension (104) would be associated with allowing someone to check their voicemail. The fifth extension (105) would be associated with a conference room. Finally, the .0.extension would be associated with the operator.

Extension context example 2: mainmenu

mainmenu	
Extension	Description
s	Welcome message and instructions
1	Sales
2	Support
3	Accounting
9	Directory
#	Hangup

Example 2, only has single digit extensions. The .S. extension is the start extension, where the caller begins. This extension would play a message along the lines of "Thank you for calling Our Company> Press 1 for sales, 2 for support, 3 for accounting, 9 for a company directory, or # to hang-up". Each menu option is an extension and could either, for example, dial someone's real extension, or send someone to another menu.

2.1.3.2.2 Pattern Matching

Extensions can also match patterns, instead of being single digits. Patterns to be pattern matched must start with the underscore character (._.) and may use any of the following special characters:

X . Any digit from 0-9
N . Any digit from 2-9
[14-6] . Any a 1,4, 5, or 6
. . Matches anything

For example, consider the following context:

routing	
Extension	Description
_61XX	Rome Office
_62XX	Paris Office
_63XX	London Office
_7[1-3]XX	New-York Office
_7[04-9]XX	India Office

This context splits calls according to their extension to be sent to various servers. In this example, it is assumed that all extensions are four digits long (iPBX has no such requirement, nor is there a requirement that all extensions be the same length). Anything starting with 61, would be sent to the Rome office, 62 would go to the Paris office, 71, 72, or 73 would go to New-York and so on.

2.1.3.2.3 Context Inclusion

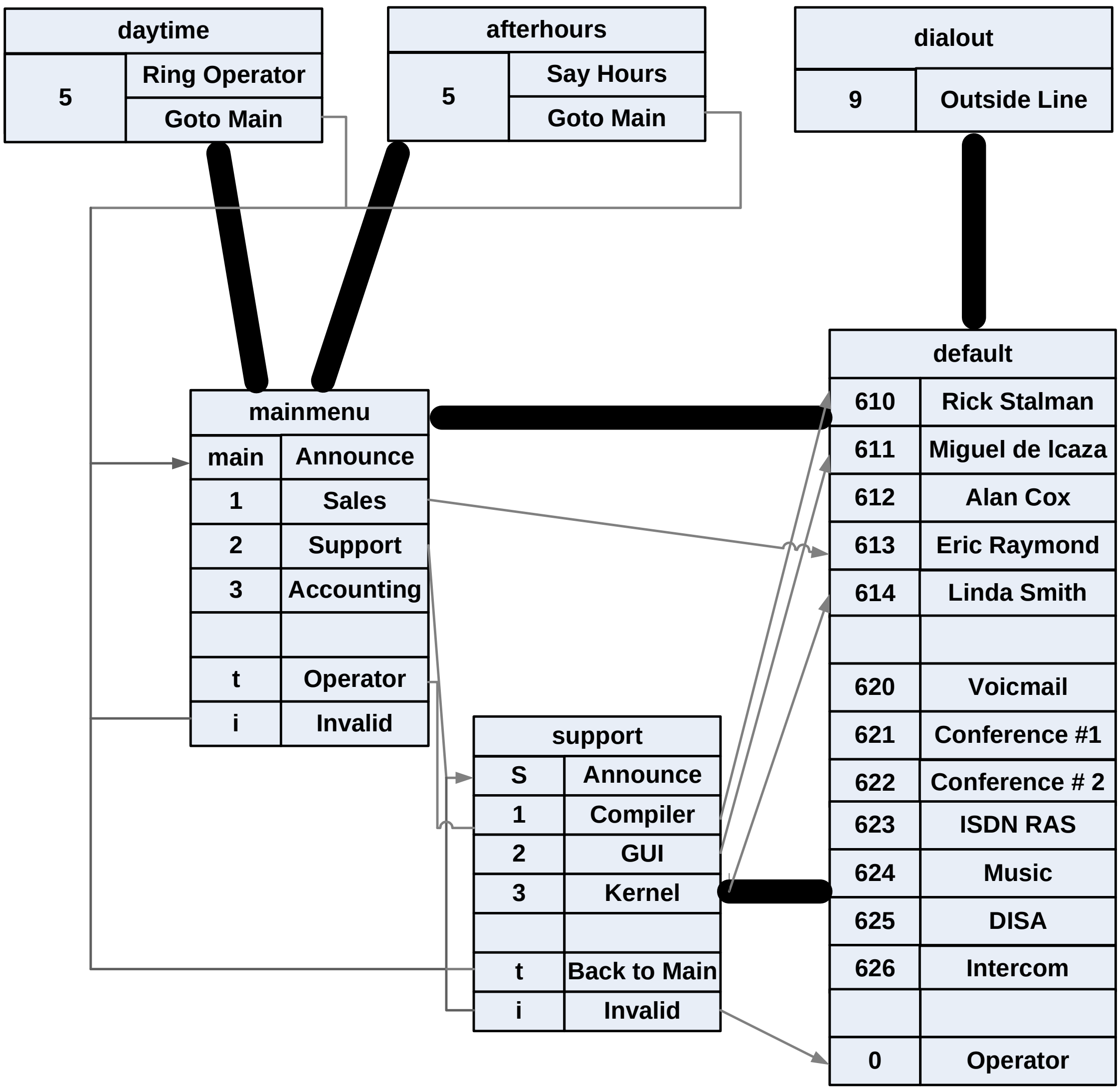
One extension context can include the contents of another. For example, consider the following contexts:

longdistance	
Extension	Description
_91NXXNXXXXXX	Long distance calls

local	
Extension	Description
_9NXXXXXX	Local calls

Here, a context called local provides a single extension for dialing local calls, and includes the default extension as well. Then, there is a longdistance context which includes an extension for long distance calling, and includes the local context. Phones which are in the longdistance context would be able to make long distance calling. Those in "local" could only make 7 digit local calls, and those in the default context would not have outside line access at all. Thus, using extension contexts, you can carefully control who has access to toll services.

2.1.3.2.4 Complete Set of Contexts



A more complete example of a dialplan may be helpful. Here is a dialplan for a fictional Computer company. Several contexts are listed with familiar names.

Thick black lines indicate an extension context being included in another. A slim gray line indicates one extension sending you into another extension or extension context. In general, extensions that are local to you company should be listed under default context since they should be accessible by anyone at more-or less anytime. Typically, no actual phones would be associated with the default context alone, but it could be associated with a VoIP “guest” account, for example. A local context, in this case dialout, can include the default context as well as either local only or long distance dialing. It is important that access to the local context no be permitted from “guest” accounts, unless it is you intention to let people use your local phone resources.

Next we have a couple of menu contexts, mainmenu and support. Both of these present menus while including the default context, so that direct extensions may be dialed at any time. The mainmenu context includes both daytime and after hours so that an incoming call rings to an operator first during the day, and directly to a company announcement after working hours.

Call Features

- ADSI On-Screen Menu System
- Alarm Receiver
- Append Message
- Authentication
- Automated Attendant
- Blacklists

Call F/T

- Blind Transfer
- Call Forward on Busy
- Call Forward on No Answer
- Call Forward Variable
- Call Transfer
- Call Detail Records

Call Options

- Call Monitoring
- Call Parking
- Call Queuing
- Call Recording
- Call Retrieval
- Call Routing (DID & ANI)
- Call Snooping
- Call Waiting

- Dial by Name
- Remote Call Pickup
- Route by Caller ID

Call ID Options

- Caller ID
- Caller ID Blocking
- Caller ID on Call Waiting
- Calling Cards (Pre/Post paid)

- Conference Bridging
- Database Store / Retrieve
- Database Integration
- Direct Inward System Access
- Distinctive Ring
- Do Not Disturb
- E911
- ENUM
- Fax Transmit and Receive (3rd Party OSS Package)
- Flexible Extension Logic
- Interactive Directory Listing

Features Summary

- Interactive Voice Response (IVR)
- Local and Remote Call Agents
- Macros
- Music On Hold
- Music On Transfer
- Flexible Mp3-based System
- Random or Linear Play
- Volume Control
- Predictive Dialer
- Privacy
- Open Settlement Protocol (OSP)
- Overhead Paging
- Protocol Conversion
- Remote Call Pickup
- Remote Office Support
- Roaming Extensions
- Route by Caller ID
- SMS Messaging

- Spell / Say
- Streaming Media Access
- Supervised Transfer
- Talk Detection
- Text-to-Speech (via Festival)
- Three-way Calling
- Time and Date
- Transcoding
- Trunking
- VoIP Gateways
- Voicemail
- Visual Indicator for Message Waiting
- Stutter Dialtone for Message Waiting
- Voicemail to email
- Voicemail Groups
- Web Voicemail Interface
- Country Specify Tones

Computer-Telephony Integration

- TAPI (Telephony Application Programming Interface)
  An API for connecting a PC running Windows to telephone services The standard supports connections by individual computers as well as LAN connections serving many computers. Within each connection type, TAPI defines standards for simple call control and for manipulating call content.
- JTAPI (JAVA Telephony Application Programming Interface)
  The Java Telephony API supports telephony call control. It is an extensible API designed to scale for use in a range of domains, from first-party call control in a consumer device to third-party call control in large distributed call centers.
- Graphical Call Manager
- Outbound Call Spooling
- Predictive Dialer
- TCP/IP Management Interface

Scalability

- TDMoE (Time Division Multiplex over Ethernet)
- Allows direct connection of Asterisk PBX
- Zero latency
- Uses commodity Ethernet hardware
- Voice-over IP
- Allows for integration of physically separate installations
- Uses commonly deployed data connections
- Allows a unified dialplan across multiple offices

Codecs

- ADPCM
- G.711 (A-Law & μ-Law)
- G.723.1 (pass through)
- G.726
- G.729
- GSM
- iLBC
- Linear
- LPC-10
- Speex

Protocols

- IAX™ (Inter-Asterisk Exchange)
- H.323
- SIP (Session Initiation Protocol)
- MGCP (Media Gateway Control Protocol
- SCCP (Cisco® Skinny®)

Traditional Telephony Interoperability

- E&M
- E&M Wink
- Feature Group D
- FXS
- FXO
- GR-303
- Loopstart
- Groundstart
- Kewlstart
- MF and DTMF support
- Robbed-bit Signaling (RBS) Types
- MFC-R2

PRI Protocols

- 4ESS
- BRI (ISDN4Linux)
- DMS100
- EuroISDN
- Lucent 5E
- National ISDN2
- NFAS

For more information please contact : sales@10levels.com