

Epicode IraCPA

Time has been a discerning factor in judging customer service and calls since losing time can be expensive for contact centers. IraCPA with its ability to filter out obstacles such as busy signals, answering machines, and disconnected calls can cut down costs on a large scale.

The combination of IraCPA with an outbound dialler has filled the gap by fueling sales and optimizing customer satisfaction and thus has emerged as a one-point solution for organizations looking to invest in opportunity scouting and accessing large amounts of data.

IraCPA enables agents to

- *Experience less idle time*
- *Switch effortlessly between inbound and outbound*
- *Contact customers at a favourable time*
- *Drive customer satisfaction and enhance customer loyalty*

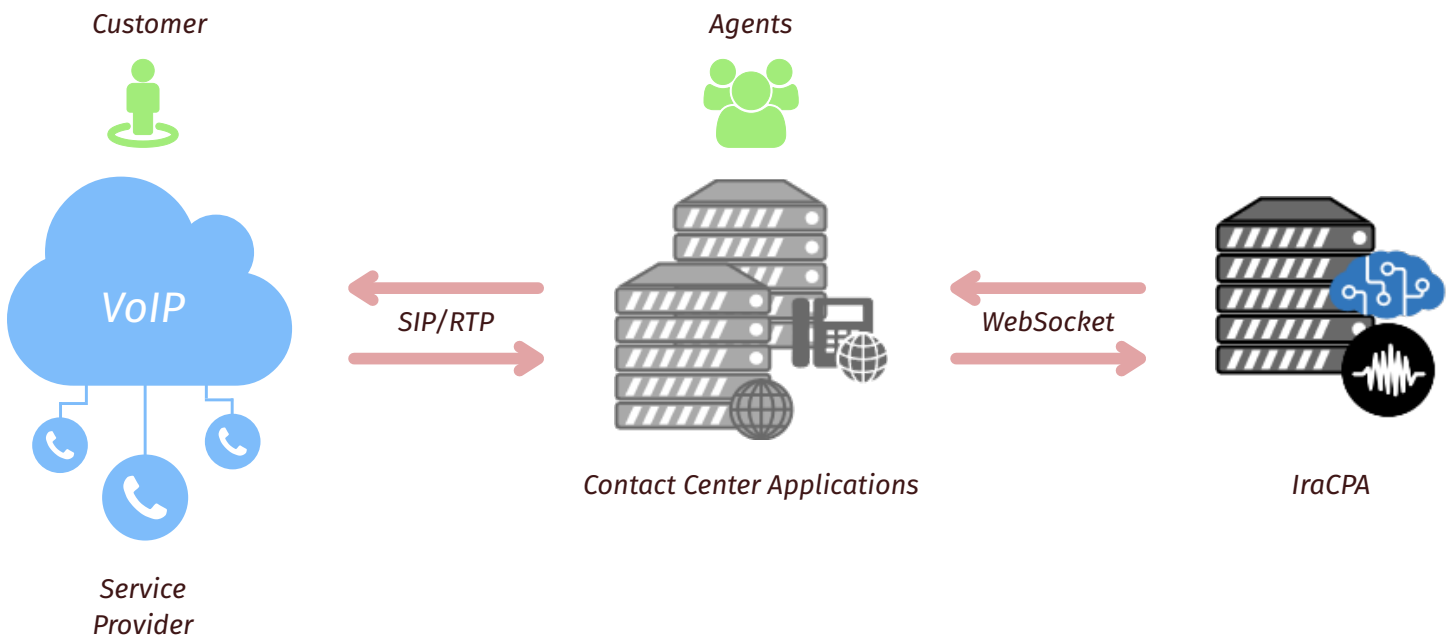


1 How does IraCPA help customers?

The Epcode IraCPA API allows users to do call progress analysis for their dialler applications. With these APIs one can:

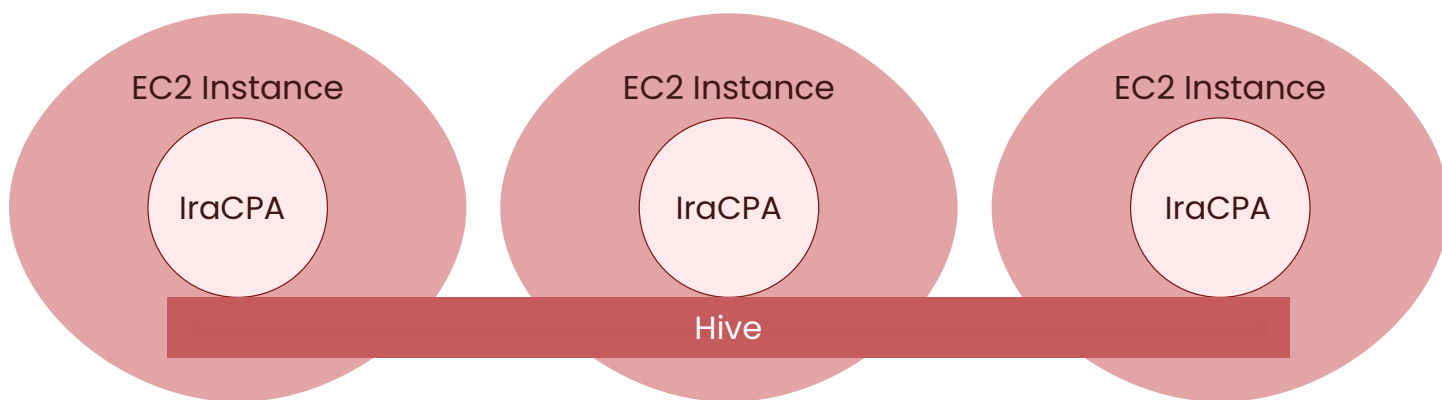
- Determine if a call is answered by an answering machine, live voice or fax machine.
- Listen to specific frequency tones like beeps from an answering machine.
- Stop the analysis upon receiving one of these events.

Each API is a JSON body request over a WebSocket connection. The following sections describe the JSON request and response, and how to send the RTP data.



2 API Interface

The API interface is over a WebSocket connection. The API caller just needs to know the IP address and the port number to connect to the IraCPA service. The URL could be `http://IP_ADDRESS:PORT`. The deployment can contain multiple EC2 instances or multiple physical servers running IraCPA software on Hive, a HA framework from Epicode. The load balancing can be achieved either by using a floating IP address or by using round-robin across all the servers.



If the port number is an even number, use plain WebSocket. If the port number is an odd number, use a secure WebSocket.

3 Set IraCPA Parameters

This API allows configuration of the various parameters which controls the behaviour of the call progress analysis engine. Some prominent configurations that can be set are:

- Time limit for first detection, that LV or AM.
- The specific frequencies to detect other than identifying humans and answering machines.
- The detection event on which the IraCPA engine can stop the analysis.
- Total analysis time limit.

3.1 Request Body (JSON)

Use text mode in the WebSocket to send this JSON request. Keep listening to any replies asynchronously.

Key	Description	Example	Default
request_name REQUIRED string	Specify the request name	set_cpa_params	None
tenant_id REQUIRED string	Name of the licensee	acme	None
api_version REQUIRED string	Currently use 1.0	1.0	None
config_name REQUIRED string	Unique name for the configuration	"my_setup"	None
analysis REQUIRED string	Name of the call progress analysis	amd	None
min_ambient_energy integer	Threshold that separates perfect initial silence and ambient silence	1	1
initial_silence_ignore integer	If the energy level is lower than minimum ambient energy, ignore the RTP packets	5000	750
sensitivity integer	Just use the value 5 for now	4	4
tones object (key, value pairs)	The single tone frequencies to be detected and the tolerance levels for each frequency. This is a json object with the each key being a name for the frequency and value being the frequency and tolerance in Hertz separated by a pipe()	{ "FX": "2100 5", "MD": "1662 5", "BP": "1000 10" }	{}
log_top_freq integer	How many top frequencies to log. Use only for debugging purposes	0	0
log_voice string	Tells the system whether or not to record and store the analysis stream. Must be either yes or no	"yes"	"no"
break_events string	Comma separated list of frequencies or events on which the CPA analysis can stop	"FX,MD,AM"	None
total_timeout integer	The time in milliseconds at which the analysis will stop	20000	15000

3.2 Sample Request (JSON)

```
{
  "request_name" : "set_cpa_params",
  "tenant_id" : "acme",
  "api_version" : "1.0",
  "config_name" : "my_setup",
  "time_limit" : 1750,
  "sensitivity" : 5,
  "log_voice" : "no",
  "tones" :
  {
    "FX" : "2100|5",
    "MD" : "1662|5",
    "BP" : "1000|10"
  },
  "break_events" : "FX,MD,BP",
  "log_top_freq" : 3,
  "total_timeout" : "20000"
}
```

3.3 Response Body (JSON)

If the IraCPA parameter is set in one server, it will take effect in every server in the IraCluster. You can set the iraCPA_parameter using one server, and use config_name in the next server.

Field	Description
result string	OK - No error 01 - Mandatory fields are not specified 02 - Could not save the IraCPA configuration
reason string	Will provide the descriptive message about the result

4 Make IraCPA Request

This API sends the IraCPA request, followed by the RTP data.

4.1 Request Body (JSON)

Use text mode in the WebSocket to send this JSON request. Keep listening to any replies asynchronously.

Key	Description	Example	Default
tenant_id REQUIRED string	Name of the licensee	acme	None
api_version REQUIRED string	Currently use 1.0	1.0	None
config_name REQUIRED string	Unique name for the configuration	"my_setup"	None

4.2 Sample Request (JSON)

This API request will initialize the IraCPA analyzer. After the request is sent, the WebSocket mode should be changed to binary, and RTP data should be sent in 16 bit PCM format. Preferably send 4000bytes every 250milliseconds. The results will be coming back whenever the IraCPA analyzer makes a detection.

```
{  
  "tenant_id": "acme",  
  "api_version": "1.0",  
  "config_name": "my_setup"  
}
```

4.3 Response Body (JSON)

Field	Description
result string	LV - Live voice found AM - Answering Machine found TO - Time out, total time exceeded <i>The result could also be a custom tone specified under tones.</i> 01 - Mandatory fields are not specified 03 - Specified IraCPA configuration not found 04 - No license available for the tenant 05 - Specified IraCPA analyzer not found 06 - RTP data was sent without initializing the analyzer
reason string	Will provide the descriptive message about the result.
break string	yes/no - Tells the caller to stop sending RTP and disconnect the session.
max_energy string	The maximum energy found in the voice data.
pattern string	Energy pattern seen in the voice data.
duration string	Duration of the data analysed.
request_count string	Count of the requests handled so far.

Conclusion

IraCPA can be deployed along with IraDialer to improve overall efficiency of the outbound voice campaigns by ensuring that only those calls that are answered by a human are routed to the call center agent while dropping the calls that land on answering machines.

IraCPA has a proven success rate of 85 % and can complete the detection process in less than two seconds to meet the telecom regulations in countries like USA (TCPA compliance)

Answering Machine Detection (AMD) is a complicated technology requiring real time audio signal processing and complex algorithms.



Unit # 2201A, 22nd Floor, World Trade Centre, Rajajinagar | Bangalore
Karnataka 560055 | Phone: +918067935393
Email: enquire@epicode.in
