EPICODE

API Reference

BotStream

Epicode Design Team

19th April, 2022

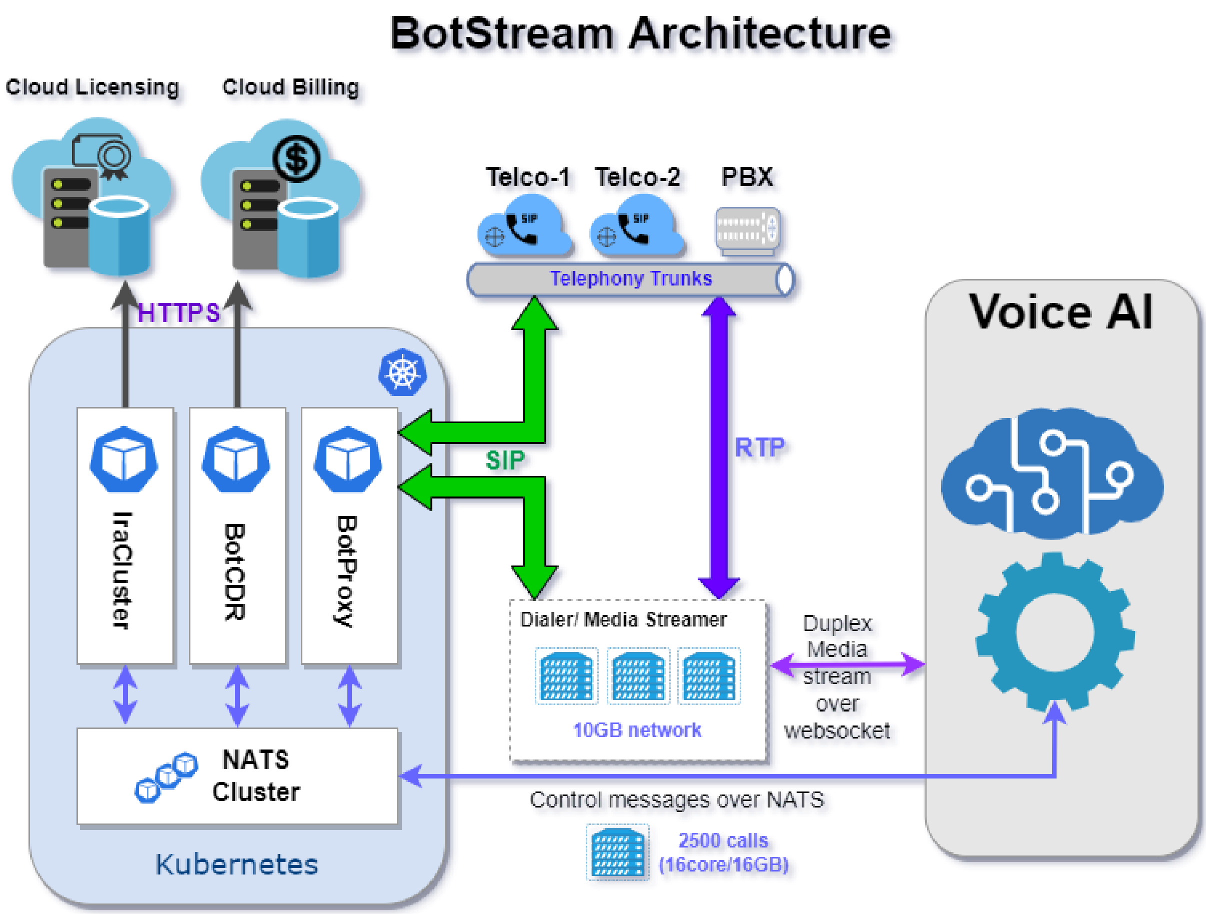
Unit # 2201A, 22nd Floor, World Trade Centre, Rajajinagar, Bangalore - 560055
CIN: U72900KA2020PTC131771
corporate@epicode.in

1 Introduction

BotStream is a bidirectional voice streamer that interfaces telephony services with conversational voice AI applications. It includes an API Dialer, Call recording, Trunk manager, QOS monitor and CPA. It supports both Inbound and outbound voice traffic and can integrate with PSTN or any PBX over E1 or SIP protocol while communicating with the conversational AI application over WebSockets. The entire BotStream setup resides in the Kubernetes cluster, and is capable of auto scaling according to the load.

BotStream comprises one or more telephony switches, front ended by a SIP proxy for load balancing. This allows BotStream to scale horizontally as the load increases. The high speed communication between the telephony switches and the VoiceAI engines is enabled via NATS platform. The BotStream is built on the popular Freeswitch telephony platform.

2 BotStream Architecture



The VoiceAI engines communicate with BotStream in two different ways. NATS for sending requests and receiving events. Web Socket for sending and receiving raw audio data. The BotStream deals with all the VOIP communication with telephony trunk providers.

3 BotStream API

Every API request sent to BotStream must be a json object with the following schema.

```
{
   "$schema": "http://json-schema.org/draft-04/schema#",
   "type": "object",
```

```
  "properties": {
    "event_name": {"type": "string", "pattern": "^request_" },
    "event_data": {"type": "object"}
  },
  "required": [ "event_name", "event_data"]
}
```

Depending on the API request, the event_data will have different schemas, which will be described in the following sections. The API request should be sent via NATS to a pre-configured subject, on which BotStream will be listening. All the APIs are synchronous, and will return a reply right away. Call the APIs using synchronous Request function from NATS, with 5 seconds as timeout.

3.1 Set CPA Parameters

This API is used for configuring the CPA, if AMD/FAX/LiveVoice detection must be performed after an outbound dialed call is answered.

Field event_name should be set to **request_set_cpa_params**.

Field event_data should conform to the following schema:

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "va_config_name": {"type": "string" },
    "tenant_id": {"type": "string" },
    "analysis": {"type": "string", "enum":["amd"]},
    "min_ambient_energy": {"type": "integer", "minimum": 1},
    "initial_silence_ignore": {"type": "integer", "minimum": 0, "maximum": 5000},
    "time_limit": {"type": "integer", "minimum": 1750, "maximum": 2250},
    "sensitivity": {"type": "integer", "minimum": 2, "maximum": 5},
    "tones": {"type": "object"},
    "log_top_freq": {"type": "integer", "minimum": 0, "maximum": 5},
    "log_voice": {"type": "boolean", "enum" : [true,false] },
    "break_events": {"type": "string"},
    "total_timeout": {"type": "integer", "minimum": 10000},
    "event_subject": {"type": "string"}
  },
  "required": [ "va_config_name", "tenant_id", "analysis"]
}
```

Field Name	Description	Default	Example
va_config_name	Unique name	NONE	my-cpa-1
tenant_id	Billing entity	NONE	acme
analysis	CPA name	amd	amd
min_ambient_energy	Ambient noise level	1	1
initial_silence_ignore	How much silence to ignore	750	2000
time_limit	AMD detection duration (ms)	1750	2250
sensitivity	Detection sensitivity	4	4
tones	Frequency and tolerance for detecting specific tones. "FX" : "2100|20" means frequencies between 2080Hz-2120Hz will be detected as FX.	NONE	{ "FX" : "2100|20", "BP" : "1000|10" }
log_top_freq	Log top frequencies found	3	0
log_voice	Should CPA audio be logged	true	false
break_events	Stop CPA upon these events	NONE	“AM,LV,FX”
total_timeout	Total duration of CPA (ms)	15000	10000

Upon success, the reply will be as follows:


```
{
        "request_name": "request_set_cpa_params",
        "status": "OK",
        "status_code": 0
}
```


3.2 Make Outgoing Call

The API is used for making outgoing calls.

Field event_name should be set to **request_make_call**.

Field event_data should conform to the following schema:

```
{
   "$schema": "http://json-schema.org/draft-04/schema#",
```

```
  "type": "object",
  "properties": {
    "gateway": {"type": "string"},
    "tenant_id": {"type": "string" },
    "to_number": {"type": "string"},
    "from_number": {"type": "string"},
    "event_subject": {"type": "string"},
    "cpa_config": {"type": "string"},
    "dial_timeout": {"type": "integer", "minimum": 5},
    "call_params": {"type": "object"}
  },
 "required": [ "gateway", "to_number", "from_number", "tenant_id","event_subject" ]
}
```

Field Name	Description	Default	Example
gateway	Logical name of the gateway. If no logical mapping is found, the field is assumed to be the actual gateway name.	NONE	chime-1
tenant_id	Billing entity	NONE	acme
to_number	Destination phone number	NONE	+918989898989
from_number	Caller ID Number	NONE	+919999999999
event_subject	NATS subject where call events are to be sent	NONE	botstream.result
cpa_config	If CPA is required, specify the previously configured va_config_name.	NONE	my-cpa-1
dial_timeout	Timeout for the dialing	30	40
call_params	Any valid json object. This will be returned as part of every call event. Keep this object small, otherwise it results in too much data traffic in NATS	NONE	{“client_id”: “1234”}
channel_vars	List of channel variables including SIP headers. Variables with prefix sip_h_X- will set the SIP headers	NONE	{"park_after_bridge": "true", "sip_h_X-company": "acme"}

Upon receiving, the BotStream will respond with the following reply:


```
{
   "call_uuid": "71fe427d-b3ef-4b0e-8ec5-2270f0dfe41c",
   "dialer_pod_id": "botstream-847584",
   "request_name": "request_make_call",
   "status": "OK",
   "status_code": 0
}
```


If there are errors, the status_code will contain the error code and the status field will have the error message. The full list of error codes can be found in the appendix.

In addition, the subject specified in the event_subject field will receive many event messages like started, ringing, answered, originate, hangup and call_quality. Some examples shown here:


```
{
   "event_name": "botstream::started",
   "event_data": {
      "call_params": {
         "client_id": "1234"
      },
      "call_uuid": "538ec253-af8e-4f35-af00-a9430130665e",
      "tenant_id": "acme",
      "timestamp": "2022-04-24T12:01:54.889974Z"
   }
}
{
   "event_name": "botstream::ringing",
   "event_data": {
      "call_params": {
         "client_id": "1234"
      },
      "call_uuid": "538ec253-af8e-4f35-af00-a9430130665e",
      "tenant_id": "acme",
      "timestamp": "2022-04-24T12:01:54.919976Z"
   }
}
{
   "event_name": "botstream::originate",
   "event_data": {
      "call_params": {
         "client_id": "1234"
      },
```

```
      "call_uuid": "538ec253-af8e-4f35-af00-a9430130665e",
      "event_subject": "botstream.result",
      "from_number": "10000",
      "gateway": "user",
      "status": "OK",
      "tenant_id": "acme",
      "to_number": "1001",
      "timestamp": "2022-04-24T12:01:59.543139Z"
   }
}
{
   "event_name": "botstream::answered",
   "event_data": {
      "call_params": {
         "client_id": "1234"
      },
      "call_type": "outbound",
      "call_uuid": "538ec253-af8e-4f35-af00-a9430130665e",
      "from_number": "10000",
      "tenant_id": "acme",
      "to_number": "1001",
      "timestamp": "2022-04-24T12:01:59.546106Z"
   }
}
{
   "event_name": "botstream::stream_started",
   "event_data": {
      "call_params": {
         "client_id": "1234"
      },
      "call_uuid": "538ec253-af8e-4f35-af00-a9430130665e",
      "tenant_id": "acme",
      "timestamp": "2022-04-24T12:01:59.718849Z"
   }
}
{
   "event_name": "botstream::stream_stopped",
   "event_data": {
      "call_uuid": "538ec253-af8e-4f35-af00-a9430130665e",
      "reason": "The operation completed successfully",
      "reason_code": 0,
      "timestamp": "2022-04-24T12:02:05.981935Z"
   }
}
```

```
{
    "event_name": "botstream::hangup",
    "event_data": {
        "call_params": {
            "client_id": "1234"
        },
        "call_uuid": "538ec253-af8e-4f35-af00-a9430130665e",
        "event_data": "NORMAL_CLEARING",
        "tenant_id": "acme",
        "timestamp": "2022-04-24T12:02:05.999967Z"
    }
}
{
    "event_name": "botstream::call_quality",
    "event_data": {
        "call_params": {
            "client_id": "1234"
        },
        "call_uuid": "538ec253-af8e-4f35-af00-a9430130665e",
        "mos": "4.50",
        "quality": "100.00",
        "tenant_id": "acme",
        "timestamp": "2022-04-24T12:02:06.001930Z"
    }
}
```


3.3 Drop Call

The API to drop either an inbound or outbound call.

Field event_name should be set to **request_drop_call**.

Field event_data should conform to the following schema:

```
{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "type": "object",
    "properties": {
      "call_uuid": {
                    "type": "string",
                    "pattern": "^[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}$"
                    }
    },
  "required": [ "call_uuid" ]
```

}

Field Name	Description	Default	Example
call_uuid	UUID of the call to be dropped	NONE	8288ef17-3ab9-4745-b6b5-699dcdf4d7cf

Upon success, the reply will be as follows:

```
{
        "request_name": "request_drop_call",
        "status": "OK",
        "status_code": 0
}
```

3.4 Set Channel Variables

The API to drop either an inbound or outbound call.

Field event_name should be set to **request_set_channel_vars**.

Field event_data should conform to the following schema:

```
{
   "$schema": "http://json-schema.org/draft-04/schema#",
   "type": "object",
   "properties": {
     "call_uuid": {
                 "type": "string",
                 "pattern": "^[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}$"
                 }
   },
  "required": [ "call_uuid" ]
}
```

Field Name	Description	Default	Example
call_uuid	UUID of the call to be dropped	NONE	8288ef17-3ab9-4745-b6b5-699dcdf4d7cf
channel_vars	List of channel variables including SIP headers. Variables with prefix	NONE	{"park_after_bridge": "true", "sip_h_X-company": "acme"}

	sip_h_X- will set the SIP headers		

Upon success, the reply will be as follows:

{
    "request_name": "request_set_channel_vars",
    "status": "OK",
    "status_code": 0
}

3.5 Start Streaming

The API to start streaming the audio from the customer side to the VoiceAI engine.

Field event_name should be set to **request_start_stream**.

Field event_data should conform to the following schema:

```
{
   "$schema": "http://json-schema.org/draft-04/schema#",
   "type": "object",
   "properties": {
     "call_uuid": {
                    "type": "string",
                    "pattern": "^[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}$"
              },
     "websocket_host": {"type": "string"}
     "websocket_port": {"type": "integer"}
   },
  "required": [ "call_uuid",, "websocket_host", "websocket_port" ]
}
```

Field Name	Description	Default	Example
call_uuid	UUID of the call	NONE	8288ef17-3ab9-4745-b6b5-699dcdf4d7cf
websocket_host	VoiceAI websocket host	NONE	wss://voiceai.com
websocket_port	VoiceAI websocket port	NONE	8000

Only secure connections are allowed, even self-signed server certificates will work. This API request will open a websocket connection to the VoiceAI address provided and then send the following json object with call_uuid in text mode.

```
{
    "call_uuid": "8288ef17-3ab9-4745-b6b5-699dcdf4d7cf"
}
```

After sending the above json object, the web socket will switch to binary mode and start streaming the audio from the customer side continuously until the socket is closed.

Upon success, the reply will be as follows:

```
{
      "request_name": "request_start_stream",
      "status": "OK",
      "status_code": 0
}
```

The VoiceAI can send audio on the web socket to BotStream, and it will be played to the customer. The audio format in both directions should be signed 16bit PCM.

3.6 Stop Streaming

The API to stop streaming a call to Voice AI.

Field event_name should be set to **request_stop_stream**.

Field event_data should conform to the following schema:

```
{
   "$schema": "http://json-schema.org/draft-04/schema#",
   "type": "object",
   "properties": {
     "call_uuid": {
                "type": "string",
                "pattern": "^[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}$"
                }
   },
  "required": [ "call_uuid" ]
}
```

Field Name	Description	Default	Example
call_uuid	UUID of the call	NONE	8288ef17-3ab9-4745-b6b5-69

			9dcdf4d7cf

Upon success, the reply will be as follows:


```
{
	"request_name": "request_stop_stream",
	"status": "OK",
	"status_code": 0
}
```


3.7 Stop Play

The API to stop playing audio to the customer. This only affects currently playing audio. It doesn't stop any new audio data sent from the VoiceAI engine from playing. Don't call this API while continuously sending audio to play.

Field event_name should be set to **request_stop_play**.

Field event_data should conform to the following schema:

```
{
   "$schema": "http://json-schema.org/draft-04/schema#",
   "type": "object",
   "properties": {
     "call_uuid": {
		"type": "string",
		"pattern": "^[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}$"
		}
   },
  "required": [ "call_uuid" ]
}
```

Field Name	Description	Default	Example
call_uuid	UUID of the call	NONE	8288ef17-3ab9-4745-b6b5-699dcdf4d7cf

Upon success, the reply will be as follows:


```
{
	"request_name": "request_drop_call",
	"status": "OK",
	"status_code": 0
}
```

3.8 Bridge Call

The API request to bridge the call to an agent, with an option to stop streaming.

Field event_name should be set to **request_bridge_call**.

Field event_data should conform to the following schema:


```
{
   "$schema": "http://json-schema.org/draft-04/schema#",
   "type": "object",
   "properties": {
     "call_uuid": {
                    "type": "string",
                    "pattern": "^[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}$"
               },
     "gateway": {"type": "string"},
     "to_number": {"type": "string"},
     "from_number": {"type": "string"},
     "stop_stream": {"type": "boolean", "enum" : [true,false] },
     "dial_timeout": {"type": "integer", "minimum": 5}
   },
  "required": [ "gateway", "to_number", "from_number", "call_uuid", "stop_stream" ]
}
```


Field Name	Description	Default	Example
call_uuid	UUID of the call	NONE	8288ef17-3ab9-4745-b6b5-699dcdf4d7cf
gateway	Logical name of the gateway. If no logical mapping is found, the field is assumed to be the actual gateway name.	NONE	10001
to_number	Destination phone number	NONE	+918989898989
from_number	Caller ID Number	NONE	+919999999999
stop_stream	Stop streaming upon bridging	NONE	true
dial_timeout	Timeout for the dialing	30	40
channel_vars	List of channel variables including SIP	NONE	{"park_after_bridge":

	headers. Variables with prefix sip_h_X- will set the SIP headers		"true", "sip_h_X-company": "acme"}

Upon success, the reply will be as follows:

```
{
        "request_name": "request_bridge_call",
        "status": "OK",
        "status_code": 0
}
```

The call resulting from the Bridge call API will have the call_type as bridge.

3.9 Transfer Call

The API to transfer the call.

Field event_name should be set to **request_transfer_call**.

Field event_data should conform to the following schema:

```
{
   "$schema": "http://json-schema.org/draft-04/schema#",
   "type": "object",
   "properties": {
     "call_uuid": {"type": "string",
                   "pattern": "^[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}$"
                 },
     "destination": {"type": "string"}
   },
  "required": [ "call_uuid", "destination" ]
}
```

Field Name	Description	Default	Example
call_uuid	UUID of the call to be transferred	NONE	8288ef17-3ab9-4745-b6b5-699dcdf4d7cf
destination	Destination of the call, as configured in the dial plan.	NONE	538ec253-af8e-4f35-af00-a9430130665e

Upon success, the reply will be as follows:

{

"request_name": "request_transfer_call",
"status": "OK",
"status_code": 0
}

3.10 Snoop on Call

The API to snoop on a call from a SIP phone.

Field event_name should be set to **request_snoop_on_call**.

Field event_data should conform to the following schema:

```
{
   "$schema": "http://json-schema.org/draft-04/schema#",
   "type": "object",
   "properties": {
     "target_call_uuid": {"type": "string",
                  "pattern": "^[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}$"},
     "listener_call_uuid": {"type": "string",
                  "pattern": "^[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}$"}
   },
  "required": [ "target_call_uuid", "listener_call_uuid" ]
}
```

Field Name	Description	Default	Example
target_call_uuid	UUID of the call to be snooped	NONE	8288ef17-3ab9-4745-b6b5-699dcdf4d7cf
listener_call_uuid	Destination of the call, as configured in the dial plan.	NONE	

Upon success, the reply will be as follows:

{
"request_name": "request_snoop_on_call",
"status": "OK",
"status_code": 0
}

3.11 Add SIP Gateway Connection

Add a SIP gateway connection to point to a SIP gateway like Chime, Twilio or E1 Voip gateway. The gateway information will be saved in the BotStream servers. If the BotStream is on virtual machines, the VoiceAI application should make sure all the SIP gateways are added upon launching a new VM. In the case of physical servers, the SIP gateway once created will persist forever.

Field event_name should be set to **request_add_sip_gateway**.

Field event_data should conform to the following schema:

```
{
   "$schema": "http://json-schema.org/draft-04/schema#",
   "type": "object",
   "properties": {
     "name": {"type": "string"},
     "params": {"type": "object" },
     "variables": {"type": "object"}
   },
  "required": [ "name", "params" ]
}
```

Example:

```
{
   "name": "gty1",
   "params": {
      "username": "acme_cust",
      "password": "djhfund",
      "realm": "172.16.17.12",
      "proxy": "172.16.17.12;transport=udp",
      "from-user": "+14343220993",
      "from-domain": "172.16.7.2",
      "register-transport": "udp",
      "register": "false",
      "auth-calls": "false",
      "caller-id-in-from": "true"
   },
   "variables": {
      "rtp_secure_media": "false",
      "sip_h_X-Name": "lab_system"
   }
}
```

Field Name	Description	Default	Example
name	Name of the gateway	NONE	customer
params	This object will have the various fields used by Freeswitch	NONE	Check this link at Freeswitch for examples of param fields.
variables	This object will have the variables that will be passed to the other end while making the calls.	NONE	"rtp_secure_media": "true"

Upon success, the reply will be as follows:


```
{
        "request_name": "request_add_sip_gateway",
        "status": "OK",
        "status_code": 0
}
```


To make any changes to an already existing SIP gateway, call the same request with new values. Any param with an empty string will be removed from the existing list.

3.12 Get SIP Gateway List

Get either a specific gateway or all the botstream gateways defined in the setup.

Field event_name should be set to **request_get_sip_gateway**.

Field event_data should conform to the following schema:

```
{
   "$schema": "http://json-schema.org/draft-04/schema#",
   "type": "object",
   "properties": {
     "name": {"type": "string"}
   },
  "required": [ "name" ]
}
```

Field Name	Description	Default	Example
name	Name of the gateway	NONE	customer

Upon success, the reply will be as follows:

```
{
        "request_name": "request_get_sip_gateway",
        "status": "OK",
        "status_code": 0,
        "sip_gateways": [...
        ]
}
```

3.13 Delete SIP Gateway

Define a SIP gateway connection to point to a SIP gateway like Chime, Twilio or E1 Voip gateway.

Field event_name should be set to **request_delete_sip_gateway**.

Field event_data should conform to the following schema:

```
{
   "$schema": "http://json-schema.org/draft-04/schema#",
   "type": "object",
   "properties": {
     "name": {"type": "string"}
   },
  "required": [ "name" ]
}
```

Field Name	Description	Default	Example
name	Name of the gateway	NONE	customer

Upon success, the reply will be as follows:

```
{
```

```
        "request_name": "request_delete_sip_gateway",
        "status": "OK",
        "status_code": 0
}
```

3.14 Add Logical Gateway Mapping

BotStream allows users to define logical gateways, which can be mapped to multiple SIP gateways. By using a logical gateway, outbound dialing can be distributed across multiple SIP gateways by weightage and one could also apply max limits if required.

Field event_name should be set to **request_add_logical_gateway**.

Field event_data should conform to the following schema:

```
{
   "$schema": "http://json-schema.org/draft-04/schema#",
   "type": "object",
   "properties": {"name": {"type": "string" },
     "gateways": { "type": "array",
       "items": [
         {
           "type": "object",
           "properties": {"name": {"type": "string" }, "weightage": { "type": "integer"},"max_limit":
{"type": "integer" } },
           "required": ["name", "weightage"]
         }
       ]
     }
   },
   "required": ["name","gateways"]
 }
```

Example:

```
{
   "name": "logical_gty1",
   "gateways":[
      {"name":"gateway_A","weightage":10,"max_limit": 100},
      {"name":"gateway_B","weightage":20},
      {"name":"gateway_C","weightage":10,"max_limit": 200}
   ]
}
```

In the above example, if 40 calls are dialed, gateway_A will get 10, gateway_B will get 20 and gateway_C will get 10. This uses a stochastic distribution algorithm and not a deterministic one. Therefore, these will be an approximate distribution.

The API will not validate if the SIP gateways actually exist. BotStream doesn't have a persistence database, therefore these logical gateway mappings will have to be created every time the BotStream installation is restarted.

Field Name	Description	Default	Example
name	Logical name of the gateway	NONE	customer
gateways[n].name	Name of the sip gateway to be mapped to the logical gateway	NONE	Should be one of the pre-defined SIP gateway
gateways[n].weightage	This object will have the variables that will be passed to the other end while making the calls.	NONE	"rtp_secure_media": "true"
gateways[n].max_limit	Absolute limit of the number of calls the SIP gateway can handle	NA	100

To make any changes to an already existing SIP gateway, call the same request with new values, with more and less SIP gateways in the list.

3.15 Delete Logical Gateway Mapping

Remove a logical gateway from the BotStream installation.

Field event_name should be set to **request_delete_logical_gateway**.

Field event_data should conform to the following schema:

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
```

    "properties": {
      "name": {"type": "string"}
    },
  "required": ["name"]
}

Field Name	Description	Default	Example
name	Name of the logical gateway	NONE	customer

Upon success, the reply will be as follows:

{
    "request_name": "request_delete_logical_gateway",
    "status": "OK",
    "status_code": 0
}

4 Special Events

These are events that are unrelated to the API.

4.1 DTMF

Whenever someone at the telephony end presses a key, the following event will be generated:

{
    "event_name": "botstream::dtmf",
    "event_data": {
        "call_params": {
            "client_id": "1234"
        },
        "call_uuid": "da2e2563-376c-40b0-a5c0-713314902876",
        "digit": "4",
        "tenant_id": "acme",
        "timestamp": "2022-04-22T16:44:00.121903Z"
    }
}

4.2 Inbound Call event

When BotStream accepts an inbound call, it starts with an answered event, with call_type as inbound. Rest of the process will be similar.


```
{
  "event_name": "botstream::answered",
  "event_data": {
    "call_params": {
      "client_id": "1234"
    },
    "call_type": "inbound",
    "call_uuid": "538ec253-af8e-4f35-af00-a9430130665e",
    "from_number": "1001",
    "tenant_id": "acme",
    "to_number": "5000",
    "timestamp": "2022-04-24T12:01:59.546106Z"
  }
}
```


5 Appendix A: Change Log

5.1 Version 1.0.0 (25th April 2022)

1. BotStream created after forking from iraDialer, to target VoiceAI applications.

5.2 Version 1.0. (8th June 2022)

1. Web socket closing issues at high volume fixed.
2. New license schemes applied (sleg/cleg/dt/amd).
3. Added trunk management APIs.
4. Added channel variables and SIP headers API.

6 Appendix B: NATS Events

Web Event	Description
originate	Dialing has been completed
started	Call has been initiated

ringing	Call is ringing
answered	Call is answered
hangup	Call is hang-up
cpa_detection	The CPA engine has detected a configured pattern
cpa_done	CPA process has ended
dtmf	The customer has pressed a key.
error	Some error has occurred, check status_code field for the details
call_quality	Sent after the call ends, with mos and quality value
stream_started	Websocket has been opened and streaming has started
stream_stopped	Websocket is closed and streaming has ended

7 Appendix C : Status Code of Errors

When the event contains a status_code, it means an error has occurred. The status field will contain the detailed description. The status code will be one of the following:

Code	Status Code
100	Can't pick up CPA configuration. This happens if the dialer can't find a matching CPA configuration for the call.
101	Could not initialize the analyser specified in the CPA configuration.
102	CPA/Stream initialization failed! This happens if RTP listener cannot be started.
107	The call_uuid mentioned in the API doesn't exist.
108	The call_uuid mentioned in the API doesn't exist anymore.
110	All licenses have expired.
111	Tenant doesn't have any licenses installed.
112	Unable to use the licenses.
113	Unknown error while trying to acquire license.

114	Call dialing failed. Check status for more details.
115	Call failed to originate, could be a throttling problem.
127	Error stopping play.
128	Invalid request.
129	Not able to set the CPA configuration
130	Not all parameters specified.
131	Unable to add a new call.
134	Incorrectly formed json request
135	Request is not according to the schema
139	API is not yet implemented
140	Error setting channel variables
141	Error in SIP gateway configuration
142	Error deleting SIP gateway
143	Error retrieving SIP gateway information
144	Stream already initiated for the call

8 Appendix D: Hangup Causes

ITU-T Q.850 Code	**SIP Equiv.**	**Enumeration**	**Cause**	**Description**
0		*UNSPECIFIED*	Unspecified. No other cause codes applicable.	This is usually given by the router when none of the other codes apply. This cause usually occurs in the same type of situations as cause 1, cause 88, and cause 100.
1	404	*UNALLOCATED_NUMBER*	Unallocated (unassigned) number [Q.850 value 1]	This cause indicates that the called party cannot be reached because, although the called party number is in a valid format, it is not currently allocated (assigned).

2	404	*NO_ROUTE_TRANSIT_NET*	No route to specified transit network (national use) [Q.850]	This cause indicates that the equipment sending this cause has received a request to route the call through a particular transit network, which it does not recognize. The equipment sending this cause does not recognize the transit network either because the transit network does not exist or because that particular transit network, while it does exist, does not serve the equipment which is sending this cause.
3	404	*NO_ROUTE_DESTINATION*	No route to destination [Q.850]	This cause indicates that the called party cannot be reached because the network through which the call has been routed does not serve the destination desired. This cause is supported on a network dependent basis.
6		*CHANNEL_UNACCEPTABLE*	channel unacceptable [Q.850]	This cause indicates that the channel most recently identified is not acceptable to the sending entity for use in this call.
7		*CALL_AWARDED_DELIVERED*	call awarded, being delivered in an established channel [Q.850]	This cause indicates that the user has been awarded the incoming call, and that the incoming call is being connected to a channel already established to that user for similar calls (e.g. packet-mode x.25 virtual calls).
16		*NORMAL_CLEARING*	normal call clearing [Q.850]	This cause indicates that the call is being cleared because one of the users involved in the call has requested that the call be cleared. Under normal situations, the source of this cause is not the network.
17	486	*USER_BUSY*	user busy [Q.850]	This cause is used to indicate that the called party is unable to accept another call because the user busy condition has been encountered. This cause value may be generated by the called user or by the network. In the case of user determined user busy it is noted that the user equipment is compatible with the call.
18	408	*NO_USER_RESPONSE*	no user responding [Q.850]	This cause is used when a called party does not respond to a call establishment message with either an alerting or connect indication within the prescribed period of time allocated.
19	480	*NO_ANSWER*	no answer from user (user alerted) [Q.850]	This cause is used when the called party has been alerted but does not respond with a connect indication within a prescribed period of time. Note - This cause is not necessarily generated by Q.931 procedures but may be generated by internal network timers.
20	480	*SUBSCRIBER_ABSENT*	subscriber absent [Q.850]	This cause value is used when a mobile station has logged off, radio contact is not obtained with a mobile

				station or if a personal telecommunication user is temporarily not addressable at any user-network interface. Sofia SIP will normally raise USER_NOT_REGISTERED in such situations.
21	603	*CALL_REJECTED*	call rejected [Q.850]	This cause indicates that the equipment sending this cause does not wish to accept this call, although it could have accepted the call because the equipment sending this cause is neither busy nor incompatible. The network may also generate this cause, indicating that the call was cleared due to a supplementary service constraint. The diagnostic field may contain additional information about the supplementary service and reason for rejection.
22	410	*NUMBER_CHANGED*	number changed [Q.850]	This cause is returned to a calling party when the called party number indicated by the calling party is no longer assigned, The new called party number may optionally be included in the diagnostic field. If a network does not support this cause, cause no: 1, unallocated (unassigned) number shall be used.
23	410	*REDIRECTION_TO_NEW_DESTINATION*		This cause is used by a general ISUP protocol mechanism that can be invoked by an exchange that decides that the call should be set-up to a different called number. Such an exchange can invoke a redirection mechanism, by use of this cause value, to request a preceding exchange involved in the call to route the call to the new number.
25	483	*EXCHANGE_ROUTING_ERROR*		This cause indicates that the destination indicated by the user cannot be reached, because an intermediate exchange has released the call due to reaching a limit in executing the hop counter procedure. This cause is generated by an intermediate node, which when decrementing the hop counter value, gives the result 0.
27	502	*DESTINATION_OUT_OF_ORDER*	destination out of order [Q.850]	This cause indicates that the destination indicated by the user cannot be reached because the interface to the destination is not functioning correctly. The term "not functioning correctly" indicates that a signal message was unable to be delivered to the remote party; e.g. a physical layer or data link layer failure at the remote party, or user equipment off-line.
28	484	*INVALID_NUMBER_FORMAT*	invalid number format (address	This cause indicates that the called party cannot be reached because the called party number is not in a valid format or is not complete.

			incomplete) [Q.850]	
29	501	*FACILITY_REJECTED*	facilities rejected [Q.850]	This cause is returned when a supplementary service requested by the user cannot be provide by the network.
30		*RESPONSE_TO_STATUS_ENQUIRY*	response to STATUS INQUIRY [Q.850]	This cause is included in the STATUS message when the reason for generating the STATUS message was the prior receipt of a STATUS INQUIRY.
31	480	*NORMAL_UNSPECIFIED*	normal, unspecified [Q.850]	This cause is used to report a normal event only when no other cause in the normal class applies.
34	503	*NORMAL_CIRCUIT_CONGESTION*	no circuit/channel available [Q.850]	This cause indicates that there is no appropriate circuit/channel presently available to handle the call.
38	503	*NETWORK_OUT_OF_ORDER*	network out of order [Q.850]	This cause indicates that the network is not functioning correctly and that the condition is likely to last a relatively long period of time e.g. immediately re-attempting the call is not likely to be successful.
41	503	*NORMAL_TEMPORARY_FAILURE*	temporary failure [Q.850]	This cause indicates that the network is not functioning correctly and that the condition is not likely to last a long period of time; e.g. the user may wish to try another call attempt almost immediately.
42	503	*SWITCH_CONGESTION*	switching equipment congestion [Q.850]	This cause indicates that the switching equipment generating this cause is experiencing a period of high traffic.
43		*ACCESS_INFO_DISCARDED*	access information discarded [Q.850]	This cause indicates that the network could not deliver access information to the remote user as requested, i.e. user-to-user information, low layer compatibility, high layer compatibility or sub-address as indicated in the diagnostic. It is noted that the particular type of access information discarded is optionally included in the diagnostic.
44	503	*REQUESTED_CHAN_UNAVAIL*	requested circuit/channel not available [Q.850]	This cause is returned when the other side of the interface cannot provide the circuit or channel indicated by the requesting entity.
45		*PRE_EMPTED*		

47			resource unavailable, unspecified [Q.850]	This cause is used to report a resource unavailable event only when no other cause in the resource unavailable class applies.
50		*FACILITY_NOT_SUBSCRIBED*	requested facility not subscribed [Q.850	This cause indicates that the user has requested a supplementary service, which is available, but the user is not authorized to use.
52	403	*OUTGOING_CALL_BARRED*	outgoing calls barred	This cause indicates that although the calling party is a member of the CUG for the outgoing CUG call, outgoing calls are not allowed for this member of the CUG.
54	403	*INCOMING_CALL_BARRED*	incoming calls barred	This cause indicates that although the called party is a member of the CUG for the incoming CUG call, incoming calls are not allowed to this member of the CUG.
57	403	*BEARERCAPABILITY_NOTAUTH*	bearer capability not authorized [Q.850]	This cause indicates that the user has requested a bearer capability that is implemented by the equipment which generated this cause but the user is not authorized to use.
58	503	*BEARERCAPABILITY_NOTAVAIL*	bearer capability not presently available [Q.850]	This cause indicates that the user has requested a bearer capability which is implemented by the equipment which generated this cause but which is not available at this time.
63		*SERVICE_UNAVAILABLE*	service or option not available, unspecified [Q.850]	This cause is used to report a service or option not available event only when no other cause in the service or option not available class applies.
65	488	*BEARERCAPABILITY_NOTIMPL*	bearer capability not implemented [Q.850]	This cause indicates that the equipment sending this cause does not support the bearer capability requested.
66		*CHAN_NOT_IMPLEMENTED*	channel type not implemented [Q.850]	This cause indicates that the equipment sending this cause does not support the channel type requested
69	501	*FACILITY_NOT_IMPLEMENTED*	requested facility not implemented [Q.850]	This cause indicates that the equipment sending this cause does not support the requested supplementary services.
79	501	*SERVICE_NOT_IMPLEMENTED*	service or option not implemented, unspecified [Q.850]	This cause is used to report a service or option not implemented event only when no other cause in the service or option not implemented class applies.

81		*INVALID_CALL_REFERENCE*	invalid call reference value [Q.850]	This cause indicates that the equipment sending this cause has received a message with a call reference which is not currently in use on the user-network interface.
88	488	*INCOMPATIBLE_DESTINATION*	incompatible destination [Q.850]	This cause indicates that the equipment sending this cause has received a request to establish a call which has low layer compatibility, high layer compatibility or other compatibility attributes (e.g. data rate) which cannot be accommodated.
95		*INVALID_MSG_UNSPECIFIED*	invalid message, unspecified [Q.850]	This cause is used to report an invalid message event only when no other cause in the invalid message class applies.
96		*MANDATORY_IE_MISSING*	mandatory information element is missing [Q.850]	This cause indicates that the equipment sending this cause has received a message which is missing an information element which must be present in the message before that message can be processed.
97		*MESSAGE_TYPE_NONEXIST*	message type non-existent or not implemented [Q.850]	This cause indicates that the equipment sending this cause has received a message with a message type it does not recognize either because this is a message not defined of defined but not implemented by the equipment sending this cause.
98		*WRONG_MESSAGE*	message not compatible with call state or message type non-existent or not implemented. [Q.850]	This cause indicates that the equipment sending this cause has received a message such that the procedures do not indicate that this is a permissible message to receive while in the call state, or a STATUS message was received indicating an incompatible call state.
99		*IE_NONEXIST*	Information element / parameter non-existent or not implemented [Q.850]	This cause indicates that the equipment sending this cause has received a message which includes information element(s)/parameter(s) not recognized because the information element(s)/parameter name(s) are not defined or are defined but not implemented by the equipment sending the cause. This cause indicates that the information element(s)/parameter(s) were discarded. However, the information element is not required to be present in the message in order for the equipment sending the cause to process the message.
100		*INVALID_IE_CONTENTS*	Invalid information	This cause indicates that the equipment sending this cause has received and information element which it has

			element contents [Q.850]	implemented; however, one or more fields in the I.E. are coded in such a way which has not been implemented by the equipment sending this cause.
101		*WRONG_CALL_STATE*	message not compatible with call state [Q.850]	This cause indicates that a message has been received which is incompatible with the call state.
102	504	*RECOVERY_ON_TIMER_EXPIRE*	recovery on timer expiry [Q.850]	This cause indicates that a procedure has been initiated by the expiration of a timer in association with error handling procedures. This is often associated with NAT problems. Ensure that "NAT Mapping Enable" is turned on in your ATA. If it is not NAT related it can sometimes be provider related, make sure to ensure another outbound provider does not solve the problem.
103		*MANDATORY_IE_LENGTH_ERROR*	parameter non-existent or not implemented - passed on (national use) [Q.850]	This cause indicates that the equipment sending this cause has received a message which includes parameters not recognized because the parameters are not defined or are defined but not implemented by the equipment sending this cause. The cause indicates that the parameter(s) were ignored. In addition, if the equipment sending this cause is an intermediate point, then this cause indicates that the parameter(s) were passed unchanged.
111		*PROTOCOL_ERROR*	protocol error, unspecified [Q.850]	This cause is used to report a protocol error event only when no other cause in the protocol error class applies.
127		*INTERWORKING*	Interworking, unspecified [Q.850]	This cause indicates that an interworking call (usually a call to SW56 service) has ended.
487	487	*ORIGINATOR_CANCEL*		
500		*CRASH*		
501		*SYSTEM_SHUTDOWN*		
502		*LOSE_RACE*		
503		*MANAGER_REQUEST*		This cause is used when you send an api command to make it hangup. For example uuid_kill <uuid>
600		*BLIND_TRANSFER*		

601		*ATTENDED_TRANSFER*		
602		*ALLOTTED_TIMEOUT*		This cause means that the server canceled the call because the destination channel took too long to answer.
603		*USER_CHALLENGE*		
604		*MEDIA_TIMEOUT*		
605		*PICKED_OFF*		This cause means the call was picked up by intercepting it from another extension (i.e. dialing **ext_number from another extension).
606		*USER_NOT_REGISTERED*		This means you tried to originate a call to a SIP user who forgot to register.
607		*PROGRESS_TIMEOUT*		
609		*GATEWAY_DOWN*		Gateway is down (not answering on OPTIONS or SUBSCRIBE)