Powered by Universal Speech Solutions LLC

UniMRCP

Google SR Plugin

Usage Guide

Revision: 2

Created: May 17, 2017

Last updated: June 19, 2017

Author: Arsen Chaloyan

Table of Contents

1 Overview

This guide describes how to configure and use the Google Speech Recognition (GSR) plugin to the UniMRCP server. The document is intended for users having a certain knowledge of Google Cloud Speech Platform and UniMRCP.

1.1 Installation

For installation instructions, use one of the guides below.

- RPM Package Installation (Red Hat / Cent OS)
- Deb Package Installation (Debian / Ubuntu)

1.2 Applicable Versions

Instructions provided in this guide are applicable to the following versions.

UniMRCP 1.4.0 and above
UniMRCP GSR Plugin 1.0.0 and above

2 Supported Features

This is a brief check list of the features currently supported by the UniMRCP server running with the GSR plugin.

2.1 MRCP Methods

- ✓ DEFINE-GRAMMAR
- ✓ RECOGNIZE
- ✓ START-INPUT-TIMERS
- ✓ STOP
- ✓ SET-PARAMS
- ✓ GET-PARAMS

2.2 MRCP Events

- ✓ RECOGNITION-COMPLETE
- ✓ START-OF-INPUT

2.3 MRCP Header Fields

- ✓ Input-Type
- ✓ No-Input-Timeout
- ✓ Recognition-Timeout
- ✓ Waveform-URI
- ✓ Media-Type
- ✓ Completion-Cause
- ✓ Confidence-Threshold
- ✓ Start-Input-Timers
- ✓ DTMF-Interdigit-Timeout
- ✓ DTMF-Term-Timeout
- ✓ DTMF-Term-Char
- ✓ Save-Waveform
- ✓ Speech-Language
- ✓ Cancel-If-Queue
- ✓ Sensitivity-Level

2.4 Grammars

- ✓ Built-in and dynamic speech contexts
- ✓ Built-in/embedded DTMF grammar

2.5 Results

- ✓ NLSML

3 Configuration Format

The configuration file of the GSR plugin is located in */opt/unimrcp/conf/umsgsr.xml*. The configuration file is written in XML.

3.1 Document

The root element of the XML document must be *<umsgsr>*.

Attributes

Name	Unit	Description
license-file	File path	Specifies the license file. File name may include patterns containing '*' sign. If multiple files match the pattern, the most recent one gets used.
gapp-credentials-file	File path	Specifies the Google Application Credentials file to use. File name may include patterns containing '*' sign. If multiple files match the pattern, the most recent one gets used.

Parent

None.

Children

Name	Unit	Description
<streaming-recognition>	String	Specifies parameters of streaming recognition employed via gRPC.
< speech-dtmf-input-detector>	String	Specifies parameters of the speech and DTMF input detector.
< utterance-manager>	String	Specifies parameters of the utterance manager.

Example

This is an example of a bare document.

```
< umsgsr license-file="umsgsr_*.lic" gapp-credentials-file="*.json">
```

```
</ umsgsr>
```

3.2 Streaming Recognition

This element specifies parameters of streaming recognition.

Attributes

Name	Unit	Description
language	String	Specifies the default language to use, if not set by the client. For a list of supported languages, visit https://cloud.google.com/speech/docs/languages
interim-results	Boolean	Specifies whether to request interim results or not.
max-alternatives	Integer	Specifies the maximum number of speech recognition result alternatives to be returned. Can be overridden by client by means of the header field *N-Best-List-Length*.

Parent

<umsgsr>

Children

None.

Example

This is an example of streaming recognition element.

```
<streaming-recognition
   interim-results="false"
   language="en-US"
   max-alternatives="1"
/>
```

3.3 Speech Contexts

This element specifies a list of speech contexts, available since GSR 1.1.0.

Attributes

None.

Parent

<umsgsr>

Children

<speech-context>

Example

The example below defines two speech contexts *booking* and *directory*.

```
<speech-contexts>
  <speech-context id="booking" enable="true">
    <phrase>I would like to book a flight from New York to Rome with a ticket eligible for
free cancellation</phrase>
    <phrase>I would like to book a one-way flight from New York to Rome</phrase>
  </speech-context>

  <speech-context id="directory" enable="true">
    <phrase>call Steve</phrase>
    <phrase>call John</phrase>
    <phrase>dial 5</phrase>
    <phrase>dial 6</phrase>
  </speech-context>
</speech-contexts>
```

3.4 Speech Context

This element specifies a speech context, available since GSR 1.1.0.

Attributes

Name	Unit	Description
id	String	Specifies a unique string identifier of the speech context to be referenced by the MRCP client.
enable	Boolean	Specifies whether the speech context is enabled or disabled.

Parent

<speech-contexts>

Children

None.

Example

This is an example of speech context element.

```
<speech-context id="directory" enable="true">
  <phrase>call Steve</phrase>
  <phrase>call John</phrase>
  <phrase>dial 5</phrase>
  <phrase>dial 6</phrase>
</speech-context>
```

3.5 Speech and DTMF Input Detector

This element specifies parameters of the speech and DTMF input detector.

Attributes

Name	Unit	Description
vad-mode	Integer	Specifies an operating mode of VAD in the range of [0 ... 3]. Default is 1. Available since GSR 1.1.0.
speech-start-timeout	Time interval [msec]	Specifies how long to wait in transition mode before triggering a start of speech input event.
speech-complete-timeout	Time interval [msec]	Specifies how long to wait in transition mode before triggering an end of speech input event.
noinput-timeout	Time interval [msec]	Specifies how long to wait before triggering a no-input event.
input-timeout	Time interval [msec]	Specifies how long to wait for input to complete.
dtmf-interdigit-timeout	Time interval [msec]	Specifies a DTMF inter-digit timeout.
dtmf-term-timeout	Time interval [msec]	Specifies a DTMF input termination timeout.
dtmf-term-char	Character	Specifies a DTMF input termination character.
speech-leading-silence	Time interval [msec]	Specifies desired silence interval preceding spoken input.

speech-trailing-silence	Time interval [msec]	Specifies desired silence interval following spoken input.
speech-output-period	Time interval [msec]	Specifies an interval used to send speech frames to the recognizer.

Parent

<umsgsr>

Children

None.

Example

The example below defines a typical speech and DTMF input detector having the default parameters set.

```
<speech-dtmf-input-detector
  speech-start-timeout="300"
  speech-complete-timeout="1000"
  noinput-timeout="5000"
  input-timeout="10000"
  dtmf-interdigit-timeout="5000"
  dtmf-term-timeout="10000"
  dtmf-term-char=""
  speech-leading-silence="300"
  speech-trailing-silence="300"
  speech-output-period="200"
/>
```

3.6 Utterance Manager

This element specifies parameters of the utterance manager.

Attributes

Name	Unit	Description
save-waveforms	Boolean	Specifies whether to save waveforms or not.
waveform-base-uri	String	Specifies the base URI used to compose an absolute waveform URI.
waveform-folder	Dir path	Specifies a folder the waveforms should be

		stored in.
expiration-time	Time interval [min]	Specifies a time interval after expiration of which waveforms are considered outdated.
purge-waveforms	Boolean	Specifies whether to delete outdated waveforms or not.
purge-interval	Time interval [min]	Specifies a time interval used to periodically check for outdated waveforms.

Parent

<umsgsr>

Children

None.

Example

The example below defines a typical utterance manager having the default parameters set.

```
<utterance-manager
  save-waveforms="false"
  waveform-base-uri="http://localhost/utterances/"
  waveform-folder=""
  expiration-time="60"
  purge-waveforms="true"
  purge-interval="30"
/>
```

4 Configuration Steps

This section outlines common configuration steps.

4.1 Using Default Configuration

The default configuration should be sufficient for the general use.

4.2 Specifying Recognition Language

Recognition language can be specified by the client per MRCP session by means of the header field *Speech-Language* set in a *SET-PARAMS* or *RECOGNIZE* request. Otherwise, the parameter *language* set in the configuration file *umsgsr.xml* is used. The parameter defaults to *en-US*.
For supported languages and their corresponding codes, visit the following link.

https://cloud.google.com/speech/docs/languages

4.3 Specifying Sampling Rate

Sampling rate is determined based on the SDP negotiation. Refer to the configuration guide of the UniMRCP server on how to specify supported encodings and sampling rates to be used in communication between the client and server.
The native sampling rate with the linear16 audio encoding is used in gRPC streaming to the Google Cloud Speech service.

4.4 Specifying Speech Input Parameters

While the default parameters specified for the speech input detector are sufficient for the general use, various parameters can be adjusted to better suite a particular requirement.

- speech-start-timeout

  This parameter is used to trigger a start of speech input. The shorter is the timeout, the sooner a *START-OF-INPUT* event is delivered to the client. However, a short timeout may also lead to a false positive.

- speech-complete-timeout

  This parameter is used to trigger an end of speech input. The shorter is the timeout, the shorter is the response time. However, a short timeout may also lead to a false positive.
  Note that both events, an expiration of the speech complete timeout and an *END-OF-SINGLE-UTTERANCE* response delivered from the Google Cloud Speech service, are monitored to trigger an end of speech input, on whichever comes first basis. In order to rely solely on an event delivered from the speech service, the parameter *speech-complete-timeout* needs to be set to a higher value.

- vad-mode

This parameter is used to specify an operating mode of the Voice Activity Detector (VAD) within an integer range of [0 … 3]. A higher mode is more aggressive and, as a result, is more restrictive in reporting speech. The parameter can be overridden per MRCP session by setting the header field *Sensitivity-Level* in a *SET-PARAMS* or *RECOGNIZE* request. The following table shows how the *Sensitivity-Level* is mapped to the *vad-mode*.

Sensitivity-Level	Vad-Mode
[0.00 … 0.25)	0
[0.25 … 0.50)	1
[0.50 … 0.75)	2
[0.75 … 1.00]	3

4.5 Specifying DTMF Input Parameters

While the default parameters specified for the DTMF input detector are sufficient for the general use, various parameters can be adjusted to better suite a particular requirement.

- dtmf-interdigit-timeout

This parameter is used to set an inter-digit timeout on DTMF input. The parameter can be overridden per MRCP session by setting the header field *DTMF-Interdigit-Timeout* in a *SET-PARAMS* or *RECOGNIZE* request.

- dtmf-term-timeout

This parameter is used to set a termination timeout on DTMF input and is in effect when dtmf-term-char is set and there is a match for an input grammar. The parameter can be overridden per MRCP session by setting the header field *DTMF-Term-Timeout* in a *SET-PARAMS* or *RECOGNIZE* request.

- dtmf-term-char

This parameter is used to set a character terminating DTMF input. The parameter can be overridden per MRCP session by setting the header field *DTMF-Term-Char* in a *SET-PARAMS* or *RECOGNIZE* request.

4.6 Specifying No-Input and Recognition Timeouts

- noinput-timeout

This parameter is used to trigger a no-input event. The parameter can be overridden per MRCP session by setting the header field *No-Input-Timeout* in a *SET-PARAMS* or *RECOGNIZE* request.

- input-timeout

This parameter is used to limit input (recognition) time. The parameter can be overridden per MRCP session by setting the header field *Recognition-Timeout* in a *SET-PARAMS* or *RECOGNIZE* request.

4.7 Specifying Utterance Parameters

The default parameters specified for the utterance manager are sufficient for the general use. However, various parameters can be adjusted to better suite a particular requirement.

- save-waveforms

  Utterances can optionally be recorded and stored if the configuration parameter *save-waveforms* is set to true. The parameter can be overridden per MRCP session by setting the header field *Save-Waveforms* in a *SET-PARAMS* or *RECOGNIZE* request.

- waveform-base-uri

  This parameter specifies the base URI used to compose an absolute waveform URI returned in the header field *Waveform-Uri* in response to a RECOGNIZE request.

- waveform-folder

  This parameter specifies a path to the directory used to store waveforms in.

- expiration-time

  This parameter specifies a time interval in minutes after expiration of which waveforms are considered outdated.

- purge-waveforms

  This parameter specifies whether to delete outdated waveforms or not.

- purge-interval

  This parameter specifies a time interval in minutes used to check for outdated waveforms if *purge-waveforms* is set to true.

5 Recognition Grammars and Results

5.1 Using Built-in Speech Contexts

Pre-set built-in speech contexts can be referenced by the MRCP client in a RECOGNIZE request as follows:

```
builtin:speech/$id
```

Where *$id* is a unique string identifier of built-in speech context.

Speech contexts are defined in the configuration file *umsgsr.xml*. A speech context is assigned a unique string identifier and holds a list of phrases which can optionally be passed to the Google Cloud Speech service to improve the recognition accuracy.

Below is a definition of a sample speech context *directory*:

```
<speech-context id="directory">
  <phrase>call Steve</phrase>
  <phrase>call John</phrase>
  <phrase>dial 5</phrase>
  <phrase>dial 6</phrase>
</speech-context>
```

Which can be referenced in a RECOGNIZE request as follows:

```
builtin:speech/directory
```

For generic speech transcription, having no speech contexts defined, a pre-set identifier *transcribe* must be used.

```
builtin:speech/transcribe
```

Note that support for speech contexts has been added since GSR 1.1.0.

5.2 Using Dynamic Speech Contexts

The MRCP client can also dynamically specify a speech context either

- in a DEFINE-GRAMMAR request by further referencing the defined speech context in a

RECOGNIZE request using the session URI scheme

- or inline in a RECOGNIZE request

While composing a DEFINE-GRAMMAR or RECOGNIZER request containing speech context definition, the following should be considered.

- The value of the header field *Content-Id* must be used as a unique string identifier of the speech context being defined.
- The value of the header field *Content-Type* must be set to *application/xml*.
- The message body must contain a definition of the speech context, composed based on the XML format of the element <speech-context>, specified in the configuration file *umsgsr.xml*. Note that the unique identifier of the speech context is set based on the header field *Content-Id*, as opposed to the attribute *Id* when loading from configuration.

5.3 Using Built-in DTMF Grammars

Pre-set built-in DTMF grammars can be referenced by the MRCP client in a RECOGNIZE request as follows:

```
builtin:dtmf/$id
```

Where *$id* is a unique string identifier of the built-in DTMF grammar.

Note that only a DTMF grammar identifier *digits* is currently supported.

5.4 Retrieving Results

Results received from the Google Cloud Speech service are transformed to the NLSML format with no semantic interpretation performed and sent to the MRCP client in a *RECOGNITION-COMPLETE* event.

6 Usage Examples

6.1 Speech Recognition without Speech Context

This examples demonstrates how to perform speech recognition by using a RECOGNIZE request, having no speech contexts defined.

C->S:

```
MRCP/2.0 336 RECOGNIZE 1
Channel-Identifier: 6e1a2e4e54ae11e7@speechrecog
Content-Id: request1@form-level
Content-Type: text/uri-list
Cancel-If-Queue: false
No-Input-Timeout: 5000
Recognition-Timeout: 10000
Start-Input-Timers: true
Confidence-Threshold: 0.87
Save-Waveform: true
Content-Length: 25

builtin:speech/transcribe
```

S->C:

```
MRCP/2.0 83 1 200 IN-PROGRESS
Channel-Identifier: 6e1a2e4e54ae11e7@speechrecog
```

S->C:

```
MRCP/2.0 115 START-OF-INPUT 1 IN-PROGRESS
Channel-Identifier: 6e1a2e4e54ae11e7@speechrecog
Input-Type: speech
```

S->C:

```
MRCP/2.0 498 RECOGNITION-COMPLETE 1 COMPLETE
Channel-Identifier: 6e1a2e4e54ae11e7@speechrecog
Completion-Cause: 000 success
Waveform-Uri: <http://localhost/utterances/utter-6e1a2e4e54ae11e7-
1.wav>;size=20480;duration=1280
```

```
Content-Type: application/x-nlsml
Content-Length: 214

<?xml version="1.0"?>
<result>
  <interpretation grammar="builtin:speech/transcribe" confidence="0.88">
    <instance>call Steve</instance>
    <input mode="speech">call Steve</input>
  </interpretation>
</result>
```

6.2 Speech Recognition with Built-in Speech Context

This examples demonstrates how to perform speech recognition by using a RECOGNIZE request to reference a pre-set built-in speech context *directory* on the server.

C->S:

```
MRCP/2.0 335 RECOGNIZE 1
Channel-Identifier: 3ea18b9854af11e7@speechrecog
Content-Id: request1@form-level
Content-Type: text/uri-list
Cancel-If-Queue: false
No-Input-Timeout: 5000
Recognition-Timeout: 10000
Start-Input-Timers: true
Confidence-Threshold: 0.87
Save-Waveform: true
Content-Length: 24

builtin:speech/directory
```

S->C:

```
MRCP/2.0 83 1 200 IN-PROGRESS
Channel-Identifier: 3ea18b9854af11e7@speechrecog
```

S->C:

```
MRCP/2.0 115 START-OF-INPUT 1 IN-PROGRESS
Channel-Identifier: 3ea18b9854af11e7@speechrecog
Input-Type: speech
```

S->C:

```
MRCP/2.0 497 RECOGNITION-COMPLETE 1 COMPLETE
Channel-Identifier: 3ea18b9854af11e7@speechrecog
Completion-Cause: 000 success
Waveform-Uri: <http://localhost/utterances/utter-3ea18b9854af11e7-
1.wav>;size=20480;duration=1280
Content-Type: application/x-nlsml
Content-Length: 213

<?xml version="1.0"?>
<result>
  <interpretation grammar="builtin:speech/directory" confidence="0.88">
    <instance>call Steve</instance>
    <input mode="speech">call Steve</input>
  </interpretation>
</result>
```

6.3 Speech Recognition with Dynamic Speech Context

This examples demonstrates how to perform speech recognition, by using a DEFINE-GRAMMAR request to specify a speech context and further reference the defined speech context in a RECOGNIZE request.

C->S:

```
MRCP/2.0 314 DEFINE-GRAMMAR 1
Channel-Identifier: 25902c3a54b011e7@speechrecog
Content-Type: application/xml
Content-Id: request1@form-level
Content-Length: 146

<speech-context>
  <phrase>call Steve</phrase>
  <phrase>call John</phrase>
  <phrase>dial 5</phrase>
  <phrase>dial 6</phrase>
</speech-context>
```

S->C:

```
MRCP/2.0 112 1 200 COMPLETE
Channel-Identifier: 25902c3a54b011e7@speechrecog
Completion-Cause: 000 success
```

C->S:

```
MRCP/2.0 305 RECOGNIZE 2
Channel-Identifier: 25902c3a54b011e7@speechrecog
Content-Type: text/uri-list
Cancel-If-Queue: false
No-Input-Timeout: 5000
Recognition-Timeout: 10000
Start-Input-Timers: true
Confidence-Threshold: 0.87
Save-Waveform: true
Content-Length: 27

session:request1@form-level
```

S->C:

```
MRCP/2.0 83 2 200 IN-PROGRESS
Channel-Identifier: 25902c3a54b011e7@speechrecog
```

S->C:

```
MRCP/2.0 115 START-OF-INPUT 2 IN-PROGRESS
Channel-Identifier: 25902c3a54b011e7@speechrecog
Input-Type: speech
```

S->C:

```
MRCP/2.0 500 RECOGNITION-COMPLETE 2 COMPLETE
Channel-Identifier: 25902c3a54b011e7@speechrecog
Completion-Cause: 000 success
Waveform-Uri: <http://localhost/utterances/utter-25902c3a54b011e7-
2.wav>;size=20480;duration=1280
Content-Type: application/x-nlsml
Content-Length: 216

<?xml version="1.0"?>
<result>
  <interpretation grammar="session:request1@form-level" confidence="0.88">
    <instance>call Steve</instance>
    <input mode="speech">call Steve</input>
```

```
  </interpretation>
</result>
```

6.4 DTMF Recognition with Built-in Grammar

This examples demonstrates how to reference a built-in DTMF grammar in a RECOGNIZE request.

C->S:

```
MRCP/2.0 266 RECOGNIZE 1
Channel-Identifier: d26bef74091a174c@speechrecog
Content-Type: text/uri-list
Cancel-If-Queue: false
Start-Input-Timers: true
Confidence-Threshold: 0.7
Speech-Language: en-US
Dtmf-Term-Char: #
Content-Length: 19

builtin:dtmf/digits
```

S->C:

```
MRCP/2.0 83 1 200 IN-PROGRESS
Channel-Identifier: d26bef74091a174c@speechrecog
```

S->C:

```
MRCP/2.0 113 START-OF-INPUT 1 IN-PROGRESS
Channel-Identifier: d26bef74091a174c@speechrecog
Input-Type: dtmf
```

S->C:

```
MRCP/2.0 382 RECOGNITION-COMPLETE 1 COMPLETE
Channel-Identifier: d26bef74091a174c@speechrecog
Completion-Cause: 000 success
Content-Type: application/x-nlsml
Content-Length: 197

<?xml version="1.0"?>
```

```
<result>
  <interpretation grammar="builtin:dtmf/digits" confidence="1.00">
    <input mode="dtmf">1 2 3 4</input>
    <instance>1234</instance>
  </interpretation>
</result>
```

6.5 Speech and DTMF Recognition

This examples demonstrates how to perform recognition by activating both speech and DTMF grammars. In this example, the user is expected to input a 4-digit pin.

C->S:

```
MRCP/2.0 275 RECOGNIZE 1
Channel-Identifier: 6ae0f23e1b1e3d42@speechrecog
Content-Type: text/uri-list
Cancel-If-Queue: false
Start-Input-Timers: true
Confidence-Threshold: 0.7
Speech-Language: en-US
Content-Length: 47

builtin:dtmf/digits?length=4
builtin:speech/pin
```

S->C:

```
MRCP/2.0 83 2 200 IN-PROGRESS
Channel-Identifier: 6ae0f23e1b1e3d42@speechrecog
```

S->C:

```
MRCP/2.0 115 START-OF-INPUT 2 IN-PROGRESS
Channel-Identifier: 6ae0f23e1b1e3d42@speechrecog
Input-Type: speech
```

S->C:

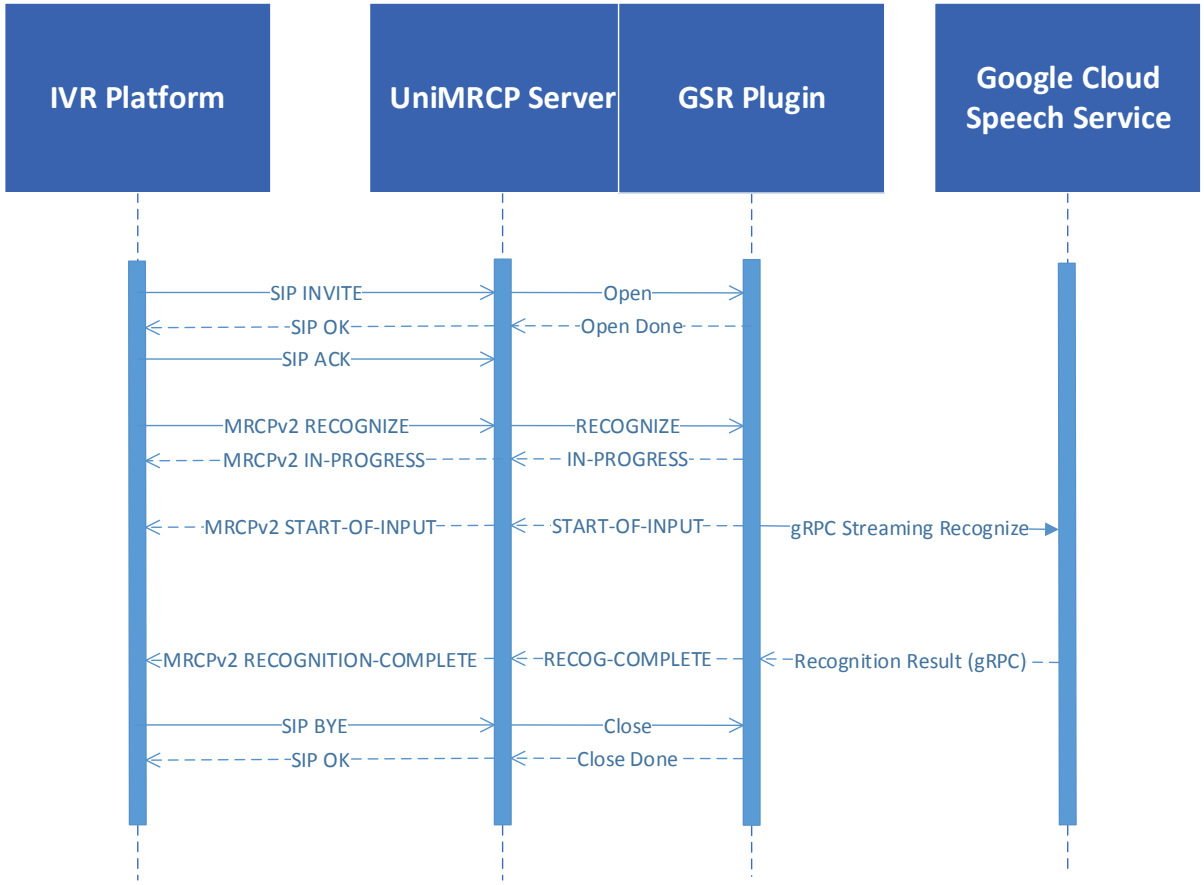
```
MRCP/2.0 399 RECOGNITION-COMPLETE 2 COMPLETE
Channel-Identifier: 6ae0f23e1b1e3d42@speechrecog
```

```
Completion-Cause: 000 success
Content-Type: application/x-nlsml
Content-Length: 214

<?xml version="1.0"?>
<result>
  <interpretation grammar=" builtin:speech/pin" confidence="1.00">
    <instance>one two three four</instance>
    <input mode="speech">one two three four</input>
  </interpretation>
</result>
```

7 Sequence Diagram

The following sequence diagram outlines common interactions between all the main components involved in a typical recognition session performed over MRCPv2.



8 References

8.1 Google Cloud Platform

- Speech API
- How-to Guides
- Best Practices

8.2 Specifications

- Speech Recognizer Resource
- NLSML Results