Powered by Universal Speech Solutions LLC

UniMRCP

Watson SR Plugin

Usage Guide

Revision: 1

Created: July 4, 2018

Last updated: July 4, 2018

Author: Arsen Chaloyan

Table of Contents

1 Overview

This guide describes how to configure and use the IBM Watson Speech Recognition (SR) plugin to the UniMRCP server. The document is intended for users having a certain knowledge of IBM Watson Speech-to-Text API 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 1.5.0 and above

UniMRCP Watson SR 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 Watson SR 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
- ✓ Speech-Complete-Timeout
- ✓ Speech-Incomplete-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 speech transcription grammar
- ✓ Built-in/embedded DTMF grammar

2.5 Results

- ✓ NLSML

3 Configuration Format

The configuration file of the Watson SR plugin is located in */opt/unimrcp/conf/umswatsonsr.xml*. The configuration file is written in XML.

3.1 Document

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

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.
credentials-file	File path	Specifies the IBM Watson 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
<ws-streaming-recognition>	String	Specifies parameters of streaming recognition employed via the WebSocket protocol.
<speech-dtmf-input-detector>	String	Specifies parameters of the speech and DTMF input detector.
<utterance-manager>	String	Specifies parameters of the utterance manager.
<rdr-manager>	String	Specifies parameters of the Recognition Details Record (RDR) manager.
<monitoring-agent>	String	Specifies parameters of the monitoring manager.
<license-server>	String	Specifies parameters used to connect to the license server. The use of the license server is

		optional.

Example

This is an example of a bare document.

```
<umswatsonsr license-file="umswatsonsr_*.lic" credentials-file="watsonsr.credentials">
</umswatsonsr>
```

3.2 Streaming Recognition

This element specifies parameters of streaming recognition employed via the WebSocket interface.

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://console.bluemix.net/docs/services/speech-to-text/input.html#models
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*.
smart-formatting	Boolean	Specifies whether the service converts dates, times, numbers, currency, and similar values into more conventional representations in the final transcript.
start-of-input	String	Specifies the source of start of input event sent to the client (use "service-originated" to rely on service-originated first interim result and "internal" for plugin-originated event).
auth-validation-period	Integer	Specifies a period in seconds used to re-validate access token based on subscription key.

Parent

<umswatsonsr>

Children

None.

Example

This is an example of streaming recognition element.

```
<ws-streaming-recognition
   language="en-US"
   max-alternatives="1"
   smart-formatting="true"
   start-of-input="internal"
   auth-validation-period="480"
/>
```

3.3 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.
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. The complete timeout is used when there is an interim result available.
speech-incomplete-timeout	Time interval [msec]	Specifies how long to wait in transition mode before triggering an end of speech input event. The incomplete timeout is used as long as there is no interim result available. Afterwards, the complete timeout is used.
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

<umswatsonsr>

Children

None.

Example

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

```
<speech-dtmf-input-detector
  vad-mode="2"
  speech-start-timeout="300"
  speech-complete-timeout="1000"
  speech-incomplete-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"
/>
```

3.4 Utterance Manager

This element specifies parameters of the utterance manager.

Attributes

Name	Unit	Description
save-waveforms	Boolean	Specifies whether to save waveforms or not.
purge-existing	Boolean	Specifies whether to delete existing records on start-up.
max-file-age	Time interval [min]	Specifies a time interval in minutes after expiration of which a waveform is deleted. Set 0 for infinite.
max-file-count	Integer	Specifies the max number of waveforms to store. If reached, the oldest waveform is deleted. Set 0 for infinite.
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.

Parent

<umswatsonsr>

Children

None.

Example

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

```
<utterance-manager
  save-waveforms="false"
  purge-existing="false"
  max-file-age="60"
  max-file-count="100"
  waveform-base-uri="http://localhost/utterances/"
  waveform-folder=""
/>
```

3.5 RDR Manager

This element specifies parameters of the Recognition Details Record (RDR) manager.

Attributes

Name	Unit	Description
save-records	Boolean	Specifies whether to save recognition details records or not.
purge-existing	Boolean	Specifies whether to delete existing records on start-up.
max-file-age	Time interval [min]	Specifies a time interval in minutes after expiration of which a record is deleted. Set 0 for infinite.
max-file-count	Integer	Specifies the max number of records to store. If reached, the oldest record is deleted. Set 0 for infinite.
record-folder	Dir path	Specifies a folder to store recognition details records in. Defaults to ${UniMRCPInstallDir}/var.

Parent

<umswatsonsr>

Children

None.

Example

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

```
<rdr-manager
  save-records="false"
  purge-existing="false"
  max-file-age="60"
  max-file-count="100"
  waveform-folder=""
/>
```

3.6 Monitoring Agent

This element specifies parameters of the monitoring agent.

Attributes

Name	Unit	Description
refresh-period	Time interval [sec]	Specifies a time interval in seconds used to periodically refresh usage details. See <usage-refresh-handler>.

Parent

<umswatsonsr>

Children

<usage-change-handler>
<usage-refresh-handler>

Example

The example below defines a monitoring agent with usage change and refresh handlers.

```
<monitoring-agent refresh-period="60">

  <usage-change-handler>
    <log-usage enable="true" priority="NOTICE"/>
  </usage-change-handler>

  <usage-refresh-handler>
    <dump-channels enable="true" status-file="umswatsonsr-channels.status"/>
  </usage-refresh-handler >

</monitoring-agent>
```

3.7 Usage Change Handler

This element specifies an event handler called on every usage change.

Attributes

None.

Parent

<monitoring-agent>

Children

<log-usage>
<update-usage>
<dump-channels>

Example

This is an example of the usage change event handler.

```
<usage-change-handler>
  <log-usage enable="true" priority="NOTICE"/>
  <update-usage enable="false" status-file="umswatsonsr-usage.status"/>
  <dump-channels enable="false" status-file="umswatsonsr-channels.status"/>
</usage-change-handler>
```

3.8 Usage Refresh Handler

This element specifies an event handler called periodically to update usage details.

Attributes

None.

Parent

<monitoring-agent>

Children

<log-usage>
<update-usage>
<dump-channels>

Example

This is an example of the usage change event handler.

```
<usage-refresh-handler>
  <log-usage enable="true" priority="NOTICE"/>
  <update-usage enable="false" status-file="umswatsonsr-usage.status"/>
  <dump-channels enable="false" status-file="umswatsonsr-channels.status"/>
</usage-refresh-handler>
```

3.9 License Server

This element specifies parameters used to connect to the license server.

Attributes

Name	Unit	Description

enable	Boolean	Specifies whether the use of license server is enabled or not. If enabled, the license-file attribute is not honored.
server-address	String	Specifies the IP address or host name of the license server.
certificate-file	File path	Specifies the client certificate used to connect to the license server. File name may include patterns containing a '*' sign. If multiple files match the pattern, the most recent one gets used.
ca-file	File path	Specifies the certificate authority used to validate the license server.
channel-count	Integer	Specifies the number of channels to check out from the license server. If not specified or set to 0, either all available channels or a pool of channels will be checked based on the configuration of the license server.

Parent

<umswatsonsr>

Children

None.

Example

The example below defines a typical configuration which can be used to connect to a license server located, for example, at 10.0.0.1.

```
<license-server
  enable="true"
  server-address="10.0.0.1"
  certificate-file="unilic_client_*.crt"
  ca-file="unilic_ca.crt"
/>
```

For further reference to the license server, visit

http://unimrcp.org/licserver

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 *umswatsonsr.xml* is used. The parameter defaults to *en-US*.
For supported languages and their corresponding codes, visit the following link.

https://console.bluemix.net/docs/services/speech-to-text/input.html#models

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. Either 8 or 16 kHz can be used by IBM Watson Speech-to-Text API.

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 suit 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. Note that if the *start-of-input* parameter in the *ws-streaming-recognition* is set to *service-originated*, then a START-OF-INPUT event is sent to the client at a later stage, upon reception of a first interim result.

- 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.

- 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 suit 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 Maintaining Utterances

Saving of utterances is not required for regular operation and is disabled by default. However, enabling this functionality allows to save utterances sent to the Watson Speech service and later listen to them

offline.

The relevant settings can be specified via the element *utterance-manager*.

- 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.

- purge-existing

  This parameter specifies whether to delete existing waveforms on start-up.

- max-file-age

  This parameter specifies a time interval in minutes after expiration of which a waveform is deleted. If set to 0, there is no expiration time specified.

- max-file-count

  This parameter specifies the maximum number of waveforms to store. If the specified number is reached, the oldest waveform is deleted. If set to 0, there is no limit specified.

- 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. The directory defaults to *${UniMRCPInstallDir}/var*.

4.8 Maintaining Recognition Details Records

Producing of recognition details records (RDR) is not required for regular operation and is disabled by default. However, enabling this functionality allows to store details of each recognition attempt in a separate file and analyze them later offline. The RDRs ate stored in the JSON format.

The relevant settings can be specified via the element *rdr-manager*.

- save-records

  This parameter specifies whether to save recognition details records or not.

- purge-existing

  This parameter specifies whether to delete existing records on start-up.

- max-file-age

  This parameter specifies a time interval in minutes after expiration of which a record is deleted. If set to 0, there is no expiration time specified.

- max-file-count

This parameter specifies the maximum number of records to store. If the specified number is reached, the oldest record is deleted. If set to 0, there is no limit specified.

- record-folder

This parameter specifies a path to the directory used to store records in. The directory defaults to *${UniMRCPInstallDir}/var*.

4.9 Monitoring Usage Details

The number of in-use and total licensed channels can be monitored in several alternate ways. There is a set of actions which can take place on certain events. The behavior is configurable via the element *monitoring-agent*, which contains two event handlers: *usage-change-handler* and *usage-refresh-handler*.

While the *usage-change-handler* is invoked on every acquisition and release of a licensed channel, the *usage-refresh-handler* is invoked periodically on expiration of a timeout specified by the attribute *refresh-period*.

The following actions can be specified for either of the two handlers.

- log-usage

This action logs the number of in-use and total licensed channels. The following is a sample log statement, indicating 0 in-use and 2 total channels.

```
[NOTICE] WSR Usage: 0/2
```

- update-usage

This action writes the number of in-use and total licensed channels to a status file *umswatsonsr-usage.status*, located by default in the directory *${UniMRCPInstallDir}/var/status*. The following is a sample content of the status file.

```
in-use channels: 0
total channels: 2
```

- dump-channels

This action writes the identifiers of in-use channels to a status file *umswatsonsr-channels.status*, located by default in the directory *${UniMRCPInstallDir}/var/status*.

5 Recognition Grammars and Results

5.1 Using Built-in Speech Transcription

For generic speech transcription, having no speech contexts defined, a pre-set identifier *transcribe* must be used by the MRCP client in a RECOGNIZE request as follows:

```
builtin:speech/transcribe
```

5.2 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.

5.3 Retrieving Results

Results received from the speech service are transformed to the NLSML format.

6 Usage Examples

6.1 Speech Transcription

This examples demonstrates how to perform speech recognition by using a RECOGNIZE request.

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="0.95">
    <instance>what's the weather like</instance>
    <input mode="speech">what's the weather like</input>
  </interpretation>
</result>
```

6.2 DTMF Recognition

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>
```

6.3 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
```

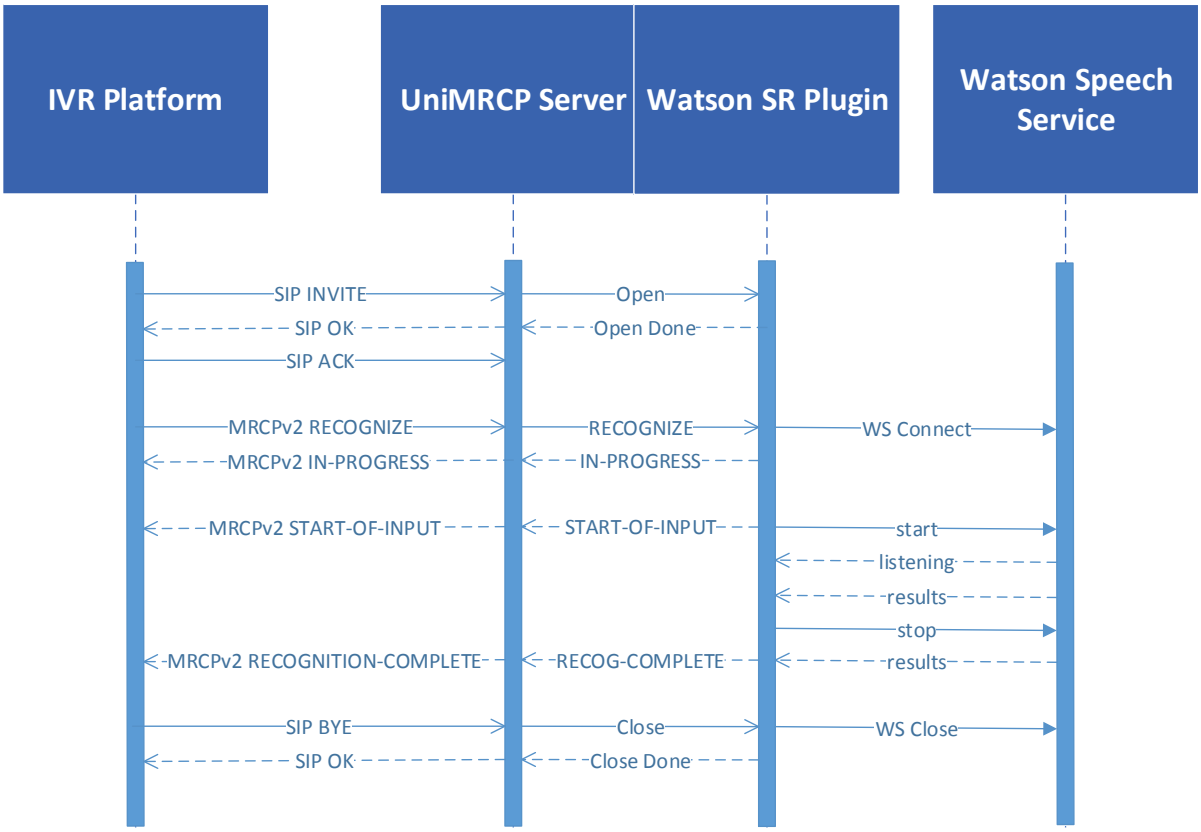
S->C:

```
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 IBM Watson

- Text to Speech API
- Credentials

8.2 Specifications

- Speech Recognizer Resource
- NLSML Results