

VOICEGENIE

VoiceGenie 7 Speech Resource Management (SRM) Users' Guide

April 13th, 2005



VoiceGenie Technologies Inc.
1120 Finch Ave. W. • Toronto, Ontario • M3J 3H7 • Canada
T. +1.416.736.4151 • F. +1.416.736.1551 • support@voicegenie.com
www.voicegenie.com

VoiceGenie Contacts

VoiceGenie Technologies Inc.
1120 Finch Avenue West
Toronto, Ontario
Canada
M3J 3H7

T. +1.416.736.4151
F. +1.416.736.1551
support@voicegenie.com

<http://www.voicegenie.com/index.html>

Proprietary / Copyright Information

This material contains proprietary and/or copyright information of VoiceGenie Technologies Inc. and may not be copied, used, or disclosed without the permission of VoiceGenie Technologies Inc.

© COPYRIGHT 2005 VoiceGenie Technologies Inc.

Notice

Every effort was made to ensure that the information in this document was complete and accurate at the time of printing. However, this information is subject to change without notice.

Please note that VoiceGenie makes no warranties with respect to future releases. The information provided herein is for informational purposes only. VoiceGenie reserves the right to change product release schedules and/or features allocated to a product release at any time.

Trademarks

All trademarks are the property of their respective owners. Where those designations appear in this document, and VoiceGenie is aware of a trademark claim, the designations have been printed in initial caps or all caps.



Table of Contents

Proprietary / Copyright Information _____ **3**

Notice_____ **3**

Trademarks _____ **3**

Table of Contents_____ **4**

Revision History_____ **5**

1 Introduction _____ **6**

 1.1 Functional and Architecture Overview _____ 6

 1.2 Other VoiceGenie Components _____ 10

 1.2.1 *Media Platform (MP)*_____ 10

 1.2.2 *OA&M Framework*_____ 10

 1.3 Application Server _____ 11

 1.3.1 *Hotkey Grammars* _____ 11

 1.3.2 *Builtin Grammars*_____ 11

 1.3.3 *Inline and Implied Grammars* _____ 11

2 The SRM Client _____ **12**

 2.1 Provisioning in VoiceGenie 7 Media Platform _____ 12

 2.1.1 *Provisioning Data* _____ 12

 2.1.2 *Provisioning and Fail-over* _____ 13

 2.2 HTTP Access to Grammars _____ 14

 2.3 Directory structure _____ 14

 2.3.1 *The software* _____ 14

 2.3.2 *Configuration* _____ 14

 2.3.3 *Hotkey and builtin Grammars* _____ 14

3 The SRM Server _____ **15**

 3.1 Architectural Overview _____ 15

 3.2 Directory Structure _____ 16

 3.3 Application Server _____ 16

 3.4 Provisioning _____ 17

 3.5 Metrics and Logging _____ 17

 3.6 CLC and SMC Operations _____ 17

 3.6.1 *CLC command to start/stop srmserver:* _____ 17

 3.6.2 *CLC commands to change srmserver operation states:* _____ 18

 3.6.3 *Other CLC commands:* _____ 18

4 SRM RTSP-TTS Engine _____ **19**

5 SRM TTY/TDD Engine _____ **19**



Revision History

Version	Date	Change Summary	Author/Editor
1	August 31 st , 2004	Initial release	Alex Lee
2	Dec 1 st , 2004	Added VRMProxy Section	Johnson Tse
3	February 16 th , 2005	Added Moncton Provisioning Section	Lin Chen
4	March 11 th , 2005	Additional Updates for VoiceGenie 7	Andrew Ho
4.1	March 29 th , 2005	Revised version	Andrew Ho
4.2	April 13 th , 2005	Final Revision for VoiceGenie 7	Andrew Ho

1 Introduction

This document contains information on how the Speech Resource Management (SRM) components are integrated into other VoiceGenie components.

The following table gives definitions of some acronyms that are used throughout this document:

Acronyms	Full Definitions
ASR	Automated Speech Recognition (Engines/Technologies)
CLC	Command Line Console -- A command line interface that can be used to query information and issue commands
MRCP	Media Resource Control Protocol -- Adopted by the VoiceGenie Media Platform to control ASR and TTS resources
SRM	Speech Resource Management -- A component integrated into the VoiceGenie Media Platform to provide Speech Recognition and Synthesis functionalities to the application developers
SMC	System Management Console -- A web based tool for administering clusters of VoiceGenie VoiceXML Platforms
OA&M	Operation, Administration and Management
TTS	Text To Speech (Engines/Technologies)

The following sections may contain references to terminology that has become obsolete:

Historical Terms	New Terms
PhoneWeb Software / NeXusPoint 6.4.x Software	VoiceGenie 7 Software
Cluster Management Platform (CMP)	OA&M Framework
Voice Resource Manager (VRM)	Speech Resource Management (SRM)
VoiceGenie Management Console (VMC)	System Management Console (SMC)

1.1 Functional and Architecture Overview

With modern day telephony applications, it is no longer sufficient to provide services to customers using touch-tones and pre-recorded audio. The Service Providers require dynamic prompts to be generated and play back to the Callers; while the Callers demand a better and more natural way of interacting with the application.

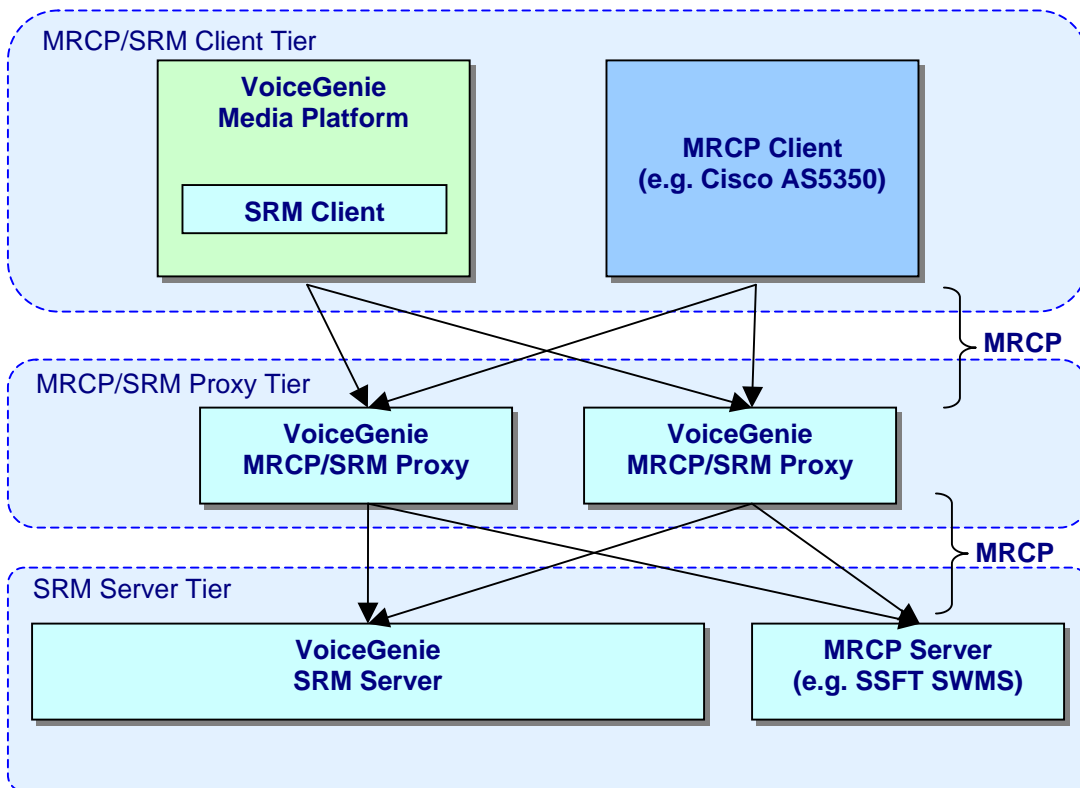
The Speech Resource Management (SRM) system is used to integrate with 3rd party Automatic Speech Recognition (ASR) and Text-to-Speech (TTS) engines. This component is integrated into the VoiceGenie Media Platform to provide Speech Recognition and Synthesis functionalities to the application developers. Using this component, products from many 3rd party vendors has been integrated to work with the VoiceGenie Media Platform.

The Speech Resource Management system consists of three components: the SRM client component, the SRM server component and the SRM proxy component. The three components are integrated with VoiceGenie Media Platform in two different architecture models, the three-tiered (client/proxy/server) model and the two-tiered(client/server)model. The [VoiceGenie 7 MRCP Proxy Users' Guide](#) and the [VoiceGenie 7 MRCP Proxy System Reference Guide](#) contain information for MRCP Proxy and the three-tiered model. This document will focus on the SRM client and server components.

The client component is a Dynamic Link Library (either a .so library on Linux or a .dll library in Windows) used by the media platform to access ASR/TTS functionality. The server component accepts and handles requests from the client component. The server component contains integration software written specifically to work with each of the 3rd party ASR/TTS products.

With the emergence of the Media Resource Control Protocol (MRCP) as a wide-supported standard, VoiceGenie has adopted MRCP as the protocol of choice for ASR and TTS integration. Many ASR/TTS vendors have also adopted MRCP as the method of control for their products.

Fig. 1 SRM Architecture Overview



The SRM client can communicate directly with these speech resources in MRCP protocol. We name such integration as MRCP Direct. Fig. 2 illustrates MRCP direct integration architecture. The SRM server is essential when the 3rd party Speech Resource products are not MRCP compliance. The communication between the SRM client and the SRM server are based on MRCP protocol while the communication

between SRM server and Speech Resources are based on vendor's proprietary protocol. We name such integration as MRCP Native. Fig.3 illustrates the architecture of the MRCP Native integration.

Fig. 2 MRCP Direct Integration Architecture

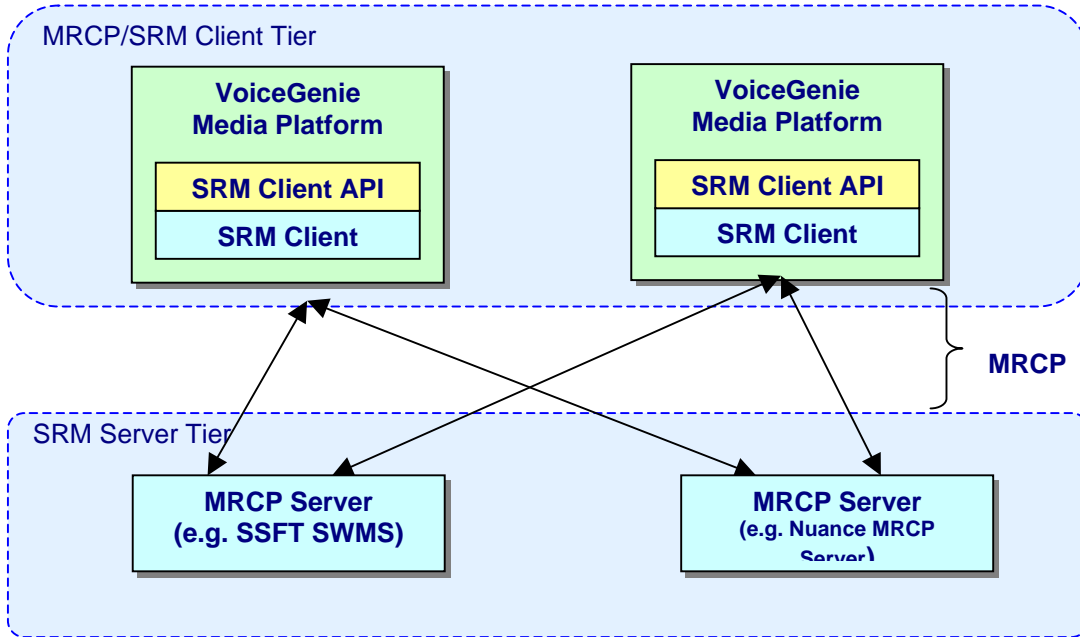
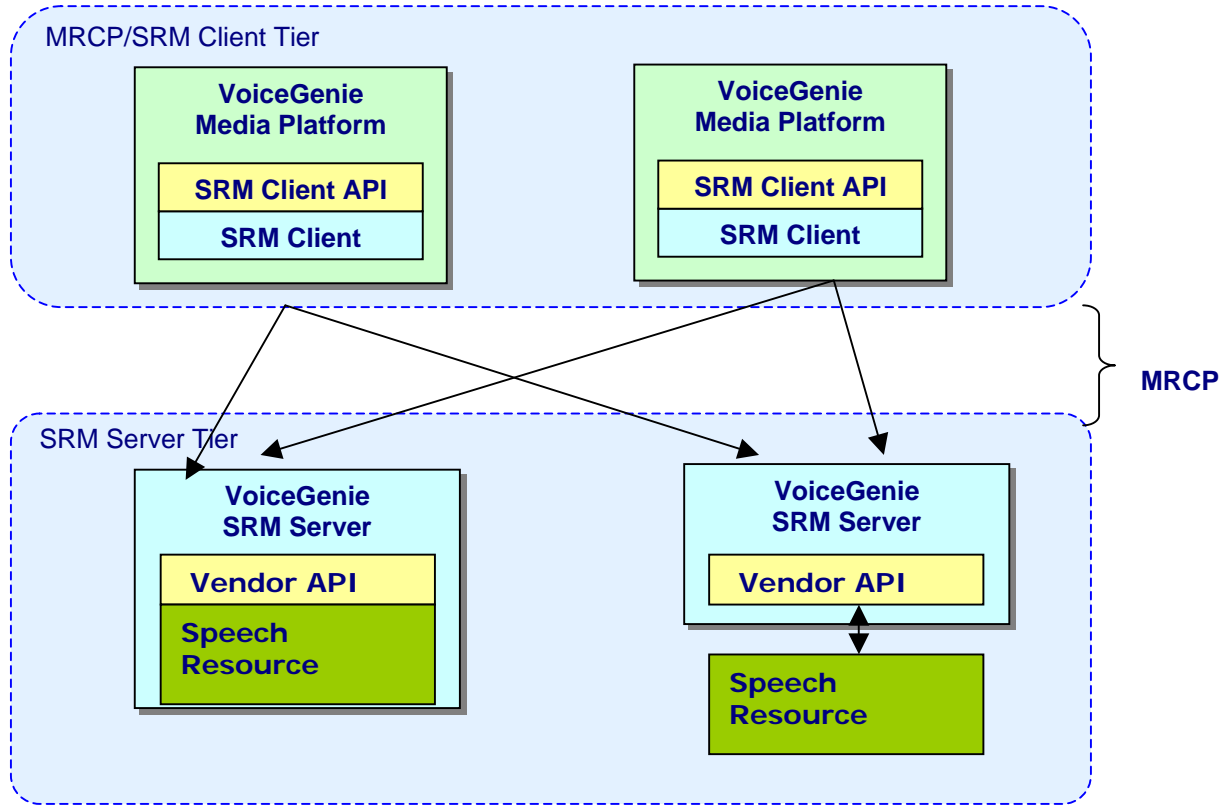


Fig. 3 MRCP Native Integration Architecture

In this situation, the VoiceGenie SRM server contains software from both VoiceGenie and other 3rd party vendors that provides ASR/TTS software.



1.2 Other VoiceGenie Components

Before explaining the SRM architecture and providing detailed configuration information, the following subsections describe the functionality of the VoiceGenie components that have interference with Speech Resource components.

1.2.1 Media Platform (MP)

Media Platform contains a Call Manager component used to manage various resources of a call. It handles incoming/outgoing calls, accepts instructions from the VoiceXML Interpreter component, and makes requests to use ASR and TTS. The Call Manager is the component that loads and uses the SRM Client.

1.2.2 OA&M Framework

The OA&M Framework is a management platform that takes care of operations, administration, and maintenance. It manages the software process of the entire components list above, and provides the following three main functionalities.

- 1) Monitoring and management agent. An OA&M Framework agent is embedded into all VoiceGenie software components to be managed and monitored by the OA&M Framework. The OA&M Framework agent will define an object-based structure that allows monitoring/management information to be exposed via SNMP. However, the OA&M Framework agent will also provide a non-SNMP based connection used by the monitoring and management engine (below) to obtain the real-time status of the system. The SRM server contains an OA&M Framework agent and is an independent process started by the OA&M Framework. In contrast, the SRM client is part of the Call Manager, and makes use of the OA&M Framework agent shared by the Call Manager.
- 2) Monitoring and management engine. This part of the OA&M Framework will extend the monitoring currently performed by platform pooling to be real-time in nature, using a persistent connection into VoiceGenie software based on the OA&M Framework agent. This engine will also be responsible for carrying out requests issued against the platform, such as starting or stopping the software, or performing an upgrade or rollback of nodes within a cluster.
- 3) System Management Console (SMC). This web-based interface will retrieve information from the monitoring/management engine through a database, and will provide a visual interface allowing the current status and configuration of VoiceGenie components to be reviewed. The web-based interface will also allow operations to be scheduled against components or groups of components. The monitoring/management engine will actually perform the operations. A command-line interface also provides most of the functionality offered by the SMC, called the command-line console (CLC). This component runs on each platform that is part of the OA&M Framework cluster.

1.3 Application Server

On the SRM client side, the application server is required to serve grammars generated by Media Platform to off-board ASR server. An Apache server is used on Linux for this purpose, while an IIS server is used on Windows Platforms.

The application server must be configured to server three types of grammars generated by MP.

1.3.1 Hotkey Grammars

Hotkey Grammars are used to match the UNIVERSALS properties, for the "Help", "Cancel", and "Exit" hotwords. These are installed in the directories /usr/local/phoneweb/grammar/<engine name>/hotkey/ and C:\VoiceGenie\MP\gramamr\<engine name>\hotkey\ respectively for Linux and Windows, for each of the engines to be used by the Media Platform.

1.3.2 Builtin Grammars

Some engines do not support VoiceXML builtin grammars internally on the engine side. For these engines, the builtin grammars also reside on the Media Platform, in the directories /usr/local/phoneweb/grammar/<engine name>/ and C:\VoiceGenie\MP\gramamr\<engine name>\ for Linux and Windows respectively.

1.3.3 Inline and Implied Grammars

Inline grammars and Implied grammars (menu/option grammar) are generated by Media Platform dynamically. These grammar are placed in /usr/local/phoneweb/tmp and in C:\VoiceGenie\MP\tmp for Linux and Windows respectively.

In the MRCP Native integration, the srmserver could locate in a separated box from the Media Platform. Application server is therefore required to server saved utterance data by SRM server. An Apache server is used on Linux for this purpose, while an IIS server is used on Windows Platforms.

2 The SRM Client

The SRM Client is a component that interacts with speech resource servers in MRCP protocol to access the speech resources required by the Media Platform. It conveys data from the Media Platform to speech resource servers. It also provides fail-over features by monitoring speech resources. Its configuration file offers user possibility to control the SRM client behavior. SMC provides a graphical User Interface for provisioning the speech resources required by the Media Platform.

2.1 Provisioning in VoiceGenie 7 Media Platform

When a Speech Resource is deployed, the OA&M Framework generates its default provisioning data. This data is saved in callmgr provision data file, /usr/local/phoneweb/config/cm_provision.dat and C:\VoiceGenie\mp\config\cm_provision.dat for Linux and Windows respectively and can only be accessed through the Configuration page in SMC in the Media Platform section.

A typically speech resource will have the following formatted provision data in cm_provision.dat file.

```
<entry id="302" type="6" name="Speech Resources" >
<param name="vrm.client.resource.name" value="SPEECHWORKS"/>
<param name="vrm.client.resource.address" value="10.0.0.78"/>
<param name="vrm.client.resource.port" value="22000"/>
<param name="vrm.client.resource.uri" value="rtsp://10.0.0.78/recognizer"/>
<param name="vrm.client.resource.type" value="ASR"/>
<param name="vrm.client.SendVGParams" value="true"/>
<param name="vrm.client.HotKeyBasePath" value="/vggrammarbase/speechworks/hotkey"/>
<param name="vrm.client.HotKeyLocalPath" value="$VGROOT/grammar/speechworks/hotkey"/>
</entry>
```

The provisioning data of a speech resource describes the specific configurations that the resource requires. User can change the default value for the provision data to meet special requirement. User can also provision speech resource to take advantages of the fail-over feature that SRM client offers.

2.1.1 Provisioning Data

The following screen capture illustrates a typical Speech Resource Provisioning.

Each speech resource has its identity data, they are the Speech Resource Name, Hostname/IP and Port. The identity data cannot be modified for each deployed resource. The Speech Resource Name is the name of the TTS/ASR engine which can be specified by property TTSENGINE/ASRENGINE by an application. The Speech Resource Name only allows upper case alphanumeric, dash, dot, and underscore.

Resource URI defines how this resource should be accessed from a MRCP client. A resource URI is normally defined in the resource configuration file on the server side. The value of this data must match with the value defined by the server.

Resource Type defines if the server is a TTS or an ASR.

Speech Resource Name:	SPEECHWORKS		ID: 79 (80)
Hostname/IP:	10.0.0.106	Port:	22000
Resource URI:	rtsp://10.0.0.106/spwx_recognizer		
Resource Type:	ASR		
Parameter Name:	vrn.client.SendVGPara	Value:	true <input type="button" value="Remove"/>
Parameter Name:	vrn.client.HotKeyBaseF	Value:	/vggrammarbase/spee <input type="button" value="Remove"/>
Parameter Name:	vrn.client.HotKeyLocalF	Value:	\$VGR00T/grammar/sp <input type="button" value="Remove"/>
Parameter Name:	<input type="text"/>	Value:	<input type="text"/> <input type="button" value="Add"/>
<input type="button" value="Update"/> <input type="button" value="Delete"/> <input type="button" value="Target as Primary"/> <input type="button" value="Target as Backup"/>			

Users can freely add the Per-EngineType parameters listed under the Speech Resources Configuration Section in each table, by typing the name (note always with prefix vrn.client.) in the "Parameter Name" box and its value in the "Value" box and clicking the Add button to confirm.

Users can remove an existing parameter by clicking the Remove button beside the parameter.

Clicking the Update button will update the database as well as the cm_provision.dat for all the changes. The user must re-target the resource to a Media Platform to ensure the change takes effect on that Media Platform.

Clicking Delete button will un-target from all the already targeted Media Platform and will delete the Speech Resource data from database and cm_provision.dat.

Clicking the "Target as Primary" button users can target/un-target the Speech Resource as a primary engine to a Media Platform.

Clicking the "Target as Backup" button users can target/un-target the Speech Resource as a backup engine to a Media Platform.

For a full list of all provisioning parameters, please refer to the [SRM System Reference Guide](#), "SRM Client Speech Resource Configuration Parameters" Section.

2.1.2 Provisioning and Fail-over

The SRM client provides fail-over ability by using the speech resources provisioning data.

Resource Failure Definition

From SRM client point of view, a speech resource is marked as failed if

- a. socket disconnection is detected
- b. resource does not reply to the ping message within vrn.client.ping.timeout value.

- c. the resource counter is 0 in the ping reply message from SRM server.

Primary speech resources with the same Speech Resource Name will be grouped as a resource pool. Requests are assigned to each resource in a round-robin fashion. When the SRM client detects a resource failure, it will skip the resource. New Requests will be assigned to the remaining resources.

When all primary resources are detected as failed, the new requests will be assigned to the back up resources. The backup resources have the same fail-over capability as the primary ones.

2.2 HTTP Access to Grammars

Off-board ASR servers need to fetch grammars (such as the hotkey and built-in grammars described above, as well as inline and implicit grammars dynamically generated by the applications) from the Media Platform. As a result, a web server is installed and configured on the Media Platforms. An Apache server is used on Linux for this purpose, while an IIS server is used on Windows Platforms.

2.3 Directory structure

This section describes the directory layout of SRM client component on the VoiceGenie platform.

2.3.1 The software

Starting from the VoiceGenie 7 release, this shared library is located in /usr/local/phoneweb/lib/libvrmclient.so (Linux) and C:\VoiceGenie\mp\lib\libvrmclient.dll (Windows).

2.3.2 Configuration

The SRM client obtains its configuration information from its respective module (i.e. the Call Manager, VoiceXML Interpreter). The configuration parameters for the SRM client are embedded in each of their configuration files, which can be accessed through the SMC interface.

2.3.3 Hotkey and builtin Grammars

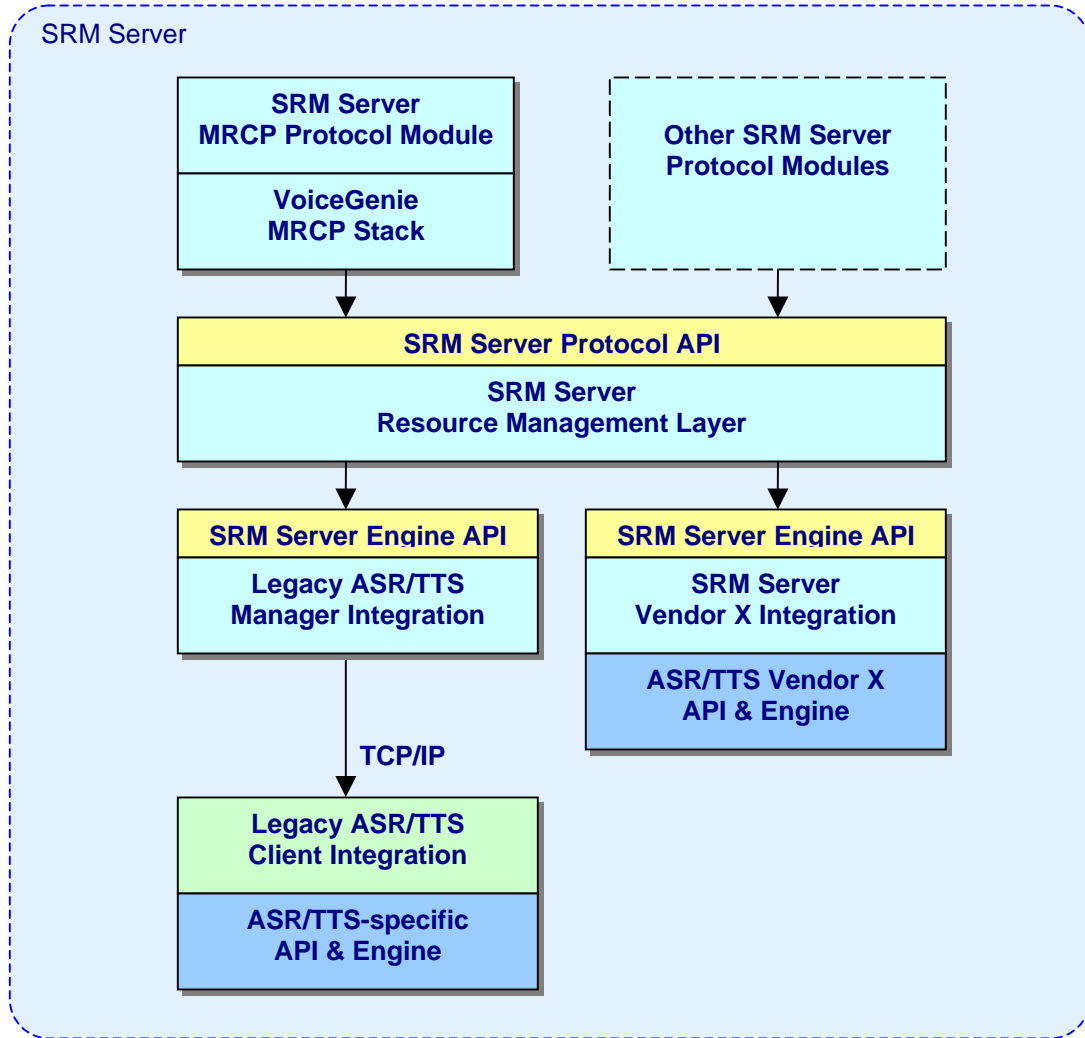
Hotkey Grammars are used to match the UNIVERSALS properties, for the "Help", "Cancel", and "Exit" hotwords. These are installed in the directories /usr/local/phoneweb/grammar/<engine name>/hotkey/ and C:\VoiceGenie\MP\gramamr\<engine name>\hotkey\ respectively for Linux and Windows, for each of the engines to be used by the Media Platform.

For some engines, they do not support built-in grammars internally on the engine side. For these engines, the built-in grammars also reside on the Media Platform, in the directories /usr/local/phoneweb/grammar/<engine name>/ and C:\VoiceGenie\MP\grammar\<engine name>\.

3 The SRM Server

3.1 Architectural Overview

The SRM server is an independent process used to handle Speech Recognition and Synthesis requests in MRCP Native configuration as illustrated in Fig.3 of section 1.1. The SRM server creates and executes other processes to perform the recognition and synthesis, which can be illustrated by the following diagram:



For each ASR/TTS product VoiceGenie has integrated with, there is a client process that works with the SRM server that acts as a bridge between the SRM server and the ASR/TTS product itself. The advantage of this approach is that, whenever changes are needed in the integration of one of the partners, or a

support for ASR/TTS products are added/deleted, it should have minimal effects on the integration of other ASR/TTS products.

As was mentioned at the beginning of this document, that the SRM Server communicates with the SRM Client using MRCP. In addition to starting, monitoring and managing these ASR/TTS clients, the SRM server accepts MRCP requests for ASR/TTS resources, and forwards these requests to the appropriate ASR/TTS clients. The communication between SRM Server and the ASR/TTS client uses a proprietary VoiceGenie messaging format. However, in order to continue support for functionalities currently not available by the MRCP protocol, the SRM Server can accept some vendor-specific parameters so such functionalities are maintained.

3.2 Directory Structure

As a SRM server can be deployed in a separated system from Media platform, we first describe the directory structure to give users an overview where the SRM server software and configuration files located.

This section describes the directory structure for the SRM server (linux version). For the Windows version, simply replace the root install directory of the Linux version (/usr/local/srm-server/) by the root install directory (C:\VoiceGenie\srm-server\).

The following directories are used in for the SRM Server:

Directory	Description
/usr/local/srm-server/bin	This directory contains the executable binaries for the SRM server, as well as all the ASR/TTS clients to be started by the SRM Server.
/usr/local/srm-server/lib	This directory contains the shared libraries to be loaded by the SRM Server.
/usr/local/srm-server/config	This directory contains the configuration files to be used for the SRM server as well as all the ASR/TTS clients.
/usr/local/srm-server/logs	This directory is where the logs from the SRM server as well as the ASR/TTS clients are written to.
/usr/local/srm-server/tmp	This directory is where the SRM Server will write its temporary files to.

3.3 Application Server

Using the convention established by the MRCP standard, the SRM server provides the utterance from a recognition session by way of an HTTP URL. As a result, a web server is installed and configured on the Media Platforms. An Apache server is used on Linux for this purpose, while an IIS server is used on Windows Platforms.

3.4 Provisioning

Via the OA&M Framework SMC, users can provision the SRM server to make use of the various ASR and TTS engines by managing the corresponding client process. The provisioning information resides in the /usr/local/srm-server/config/srmsserver.cfg file. This file should not be modified directly.

The provisioning parameters are organized in a tree-like structure, where the top-level parameter(s) are used to define the other parameters to be used and read.

Please refer to the [SRM System Reference Guide](#) for a full list of all SRM Server, ASR/TTS clients and OSR 3.0 configuration parameters.

3.5 Metrics and Logging

Metrics data is currently written to file(s) on the platform. All Metrics/logging information can be found in the pw_metricsfile under /usr/local/phoneweb/logs (linux) or (\$INSTALLROOT)\mp\logs (windows) directory.

Each record has the following fields:

Field	Contents
Timestamp	Timestamp structured as yyyy-mm-dd/hh:mm:ss.mmm. Time is 24 hour, based on the platform timezone.
Record Type	Always METRIC
Session ID	Globally unique session identifier.
Operation	incall_initiated, incall_begin, incall_end, incall_reject, bridge_initiated, bridge_begin, bridge_reject, call_initiated, call_begin, call_end, call_reject, outcall_requested, outcall_initiated, outcall_begin, outcall_end, outcall_reject, transfer_initiated, transfer_connected, transfer_result, call_reference
Operation Data	Data specific to the record

The following are the metrics entries logged by the SRM Server.

Note:

1. An ASR metrics has a session id suffixed with "-asr" to VG call ID.
2. A TTS metrics has a session id suffixed with "-tts" to VG call ID.
3. A state in an ASR metrics can be one of "UNAVAILABLE", "INIT", "INIT_READY", "INIT_LOADED", "IDLE", "USED", "LOADING", "LOADED", "STARTING", "RECOGNIZING", "STOPPING" and "UNLOAD".
4. A state in a TTS metrics can be one of "UNAVAILABLE", "IDLE", "USED", "SPEAK", "SPEAKING" and "STOPPING".

Please refer to the [SRM System Reference Guide](#) for the list of all SRM Server entries, their causes and recommended actions.

3.6 CLC and SMC Operations

3.6.1 CLC command to start/stop srmsserver:

stop srmsserver <localhost> <instance>
to stop the srmsserver process

start srmserver <localhost> <instance>

to start the srmserver process.

3.6.2 CLC commands to change srmserver operation states:

The srmserver has the following three operation states:

ACTIVE:

In this state srmserver accepts client request, allocates right resource and maintains a media (TTS/ASR) session.

SUSPENDED:

In this state the srmserver continues to service the existing sessions, until the sessions complete. At the same time it will not accept new requests from client.

DISABLED:

In the state srmserver will terminate all the on going sessions and not accept any new requests.

suspend srmserver <host> <instance> <>false>

to transfer srmserver from either DISABLED or ACTIVE to SUSPENDED

suspend srmserver <host> <instance> true

to transfer srmserver from either SUSPENDED or ACTIVE to DISABLED

resume srmserver <host> <instance>

to transfer srmserver from either SUSPENDED or DISABLED to ACTIVE

The SMC has the similar operations as the above CLC commands in the Operation->Start/Stop Software page. The only difference is that SMC cannot transfer the srmserver state to DISABLED

3.6.3 Other CLC commands:

clearstats srmserver <localhost> <instance>

to reset all the statistic data of each host resource and module.

clienttrace [service] <host> <instance> [asr/tts engine name] <lev>:

to enable and disable trace of a hosted TTS/ASR client process

where:

[asr/tts engine name] can be any of the engines displayed in the srmserver health information.
<lev> must be 'enable' or 'disable'

4 SRM RTSP-TTS Engine

The Real Time Streaming Protocol (RTSP, described in <http://www.ietf.org/rfc/rfc2326.txt>) is a popular protocol for streaming real-time media from a content server to listeners. VoiceGenie Media Platform will communicate with the SRM Server to make MRCP TTS requests. The RTSP-TTS Server then translates the incoming MRCP TTS requests into RTSP PLAY requests and forwards them to the RTSP Stream Server.

For details regarding the SRM RTSP-TTS Engine, please refer to the [VoiceGenie 7 RTSP Users' Guide](#).

5 SRM TTY/TDD Engine

TTY, also known as TDD, enables people with hard hearing capability to communicate over telephone lines. These instruments use a typewriter style with a display to allow users to communicate over text messages. TDD/TTY is based on frequency shift keying (FSK), in which data is transmitted character-at-a-time, with a 5-bit character set, using a pair of frequencies (1800Hz and 1400Hz) to represent zero and one respectively. TDD/TTY operates at two rates; one at 20ms per bit (50 baud), and the other at 22ms per bit (45.45 baud).



Note: Only the mulaw audio format is supported for TTY/TDD

For details regarding the SRM TTY/TDD Engine, please refer to the [VoiceGenie 7 TDD Users' Guide](#).