Yandex
SpeechKit

Руководство по интеграции системы распознавания и синтеза речи Яндекс SpeechKit Box

с платформой интерактивного речевого взаимодействия на базе Asterisk

Яндекс

speechkit.yandex.ru

Содержание

1. Введение

1.1 Сведения о документе

В настоящем д окументе представлена техническая информация, необходимая для осуществления интеграции системы распознавания речи Яндекс SpeechKit Box с платформой интерактивного голосового взаимодействия (IVR) на базе программных средств Asterisk. В документе описаны конфигурационные настройки системы.

Документ предназначен для инженерного состава, осуществляющего установку и настройку систем голосового взаимодействия.

Документ не содержит сведений об установке и первичной настройке системы распознавания речи Яндекс SpeechKit Box, а также об установке и первичной настройке платформы интерактивного голосового взаимодействия Asterisk.

1.2 Условные обозначения

AGI — Asterisk Gateway Interface. Интерфейс Asterisk.
ASR — Automatic Speech Recognition. Сервис распознавания речи.
DN — Directory Number. Телефонный номер, назначаемый на тот или иной сервис.
IVR — Interactive Voice Response. Платформа интерактивного голосового взаимодействия.
MRCP — Media Resource Control Protocol. Протокол управления медиаресурсами.
NLSML — Natural Language Semantics Markup Language. Язык разметки для представления результатов распознавания.
SSML — Speech Synthesis Markup Language. Язык разметки синтеза речи.
TTS — Text-to-Speech. Сервис синтеза речи.
VXML — Voice XML. XML-файлы с распознаванием голоса.

1.3 Требования к инженерному составу

Инженер, осуществляющий интеграцию системы распознавания и синтеза речи Яндекс SpeechKit Box с IVR на базе Asterisk, должен обладать следующей минимальной квалификацией:

- иметь опыт установки и обслуживания систем IVR, организованных на базе программных средств Asterisk;
- иметь опыт установки дополнительных пакетов в инфраструктуру Asterisk;
- уметь работать в командной строке Linux;
- уметь изменять конфигурационные файлы в командной строке Linux;
- понимать принципы взаимодействия в IP-сетях.

2. Краткое описание интегрируемых систем

2.1 Платформа интерактивного голосового взаимодействия на базе Asterisk

Asterisk — это программная АТС, которая сопрягается с Яндекс SpeechKit Box через программный модуль UniMRCP.

UniMRCP — это кроссплатформенное программное обеспечение с открытым исходным кодом, способное выполнять функции MRCP-клиента и MRCP-сервера. Модуль UniMRCP может быть установлен на платформе RedHat/CentOS 5.x, 6.x, а также на Debian, Ubuntu и Windows. Поддерживаются версии Asterisk 1.6, 1.8, 10, 11, 12, 13, а также протокол MRCP версий 1 и 2.

Совместимость между каждым конкретным дистрибутивом Asterisk и Яндекс SpeechKit Box следует рассматривать как совместимость между дистрибутивом Asterisk и модулем UniMRCP. Взаимодействие между модулем UniMRCP и Яндекс SpeechKit Box происходит через протокол MRCP и не зависит от дистрибутива Asterisk.

Настоящая документация адаптирована к дистрибутиву AsteriskNow 13, распространяемому компанией Digium, и UniMRCP 1.3.1. В случае использования других дистрибутивов данная документация может быть не точна.

Производитель: Digium Inc
Российский представитель
AsteriskNow

2.2 Система распознавания речи Яндекс SpeechKit Box

Яндекс SpeechKit Box — это система генерации и распознавания речи, базирующаяся на технологии свободных грамматик и позволяющая создавать высоконагруженные сервисы с функциями распознавания и синтеза речи, а также смыслового разбора сказанного.

Распознавание речи возможно на русском и английском языках. Стандартные языковые модели для русского языка: короткие запросы, адреса, музыка, даты, имена, числа, заказы, тексты. Синтез русской речи: четыре стандартных голоса (два мужских, два женских).

Производитель: ООО «Яндекс»

3. Архитектура решения

Программный комплекс Asterisk включает в себя:

- IP-АТС;
- систему автоматического распределения вызовов (ACD);
- систему автоматической обработки вызовов (IVR).

В платформе Asterisk ядро системы, обеспечивающее внутреннюю взаимосвязь между компонентами, отделено от модулей: протоколов, кодеков и аппаратных интерфейсов. Такой подход позволяет использовать не только технологии, существующие сейчас, но и те, которые появятся в будущем.

Ядро системы Asterisk обеспечивает следующие функциональные уровни:

- Коммутация. Позволяет выполнять коммутацию каналов (вызовов, сервисов), как это делает обычная АТС или коммутатор. Задача этого уровня — создание соединения между двумя или несколькими направлениями, не учитывая наличия и специфику многих аппаратных или программных интерфейсов.
- Выполнение прикладных задач. Этот уровень обеспечивает выполнение задач (как встроенных, так и назначенных пользователем) для предоставления нужной услуги (например: голосовая почта, конференц-связь, проигрывание голосовых сообщений, а также работа с базой данных).
- Преобразование кодирования. Благодаря загружаемым функциональным модулям данный уровень гарантирует совместимость коммутируемых каналов по типу кодирования речи или видеоизображения.
- Менеджер планирования задач и каналов ввода/вывода. Этот уровень управляет взаимодействием между модулями, обеспечивающими выполнение различных прикладных задач.

Существует четыре независимых интерфейса, которые обеспечивают взаимодействие модулей на программном и аппаратном уровнях (см. рисунок 1). Использование такой модульной системы позволяет ядру Asterisk не зависеть от деталей: как соединён абонент, какой выбран кодек, и т. п.

- Интерфейс канала. Управляет типом соединения, которое использует абонент, будь то VoIP- соединение, ISDN, PRI или соединение по любой другой технологии. Динамически загружаемые модули управляют низшим уровнем этих соединений.
- Интерфейс приложений. Позволяет выполнять различные модули для реализации услуг АТС Конференция, пейджинг, просмотр содержимого каталогов, голосовая почта, передача данных и многие другие функции АТС выполняются при помощи отдельных модулей.
- Интерфейс кодеков. Загружает модули кодеков для поддержки различных форматов сжатия аудио потока (GSM, Mu-Law, A-law, G723, G729, iLBC, speech, MP3).
- Интерфейс формата файлов. Управляет записью и чтением различных форматов файлов.

Таким же образом внедряется модуль UniMRCP, позволяющий добавить подключение к ASR/TTS серверу посредством протокола MRCP.

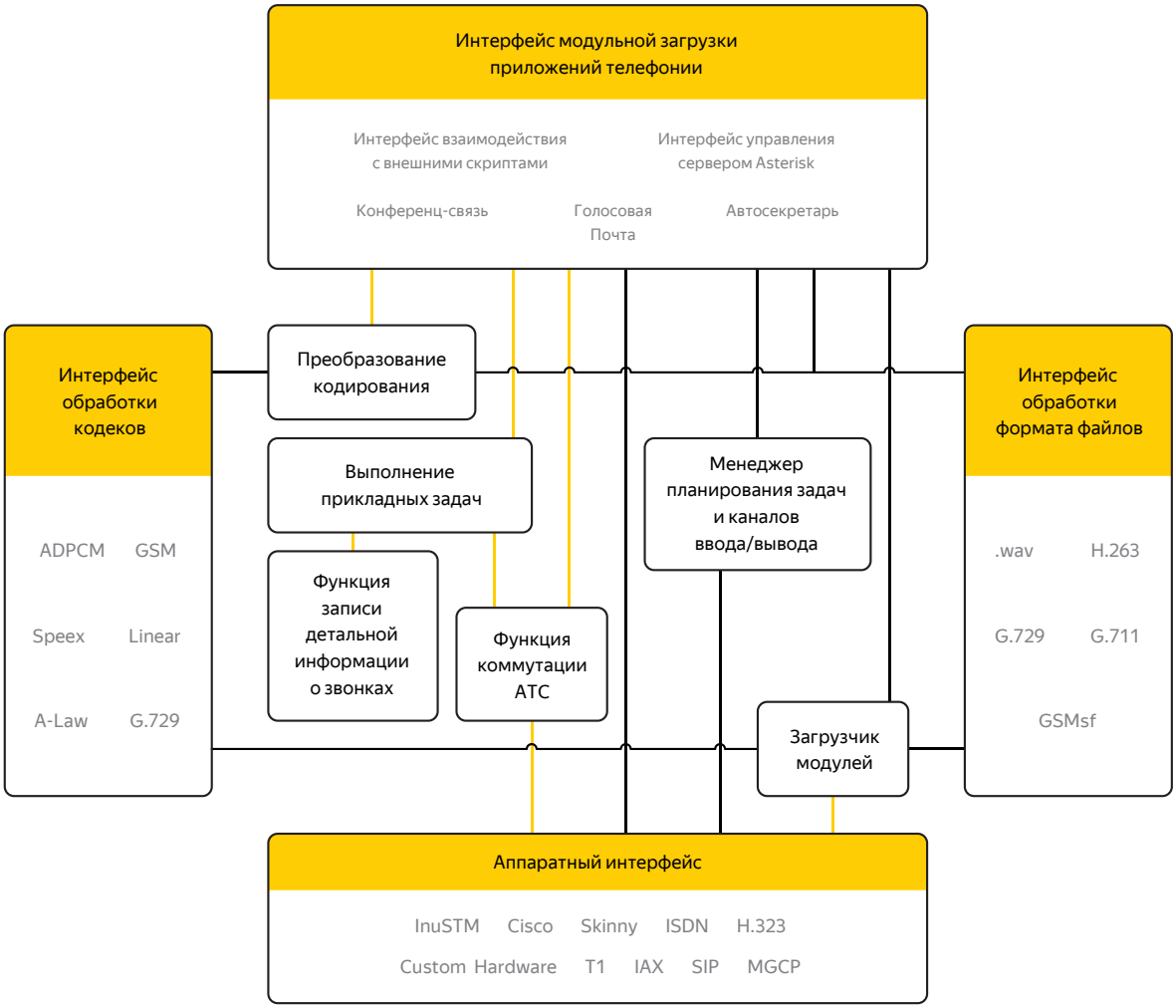

Рисунок 1. Архитектура решения Asterisk

Звонящий из городской сети общего пользования через IP-АТС, инициируя звонок на некий номер, маршрутизируется на систему Asterisk. Далее, согласно набранному номеру, запускается приложение автоматического взаимодействия со звонящим. В зависимости от задачи подключаются ресурсы ASR или TTS.

Яндекс SpeechKit Box устанавливается как отдельный сервер, подключаемый к той же самой маршрутизируемой сети IP. Во избежание задержек на маршрутизаторах и потенциальных ограничений портов на встроенных файрволах рекомендуется устанавливать Яндекс SpeechKit Box в той же самой сети L2. Протокол MRCP используется между программным комплексом Asterisk и Яндекс SpeechKit Box для управления ресурсами распознавания и синтеза речи.

4. Установка модуля UniMRCP в Asterisk

Установка программного обеспечения UniMRCP должна производиться на компьютер с операционной системой Linux, на котором установлено функционирующее программное обеспечение Asterisk версии 1.6, 1.8, 10, 11, 12 или 13.

Все пакеты с исходным кодом доступны на UniMRCP или в репозитории github.

При подписке на UniMRCP возможно получить уже готовые RPM-пакеты для установки UniMRCP на Redhat и CentOS версий 5 и 6.

В разделе описывается процесс установки UniMRCP из пакета с исходным кодом на RedHat или CentOS версии5 или 6. Установка на Debian или Ubuntu проходит подобным образом.

Чтобы установить модуль UniMRCP в Asterisk, выполните следующие действия.

1. Остановите Asterisk из консоли командой core stop now.
2. Установите необходимые инструменты для компиляции пакетов с исходным кодом командой yum groupinstall «Development Tools».
3. Установите пакеты с исходным кодом Asterisk командой yum install asterisk-devel.
4. Загрузите и скопируйте во временный каталог Asterisk последнюю версию пакета.
5. Распакуйте пакет командой tar –xf.
6. Клонируйте git-репозиторий, содержащий пакет с исходным кодом UniMRCP командой git clone https://github.com/unispeech/unimrcp.git.
7. Распакуйте архив, полученный на предыдущем шаге. Скопируйте содержимое каталога, полученного после распаковки архива, в каталог unimrcp командой cp –r * /tmp/unimrcp.
8. Перейдите в каталог unimrcp и запустите скрипт build-dep-libs.sh.
9. Выполните команду ./bootstrap.
10. Выполните команду ./configure.
11. Выполните команду make.
12. Выполните команду make install.

В результате выполнения этих команд в каталоге /usr/local/unimrcp должны быть созданы каталоги bin, conf, data, include, lib, log, plugin, var, содержащие как конфигурационные файлы, так и исполняемые.

13. Вернитесь в ваш временный каталог и загрузите пакет коннектора между UniMRCP и Asterisk командой git clone https://github.com/unispeech/asterisk-unimrcp.git
14. Перейдите в каталог asterisk-unimrcp и выполните команду ./bootstrap
15. Выполните команду ./configure
16. В Asterisk есть защита от запуска неизвестных модулей. На шаге 3 устанавливается ПО для компиляции нового (неизвестного) UniMRCP-модуля. Чтобы обойти защиту, перед сборкой модулей из пакета asterisk-unimrcp следует использовать данные уже собранных модулей из каталога /usr/lib/asterisk/modules. В конце каждого файла .so есть идентификатор сборки, подобный этому: b339725e5e804e591435c0af54be8ae7. Добавьте идентификатор в файл /usr/include/asterisk/buildopts.h в строку #define AST_BUILDOPT_SUM вместо имеющегося. В результате компилятор будет собирать новые модули уже с реально работающим в вашей системе идентификатором.
17. Выполните команду make install.
    В результате установки в каталоге cd /usr/lib/asterisk/modules/ должны появиться файлы app_unimrcp.so и res_speech_unimrcp.so.
18. После запуска Asterisk выполните команду module show и убедитесь, что модули app_unimrcp.so и res_speech_unimrcp.so находятся в состоянии Running.

5. Установка модуля UniMRCP в Asterisk

Для сопряжения модуля UniMRCP с программными средствами Яндекс SpeechKit Box, требуется внести изменения в конфигурационный файл mrcp.conf, который после установки пакета asterisk-unimrcp располагается по адресу /etc/asterisk/mrcp.conf.

После редактирования файла потребуется перезапустить Asterisk.

Описание основных параметров файла конфигурации mrcp.conf представлено ниже:

Параметр	Описание
default-asr-profile	Профиль распознавания, используемый по умолчанию.
default-tts-profile	Профиль синтеза, используемый по умолчанию.
version	Версия протокола MRCP.
server-ip	Адрес MRCP-сервера.
server-port	Порт MRCP-сервера для приема запросов.
resource-location	Название группы ресурсов.
speechsynth	Название ресурса синтеза.
speechrecog	Название ресурса распознавания.
rtp-ip	Адрес интерфейса, с которого будет сниматься RTP-трафик Следует указать адрес интерфейса на Asterisk, который находится в той же сети, что и Яндекс SpeechKit Box. Именно на этот адрес будут направляться RTP-пакеты с MRCP-сервера.
rtp-port-min rtp-port-max	Диапазон RTP-портов (для передачи RTP-трафика).
max-connection-count	Число портов, лицензированных на Яндекс SpeechKit Box.
server-port	Порт MRCP-сервера для приема MRCP-пакетов. Для Яндекс SpeechKit Box и MRCP V2 следует указать порт 8060. Для MRCP V1 следует указать порт 1554.
client-ip	IP-адрес клиента (Asterisk). Следует указать адрес интерфейса на Asterisk, который находится в той же сети, что и Яндекс SpeechKit Box.
sip-transport	SIP-протокол: UDP, TCP. По умолчанию используется UDP (это же значение по умолчанию установлено в Яндекс SpeechKit Box). Однако, если если это значение для Яндекс SpeechKit Box изменено на TCP, то и в этом поле ее следует поменять.

Описание основных параметров mrcp.conf

Если MRCP-сервер находится в публичном интернете, следует установить в величины client-ext-ip и rtp-ext-ip публичный IP-адрес.

Пример файла mrcp.conf, содержащий рекомендованные настройки, представлен ниже:

```
[general]
; Default ASR and TTS profiles.
default-asr-profile = yandexspeechbox-mrcp2
default-tts-profile = yandexspeechbox-mrcp2
; UniMRCP logging level to appear in Asterisk logs. Options are:
; EMERGENCY|ALERT|CRITICAL|ERROR|WARNING|NOTICE|INFO|DEBUG -->
log-level = DEBUG
max-connection-count = 100
offer-new-connection = 1
; rx-buffer-size = 1024
; tx-buffer-size = 1024
; request-timeout = 5000
;
; Profile for Nuance Speech Server MRCPv2
;
[yandexspeechbox-mrcp2]
; MRCP version.
version = 2

; === SIP settings ===
; Must be set to the IP address of the MRCP server.
server-ip = 192.168.0.30
; SIP port on the MRCP server.
server-port = 8060
; server-username = test
force-destination = 1

; === SIP agent ===
; client-ip = 0.0.0.0
; Must be set to the IP address of the MRCP client.
client-ip = 192.168.0.29
; client-ext-ip = auto
; SIP port on the MRCP client.
client-port = 25097
; SIP transport either UDP or TCP.
sip-transport = udp
; ua-name = Asterisk
; sdp-origin = Asterisk
; sip-t1 = 500
; sip-t2 = 4000
; sip-t4 = 4000
; sip-t1x64 = 32000
; sip-timer-c = 185000

; === RTP factory ===
; rtp-ip = 0.0.0.0
; Must be set to the IP address of the MRCP client.
rtp-ip = 192.168.0.29
; rtp-ext-ip = auto
; RTP port range on the MRCP client.
rtp-port-min = 28000
rtp-port-max = 29000

; === Jitter buffer settings ===
playout-delay = 50
; min-playout-delay = 20
max-playout-delay = 200

; === RTP settings ===
ptime = 20
codecs = PCMU PCMA L16/96/8000 telephone-event/101/8000

; === RTCP settings ===
rtcp = 1
rtcp-bye = 2
rtcp-tx-interval = 5000
rtcp-rx-resolution = 1000
```

Файл mrcp.conf

6. Использование модуля UniMRCP в Asterisk

В процессе синтеза и распознавания речи Asterisk и Яндекс SpeechKit Box взаимодействуют с помощью приложения SynthAndRecog.

Синтаксис: SynthAndRecog (prompt, grammar, options)

Это приложение устанавливает две MRCP-сессии: одну для синтеза речи, вторую для распознавания речи. Как только пользователь начинает говорить, сессия синтеза останавливается и начинается сессия распознавания. Как только распознавание заканчивается, приложение выходит и возвращает результат в dialplan Asterisk.

Входные параметры:

- prompt. Текст, SSML-контент (см. пример 1), файл или URI-ссылка, используемые для синтеза.
- grammar. Текст грамматики или URL файла грамматики, используемой для распознавания.
- options. Список допустимых значений приведен на странице.

Возвращаемые параметры:

- recog_status. Может принимать следующие значения:
  OK — распознавание состоялось;
  ERROR — распознавание не может быть начато;
  INTERRUPTED — звонящий положил трубку, в процессе распознавания.
- recog_completion_cause. Показывает, чем закончилось распознавание. Может принимать следующие значения:
  000 — состоялось;
  001 — не соответствует;
  002 — не введено.
- recog_result. Если распознавание закончено успешно, то переменная recog_result содержит результат распознавания, полученный от MRCP-сервера в формате NLSML (см. примечание 2). Альтернативно результат распознавания может быть получен при использовании следующих функций dialplan: recog_confidence(), recog_grammar(), recog_input(), и recog_instance().

Яндекс SpeechKit Box использует для работы свободные грамматики и языковые модели, поэтому нет необходимости передавать в SRGS-грамматике все компоненты. Но указать грамматику в MRCPRecog необходимо согласно протоколу MRCP. Для упрощения работы и сокращения dialplan в Asterisk рекомендуется выложить файл грамматик на доступный web-сервер и при обращении к нему указывать его URL.

Языковая модель для распознавания определяется в файле /usr/local/unimrcp/conf/defaultgramma.xml на сервере Яндекс SpeechKit Box.

Язык для синтеза речи определяется в файле //usr/local/unimrcp/conf/defaultSpeak.xml на сервере Яндекс SpeechKit Box.

Пример файла грамматик приведен ниже:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<grammar xmlns="http://www.w3.org/2001/06/grammar" mode="voice"
root="Mygrammar" tag-format="semantics/1.0-literals" version="1.0" xml:lang=
"ru-ru">
        <rule id="Mygrammar" scope="public">
            <ruleref uri="#yandex"/>
        </rule>
        <rule id="yandex" scope="private">
            <one-of>
                <item>topic<tag>numbers</tag>
                </item>
                <item>lang<tag>ru-RU</tag>
                </item>
                </one-of>
        </rule>
</grammar>
```

Файл грамматик

Примеры запуска сессии синтеза и распознавания в dialplan приведены ниже.

Пример 1

Этот пример демонстрирует, как использовать приложение SynthAndRecog() с SSML prompt и SRGS XML DTMF грамматикой:

```
exten => s,1,Answer
exten => s,n,SynthAndRecog(<?xml version=\"1.0\"?><speak version=\"1.0\"
xmlns=\"http://www.w3.org/2001/10/synthesis\" xmlns:xsi=\"http://www.
w3.org/2001/XMLSchema- instance\" xsi:schemaLocation=\"http://www.
w3.org/2001/10/synthesis http://www.w3.org/TR/speech- synthesis/synthesis.
xsd\" xml:lang=\"ru-RU\">Пожалуйста выберите цвет: для красного нажмите
1 <break/> для зеленого нажмите 2 <break/> для голубого нажмите 3 </
speak>,<?xml version=\"1.0\"?><grammar

xmlns=\"http://www.w3.org/2001/06/grammar\" version=\"1.0\" mode=\"dtmf\"
tag- format=\"semantics/1.0-literals\" root=\"color\"><rule id=\"color\"><one-
of><item>1<tag>красный</tag></item><item>2<tag>зеленый</
tag></item><item>3<tag>голубой</tag></item> </one-of></rule></
grammar>,t=5000&b=1&ct=0.7)
exten => s,n,Verbose(1, ${RECOG_STATUS}, ${RECOG_COMPLETION_CAUSE},
${RECOG_RESULT})
exten => s,n,Hangup
```

Пример 1

Пример 2

Этот пример демонстрирует, как использовать приложение SynthAndRecog() с prompt и грамматикой, получаемыми из файлов:

```
exten => s,1,Answer
exten => s,n,SynthAndRecog(/usr/local/unimrcp/data/speak.xml,/usr/local/unimrcp/
data/grammar.xml,t=5000& b=1&ct=0.7&spl=ru-RU)
exten => s,n,Verbose(1, ${RECOG_STATUS}, ${RECOG_COMPLETION_CAUSE},
${RECOG_RESULT})
exten => s,n,Hangup
```

Пример 2

Пример 3

Этот пример демонстрирует, как использовать приложение SynthAndRecog() с грамматикой, загружаемой по HTTP-ссылке:

```
exten => s,1,Answer
exten => s,n,SynthAndRecog(/usr/local/unimrcp/data/speak.xml,/usr/local/unimrcp/
data/grammar.xml,t=5000& b=1&ct=0.7&spl=ru-RU)
exten => s,n,Verbose(1, ${RECOG_STATUS}, ${RECOG_COMPLETION_CAUSE},
${RECOG_RESULT})
exten => s,n,Hangup
```

Пример 3

Более подробная информация по ссылке.
Примечание 1, Примечание 2.

Yandex
SpeechKit