
VoiceKey.IVR

Программный компонент голосового
взаимодействия для IVR-систем

STC-S9726.02

Обзор VoiceKey.IVR

НЦДА.00744-01 98

Версия документа: 1-080724

СОДЕРЖАНИЕ

1. ОБЩИЕ СВЕДЕНИЯ	6
1.1 Основные сведения	6
1.2 Служба технической поддержки	6
1.3 Назначение	6
2. О ПРОДУКТЕ	7
2.1 Применение комплекса	7
3. КОМПОНЕНТЫ	8
3.1 Архитектура	8
3.2 Сервисы VoiceKey.PLATFORM	8
3.2.1 Monitoring Component	8
3.2.2 LoadBalancer Component	9
3.2.3 Configurator Component	9
3.2.4 Chronicler Component	9
3.2.5 SessionService Application	10
3.2.6 MediaHub Component	10
3.2.7 VoiceKey Portal	10
3.2.8 stc-net-license	10
3.2.9 MediaStorage Component	11
3.2.10 Reporter Component	11
3.2.11 Router Component	11
3.2.12 Security Component	12
3.3 Сервисы VoiceKey.IVR	12
3.3.1 Siphon Application	12
3.3.2 MRCP Application	12
3.3.3 Dialog Platform	13
3.3.4 VoiceRecognizer Processor	13
3.3.5 TextTransform Processor	13
3.3.6 TTS Processor	13
3.3.7 Vocabulary Application	13
3.3.8 Buzzer Application	13
3.3.9 TTSCloud Application	13

3.3.10 ASRCloud Application	14
3.3.11 Ripper Component	14
4. РЕШЕНИЯ НА БАЗЕ VOICEKEY.IVR	15
4.1 VoiceKey.IVR SIP	16
4.1.1 Компоненты, необходимые для решения	17
4.2 VoiceKey.IVR MRCP	18
4.2.1 Компоненты, необходимые для решения	19
4.3 VoiceKey.IVR ASR	20
4.3.1 Компоненты, необходимые для решения	21
4.4 VoiceKey.IVR TTS	22
4.4.1 Компоненты, необходимые для решения	23
4.5 VoiceKey.IVR ASR gRPC	24
4.5.1 Компоненты, необходимые для решения	24
4.6 VoiceKey.IVR TTS gRPC	25
4.6.1 Компоненты, необходимые для решения	25

ВВЕДЕНИЕ

Общие положения

Настоящий документ предназначен для ознакомления с функциями и принципами работы программного компонента голосового взаимодействия для IVR-систем **VoiceKey.IVR** (далее — комплекс **VoiceKey.IVR**).

Сведения, приведённые в документе, относятся к версии 4.14 комплекса **VoiceKey.IVR**.

Соглашения и обозначения

В документе приняты следующие типографские соглашения:

Формат текста	Описание
Обычный	Основной текст документа
Полужирный	Наименования компаний, программ и программных компонентов, а также наименования элементов интерфейса (заголовки, кнопки и т. п.)
<i>Полужирный курсив</i>	Имена файлов и пути доступа к ним
Моноширинный	Параметры конфигурации, параметры API, тексты примеров API
<u>Подчёркнутый бирюзовый</u>	Внешние ссылки и ссылки на разделы документа
Элемент > Вложенный элемент	Выбор пункта меню. Например, текст Файл > Выход означает: выбрать меню Файл , а затем — команду Выход из меню Файл
<метка>	Фрагмент текста, который должен быть заменён реальным значением. Например, в записи <i>D:\<DirName>\file.txt</i> текст <i><DirName></i> используется для обозначения названия каталога в файловой системе



Обязательные действия, невыполнение которых может привести к ошибке.



Примечания, рекомендации и прочие сведения.



Ссылки на другие документы.

Товарные знаки

Названия компаний и названия продуктов, упомянутые в документе, являются собственностью их соответствующих владельцев.

1. ОБЩИЕ СВЕДЕНИЯ

1.1 Основные сведения

Наименование: Программный компонент голосового взаимодействия для IVR-систем
VoiceKey.IVR

Условное наименование: STC-S9726.02

Обозначение: НЦДА.00744-01

Изготовитель: Общество с ограниченной ответственностью «ЦРТ-инновации»

Адрес: 194044, Санкт-Петербург, ул. Гельсингфорсская, д. 3, к. 11 лит. Д, пом. 229

Телефон: +7 (812) 325-88-48

Факс: +7 (812) 327-92-97

1.2 Служба технической поддержки

Адрес службы сервисного обслуживания и технической поддержки в интернете:

Электронная почта: info@speechpro.com

Адрес в сети Интернет: <http://www.speechpro.ru/support>

При обращении в службу технической поддержки необходимо представить четкое описание возникшей проблемы.

Предварительно подготовьте следующую информацию:

- сведения о конфигурации серверов;
- название и версии используемых операционных систем;
- версия программного обеспечения.

1.3 Назначение

VoiceKey.IVR — программный компонент голосового взаимодействия с диалоговыми ассистентами и IVR-системами, обеспечивающий синтез и распознавание речи, управление исходящими оповещениями и обработку входящих вызовов.

2. О ПРОДУКТЕ

2.1 Применение комплекса

VoiceKey.IVR находит широкое применение в различных сферах, в том числе:

- в сфере предоставления услуг при построении систем голосового дистанционного обслуживания в контактных центрах, интерактивных сервисах и виртуальных ассистентах с голосовым управлением;
- в сфере финансов, межведомственного электронного взаимодействия, банковской сфере, телекоме и предоставления государственных услуг, в корпоративных информационных системах.

3. КОМПОНЕНТЫ

3.1 Архитектура

Комплекс имеет двухуровневую архитектуру.

Базовым уровнем комплекса является технологическая платформа **VoiceKey.PLATFORM**, в состав которой входят: сервисы, обработчики, хранилище данных.

Сервисы обрабатывают запросы, поступающие от веб-сервисов, распределяют их между обработчиками и управляют передачей сообщений между компонентами.

Обработчики обеспечивают выполнение поступивших в технологическую платформу запросов.

Хранилище медиаданных используется для сохранения аудиозаписей.

Информационное взаимодействие компонентов технологической платформы обеспечивает шина данных и gRPC-интерфейс.

Прикладной уровень комплекса представлен набором отдельных программных решений, использующих функциональность технологической платформы для решения специфических задач, требующих распознавания и синтеза речи. Каждое программное решение реализовано в виде веб-сервиса, предоставляющего прикладной программный интерфейс.

Взаимодействие веб-сервисов с технологической платформой осуществляется через программный интерфейс на языке Java.

В соответствии с требованиями к производительности и резервированию, модули комплекса могут устанавливаться на одном или нескольких серверах. Поддерживается работа как на физических, так и виртуальных серверах.

3.2 Сервисы VoiceKey.PLATFORM

3.2.1 Monitoring Component

Monitoring Component (менеджер мониторинга) реализует систему технологического мониторинга компонентов.

Monitoring Component регистрирует компоненты в системе мониторинга и собирает данные о технологическом состоянии компонентов комплекса.

После старта каждый компонент комплекса регистрируется в **Monitoring Component** (отправляя RegisterMonitoringComponentMessage). Если в системе мониторинга компонент с таким instanceId уже существует, то все метрики и статистики переносятся в новый компонент.

После регистрации в системе мониторинга, компонент начинает отправлять keepAlive сообщения (KeepAliveMessage) менеджеру мониторинга, а также регистрирует пользовательские и системные метрики (отправляя RegisterMetricsMessage). Затем отправляет значения данных метрик (MetricsValueMessage) с заданным интервалом.

Monitoring Component предоставляет информацию о текущих значениях метрик в формате json или в формате prometheus.

3.2.2 LoadBalancer Component

LoadBalancer Component (балансировщик загрузки) обеспечивает распределение поступающих на вход запросов между обработчиками в соответствии с их текущей загруженностью.

LoadBalancer Component предназначен для:

- получения поступивших на обработку запросов веб-сервисов;
- разбора поступивших запросов по операционной специфике;
- распределения и отправки запросов на доступные обрабатывающие узлы;
- сбора поступающих с обрабатывающих узлов результатов;
- формирования итоговых ответов на обрабатываемые запросы;
- отправки результатов выполнения запросов веб-сервисам.

Принципы работы

- выход из строя любого обработчика не влияет на общую работоспособность системы;
- при увеличении нагрузки на обрабатывающие узлы система продолжает обрабатывать поступающие запросы без задержек;
- при предельной загрузке обрабатывающих узлов обеспечивается возможность подключения дополнительных узлов обработки.

3.2.3 Configurator Component

Configurator Component (компонент конфигурации) реализует функцию централизованного управления конфигурациями всех компонентов.

Компонент конфигурации обеспечивает централизованное хранение настроек всех компонентов комплекса. Настройки компонентов хранятся в общем репозитории конфигураций на том сервере, где установлен компонент конфигурации, в виде набора параметров, сгруппированных в секции.

Компоненты при первом запуске получают свои настройки от компонента конфигурации и сохраняют их в локальные файлы конфигурации.

Параметры конфигурации имеют иерархическую структуру:

- на глобальном уровне хранятся значения параметров общие для всех компонентов;
- на уровне компонентов хранятся значения параметров общие для компонентов определённого вида;
- на уровне экземпляров хранятся настройки специфические для конкретного экземпляра компонента.

3.2.4 Chronicler Component

Chronicler Component (компонент журналирования) регистрирует события, возникающие в ходе работы компонентов.

Принципы работы

- компоненты при первом запуске передают компоненту журналирования информацию о структуре записи, формируемой компонентом по факту возникновения значимых, подлежащих регистрации в журнале, событий;
- компонент журналирования создаёт в базе данных журналы, соответствующие переданным компонентами структурам записей;
- в ходе работы компонент журналирования регистрирует новые события, которые ему отправляют компоненты, в соответствующих журналах.

3.2.5 SessionService Application

SessionService Application — веб-сервис управления сессиями.

3.2.6 MediaHub Component

MediaHub Component (медиаконцентратор) реализует функцию коммуникации внешнего источника медиаданных и целевых операционных сервисов для решения поставленной задачи.

Принципы работы

- медиаконцентратор для подключения к источнику медиаданных использует настройки, полученные от управляющего сессией компонента;
- вместе с настройками источника медиаконцентратор получает настройки потребителя и описание задачи, последовательность операций, которые требуется выполнить в рамках текущей сессии;
- медиаконцентратор обеспечивает подключение и вызов всех необходимых для выполнения задачи сервисов.

3.2.7 VoiceKey Portal

VoiceKey.PLATFORM — пользовательский интерфейс для: **VoiceKey.TTS**, **Vocabulary Application**, **MediaStorage Component**, **Security Component**, **Configurator Component**, **Reporter Component**.

3.2.8 stc-net-license

stc-net-license (менеджер лицензий) реализует функцию универсального провайдера лицензий.

Менеджер лицензий обеспечивает централизованное хранение информации о приобретённых лицензиях и предоставление сведений о наличии или отсутствии лицензии на использование заданного сервиса.

Принципы работы

- клиенты напрямую не работают с носителями лицензионных ключей и запрашивают именованные лицензии у менеджера лицензий;
- для оперативного предоставления клиентам информации об имеющихся лицензиях, извлечённые из лицензионных ключей лицензии именуются и сохраняются в кэш менеджера лицензий;

- текущее доступное количество лицензий хранится в кэше менеджера лицензий.

3.2.9 MediaStorage Component

MediaStorage Component (хранилище медиаданных) — хранилище медиаданных, полученных во время регистрации клиентов.

Компонент сохраняет следующие бинарные данные:

- фонограммы, расшифровки и другие данные, относящихся к вызовам;
- произвольные метаданные в привязке к хранимым данным.

Компонент позволяет:

- получать, искать и извлекать данные по внутреннему API платформы;
- искать и извлекать данные по HTTP для внешних систем;
- хранить исходные данные.

3.2.10 Reporter Component

Reporter Component (компонент формирования отчётности) отвечает за формирование отчётов по данным, собираемым в журналы регистрации событий в ходе функционирования программного комплекса.

Принципы работы

- исходными данными для отчётов служат данные, собираемые компонентом журналирования (**Chronicler Component**);
- отчёты строятся на основе шаблонов, подготовленных и прошедших проверку на отсутствие ошибок;
- шаблон представляет собой описание структуры отчёта и SQL-запрос, формирующий отчёт;
- отчёт представляет собой таблицу в базе данных, сформированную в соответствии с заданным шаблоном;
- отчёт может быть экспортирован в файл формата CSV, JSON, XLS, XLSX;
- отчёты можно формировать по заданному расписанию.

У **Reporter Component** есть интерфейс разработчика отчётов для создания, настройки шаблонов и расписания, формирования отчётов.

3.2.11 Router Component

Router Component (компонент маршрутизации) реализует сервис маршрутизации сообщений.

Router Component обеспечивает:

- централизованное хранение информации об обработчиках сообщений, реализованных в программных компонентах комплекса;

– формирование технологических маршрутов обработки сообщений в ходе системного взаимодействия компонентов.

3.2.12 Security Component

Security Component (подсистема разграничения доступа) реализует централизованное управление разграничением прав доступа пользователей к ресурсам системы.

Security Component обеспечивает:

- формирование структуры разделяемых ресурсов системы и назначаемых на доступ к ресурсам привилегий;
- определение привилегий на выполнение заданных действий над ресурсами в разрезе зарегистрированных приложений;
- реализация ролевой модели доступа к ресурсам;
- авторизацию клиента на выполнение запрошенных действий над ресурсами;
- предотвращение попыток несанкционированного доступа.

Для разграничения прав доступа пользователей к ресурсам системы может использоваться:

- веб-интерфейс подсистемы разграничения доступа;
- механизм интеграции с **Active Directory**. В этом случае пользователи **Active Directory** конвертируются в пользователей системы. Роли, привилегии и группы системы соотносятся с определёнными группами **Active Directory**, позволяя управлять правами пользователей платформы из **Active Directory**.

3.3 Сервисы VoiceKey.IVR

Enter topic text here.

3.3.1 Siphon Application

Siphon Application — менеджер телефонного взаимодействия:

- реализует протокол SIP;
- обеспечивает соединение с АТС;
- управляет взаимодействием агентов в ходе сессии.

3.3.2 MRCP Application

MRCP Application (сервис взаимодействия по MRCP):

- реализует протокол MRCP;
- обеспечивает интеграцию с голосовой платформой клиента и обработку поступающих от голосовой платформы запросов;
- взаимодействует с обработчиками распознавания и синтеза речи.

3.3.3 Dialog Platform

Dialog Platform — пользовательский интерфейс для подсистемы телефонного взаимодействия по протоколу SIP **VoiceKey.IVR SIP**.

3.3.4 VoiceRecognizer Processor

VoiceRecognizer Processor — обработчик запросов на распознавание речи.

Реализует преобразование речи, используя технологию распознавания речи.

3.3.5 TextTransform Processor

TextTransform Processor — обработчик для преобразования распознанного текста в другой вид. Результат будет зависеть от выбранной модели свёртки.

3.3.6 TTS Processor

TTS Processor — обработчик запросов на синтез речи.

Реализует преобразование текста в речь, используя технологию синтеза речи.

3.3.7 Vocabulary Application

Vocabulary Application — веб-сервис для добавления словарей и отдельных слов для распознавания.

3.3.8 Buzzer Application

Buzzer Application — веб-сервис управления кампаниями:

- реализует необходимую для формирования и управления кампаниями функциональность;
- обеспечивает взаимодействие с **Siphon Application** (менеджером телефонного взаимодействия) в ходе выполнения кампании;
- осуществляет контроль хода выполнения кампании и статусов оповещений;
- предоставляет REST-интерфейс для доступа клиентских приложений к методам формирования и управления кампаниями.

3.3.9 TTSCloud Application

Веб-сервис синтеза речи (**TTSCloud Application**):

- предоставляет REST-интерфейс для доступа клиентских приложений к функциональности **VoiceKey.IVR**, необходимой для синтеза речи;
- предоставляет медиаконцентратору (**MediaHub Component**) настройки для подключения к источнику и потребителю медиаданных и описание задачи, последовательность операций, которые требуется выполнить в рамках текущей сессии;
- в ходе выполнения запросов, поступающих от клиентского приложения, осуществляет операционное взаимодействие с обработчиком запросов синтеза речи (**TTS Processor**).

3.3.10 ASRCloud Application

Веб-сервис распознавания речи (ASRCloud Application):

- предоставляет REST-интерфейс для доступа клиентских приложений к функциональности **VoiceKey.IVR**, необходимой для распознавания речи;
- предоставляет медиаконцентратору (**MediaHub Component**) настройки для подключения к источнику и потребителю медиаданных и описание задачи, последовательность операций, которые требуется выполнить в рамках текущей сессии;
- в ходе выполнения запросов, поступающих от клиентского приложения, осуществляет операционное взаимодействие с обработчиком запросов распознавания речи (VoiceRecognizer Processor).

3.3.11 Ripper Component

Ripper	Component	(модуль	обработки	медиаданных)	обеспечивает
перекодирование/передискретизацию звука.					

4. РЕШЕНИЯ НА БАЗЕ VOICEKEY.IVR

Реализация комплекса технологий позволяет с успехом использовать **VoiceKey.IVR** во всех каналах дистанционного обслуживания:

- контактный центр;
- мобильное приложение;
- IVR или виртуальный ассистент системы самообслуживания;
- веб-приложение.

В настоящее время на базе комплекса **VoiceKey.IVR** построены следующие типовые решения:

Решение	Описание
VoiceKey.IVR ASR	Решение для встраивания возможностей распознавания речи в программное обеспечение, онлайн-сервисы и мобильные приложения. Преобразует речь в текст с использованием технологии распознавания речи.
VoiceKey.IVR ASR gRPC	Решение предназначено для встраивания возможностей распознавания речи в программное обеспечение, онлайн-сервисы, мобильные приложения и другие продукты. Взаимодействие с VoiceRecognizer Processor и TextTransform Processor осуществляется по протоколу gRPC. Решение позволяет преобразовать речь на русском языке в текст с использованием технологии распознавания речи, а затем привести текстовый результат распознавания к форме, заданной пользователем, если это требуется.
VoiceKey.IVR SIP	Решение для контакт-центров, которое позволяет использовать технологии синтеза и распознавания речи при работе в голосовом меню. Решение предназначено для управления кампаниями на входящее/исходящее оповещение. VoiceKey.IVR SIP также предоставляет информацию о ходе кампаний по исходящему оповещению и информацию по бизнес-статусам и ответам клиентов, полученным во время диалога. Кроме того, VoiceKey.IVR SIP позволяет вести справочник контактов.
VoiceKey.IVR MRCP	Решение для контакт-центров, которое позволяет использовать технологии синтеза и распознавания речи при работе в голосовом меню. Решение предоставляет сервисы синтеза и распознавания по протоколу MRCPv1 (RFC 4463) и MRCPv2 (RFC 6787).
VoiceKey.IVR TTS	Решение для встраивания возможностей синтеза речи в программное обеспечение, онлайн-сервисы и мобильные приложения. Позволяет озвучить текст с использованием технологии синтеза речи.

Решение	Описание
VoiceKey.IVR TTS gRPC	Решение предназначено для встраивания возможностей синтеза речи в программное обеспечение, онлайн-сервисы, мобильные приложения и другие продукты. Взаимодействие с TTS Processor осуществляется по протоколу gRPC. Решение позволяет озвучить текст на одном из нескольких языков с использованием технологии синтеза речи. Для каждого языка представлен как минимум один голос.

4.1 VoiceKey.IVR SIP

На рисунке приведена типовая схема встраивания возможностей управления оповещениями и входящими звонками во внешнюю систему.

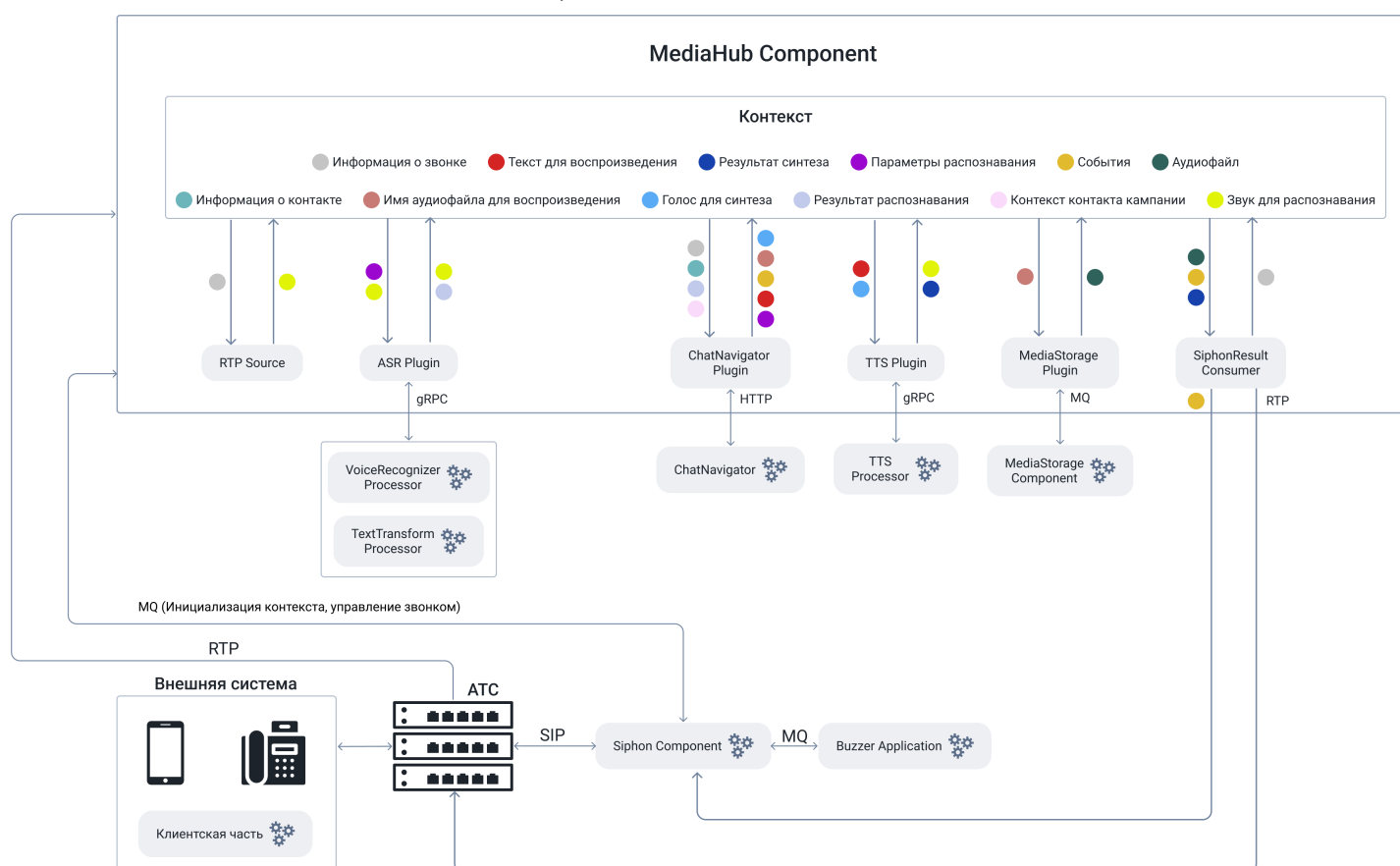


Рисунок 1 — Схема интеграции решения с внешней системой

Основной принцип работы

VoiceKey.IVR SIP обеспечивает обработку входящих запросов, поступающих по протоколу SIP с помощью компонента **Siphon Application** и обеспечивает исходящие оповещения по сформированным кампаниям с помощью компонента **Buzzer Application**.

Распознавание речи

Во время телефонного разговора происходит распознавание речи клиента с использованием **VoiceRecognizer Processor**. Финальный результат трансформируется в зависимости от используемой модели распознавания с помощью **TextTransform Processor**. Сценарий ответа и исходящих оповещений формируется в системе **ChatNavigator**. Ответ и оповещение синтезируется с помощью **TTS Processor** и/или **MediaStorage Component**.

Общие положения

Взаимодействие **Siphon Application** с АТС происходит по протоколу SIP v2.0 ([RFC 3261](#)), а **MediaHub Component** с АТС — по протоколу RTP.

Составляющие **MediaHub Component**, отображённые на схеме, входят в состав компонента и устанавливаются совместно с ним.

Запросы к **VoiceKey.IVR SIP** выполняются через программный интерфейс на базе архитектуры REST (далее — REST API). Взаимодействие осуществляется при помощи отправки HTTP-запросов веб-сервису **VoiceKey.IVR SIP**.

Стандартом для представления входных и выходных данных является JSON.

4.1.1 Компоненты, необходимые для решения

Для корректной работы **VoiceKey.IVR SIP** необходимы следующие компоненты:

Компонент	Комментарий
SessionService Application	
Buzzer Application	
Siphon Application	
Configurator Component	
Router Component	
Security Component	
stc-net-license	
Monitoring Component	
MediaStorage Component	
LoadBalancer Component	
MediaHub Component	возможна совместная установка нескольких экземпляров компонентов для масштабирования нагрузки
TTS Processor	возможна совместная установка нескольких экземпляров компонентов для масштабирования нагрузки

TextTransform Processor	возможна совместная установка нескольких экземпляров компонентов для масштабирования нагрузки при этом все экземпляры должны иметь одинаковый набор моделей
VoiceRecognizer Processor	возможна совместная установка нескольких экземпляров компонентов для масштабирования нагрузки при этом все экземпляры должны иметь одинаковый набор моделей
VoiceKey Portal	
Dialog Platform	
Chronicler Component	устанавливается при необходимости
Reporter Component	устанавливается при необходимости

4.2 VoiceKey.IVR MRCP

На рисунках ниже приведены типовые схемы встраивания возможностей голосового взаимодействия во внешнюю систему.

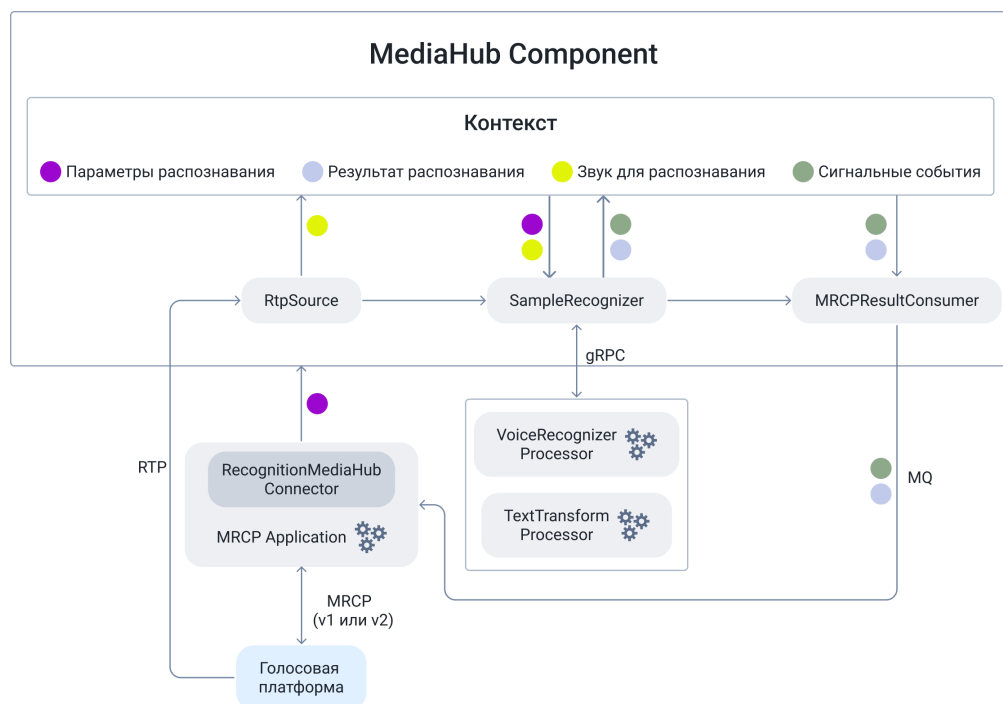


Рисунок 2 — Схема интеграции сервиса распознавания с внешней системой

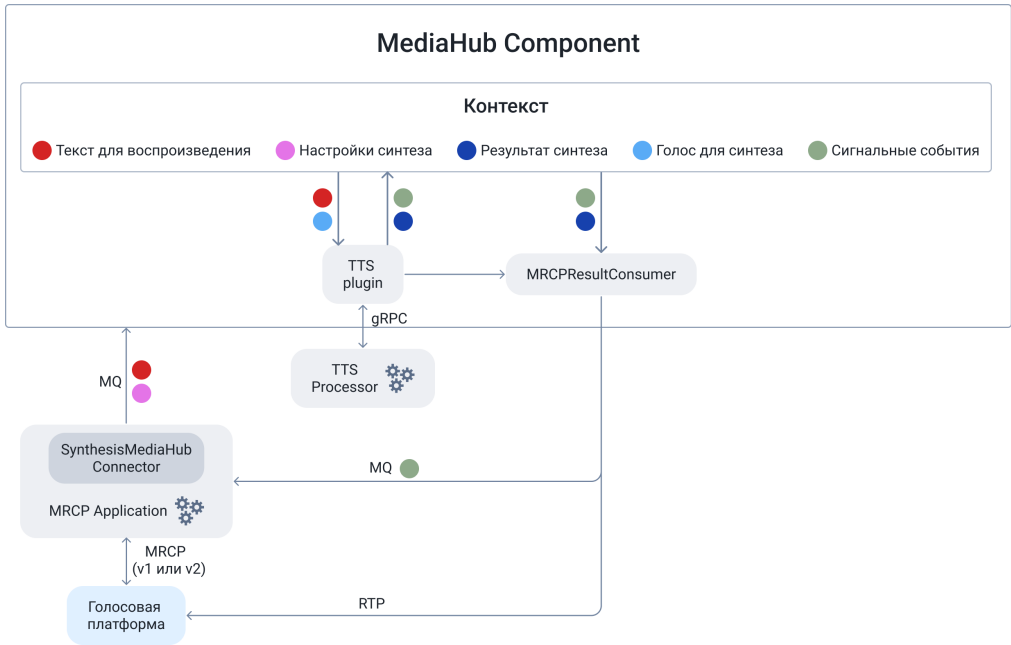


Рисунок 3 — Схема интеграции сервиса синтеза с внешней системой

Голосовая платформа инициирует запрос на распознавание и/или синтез речи по протоколу MRCP. **MRCP Application** обрабатывает запрос и передаёт его компоненту **VoiceRecognizer Processor** и **TTS Processor**. Обработанные данные возвращаются голосовой платформе.

Информационный обмен между компонентами комплекса и **MRCP Application** осуществляется через шину данных, построенную на базе связующего программного обеспечения **RabbitMQ**, а также через **WebSocket**, через который также осуществляется синтез.

4.2.1 Компоненты, необходимые для решения

Для корректной работы **VoiceKey.IVR MRCP** необходимы следующие компоненты:

Компонент	Комментарий
MRCP Application	
Configurator Component	
Router Component	
Security Component	
stc-net-license	
Monitoring Component	
MediaHub Component	возможна совместная установка нескольких экземпляров компонентов для масштабирования нагрузки
TextTransform Processor	возможна совместная установка нескольких экземпляров компонентов для масштабирования нагрузки

	при этом все экземпляры должны иметь одинаковый набор моделей
TTS Processor	возможна совместная установка нескольких экземпляров компонентов для масштабирования нагрузки
VoiceRecognizer Processor	возможна совместная установка нескольких экземпляров компонентов для масштабирования нагрузки при этом все экземпляры должны иметь одинаковый набор моделей
LoadBalancer Component	обязателен при использовании двух и более VoiceRecognizer Processor или TTS Processor
MediaStorage Component	устанавливается при необходимости
Chronicler Component	устанавливается при необходимости
Reporter Component	устанавливается при необходимости
SessionService Application	устанавливается при необходимости
VoiceKey Portal	устанавливается при необходимости

4.3 VoiceKey.IVR ASR

На рисунке приведена типовая схема встраивания возможностей распознавания речи во внешнюю систему.

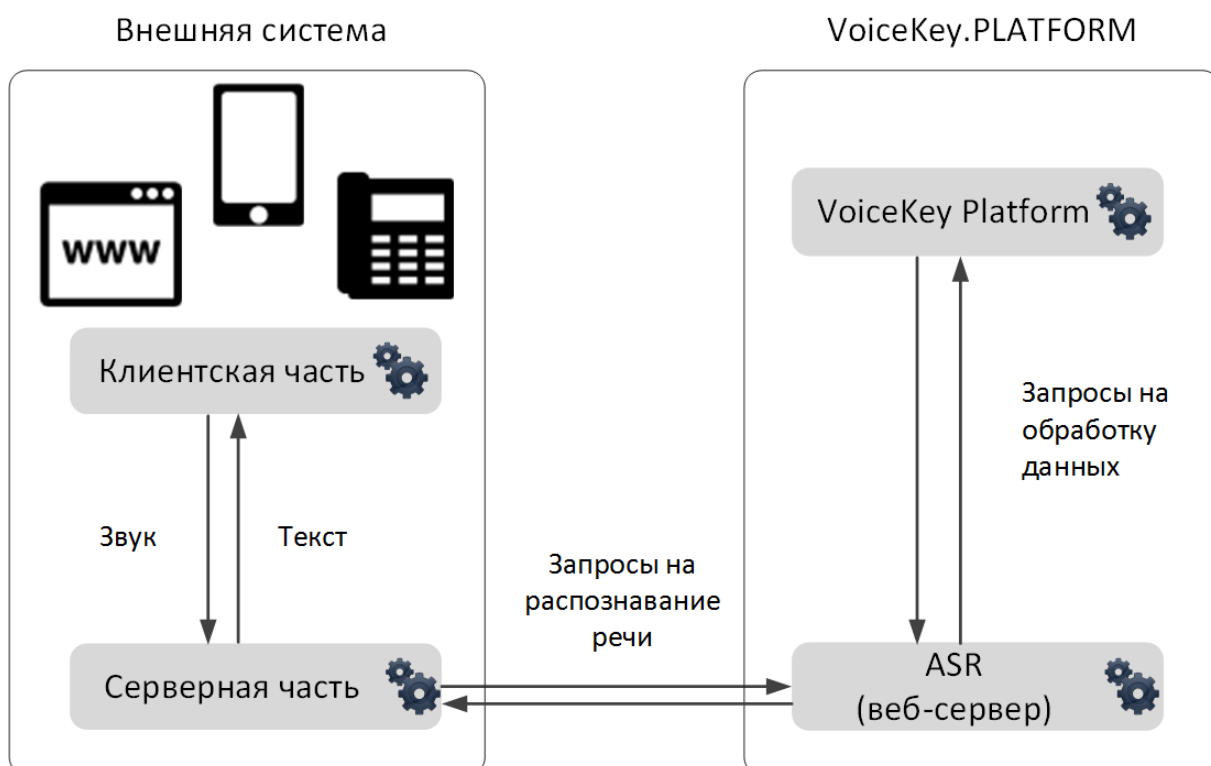


Рисунок 4 — Схема интеграции с внешней системой

В качестве клиентской части может выступать мобильное приложение, сайт или другой сервис, для которого требуется преобразование речи в текстовые данные. Клиентская часть отправляет данные серверной части внешней системы. Серверная часть, в свою очередь, адресует запросы к **VoiceKey.ASR**, используя специальный программный интерфейс на базе архитектуры REST (далее — API). Взаимодействие осуществляется при помощи отправки HTTP-запросов веб-сервису **ASR Application**. Стандартом для представления входных и выходных данных является JSON.

Информационный обмен между компонентами комплекса и **ASR Application** осуществляется через шину данных, построенную на базе связующего программного обеспечения **RabbitMQ**.

4.3.1 Компоненты, необходимые для решения

Для корректной работы **VoiceKey.ASR** необходимы следующие компоненты:

Компонент	Комментарий
SessionService Application	
ASRCloud Application	
Vocabulary Application	
Configurator Component	
Router Component	
Security Component	
stc-net-license	
Monitoring Component	
Ripper Component	
MediaHub Component	возможна совместная установка нескольких экземпляров компонентов для масштабирования нагрузки
TextTransform Processor	возможна совместная установка нескольких экземпляров компонентов для масштабирования нагрузки при этом все экземпляры должны иметь одинаковый набор моделей
VoiceRecognizer Processor	возможна совместная установка нескольких экземпляров компонентов для масштабирования нагрузки при этом все экземпляры должны иметь одинаковый набор моделей

MediaStorage Component	Обязателен при включении синхронизации кэша графов с MediaStorage Component . Синхронизация включается в параметре local_mode в секции asr_defaults .
LoadBalancer Component	обязателен при использовании двух и более TextTransform Processor или VoiceRecognizer Processor
Chronicler Component	устанавливается при необходимости
Reporter Component	устанавливается при необходимости
VoiceKey Portal	устанавливается при необходимости

4.4 VoiceKey.IVR TTS

Передача по протоколу WebSocket

На рисунке ниже приведена схема взаимодействия компонентов при работе в режиме передачи звука по протоколу WebSocket.

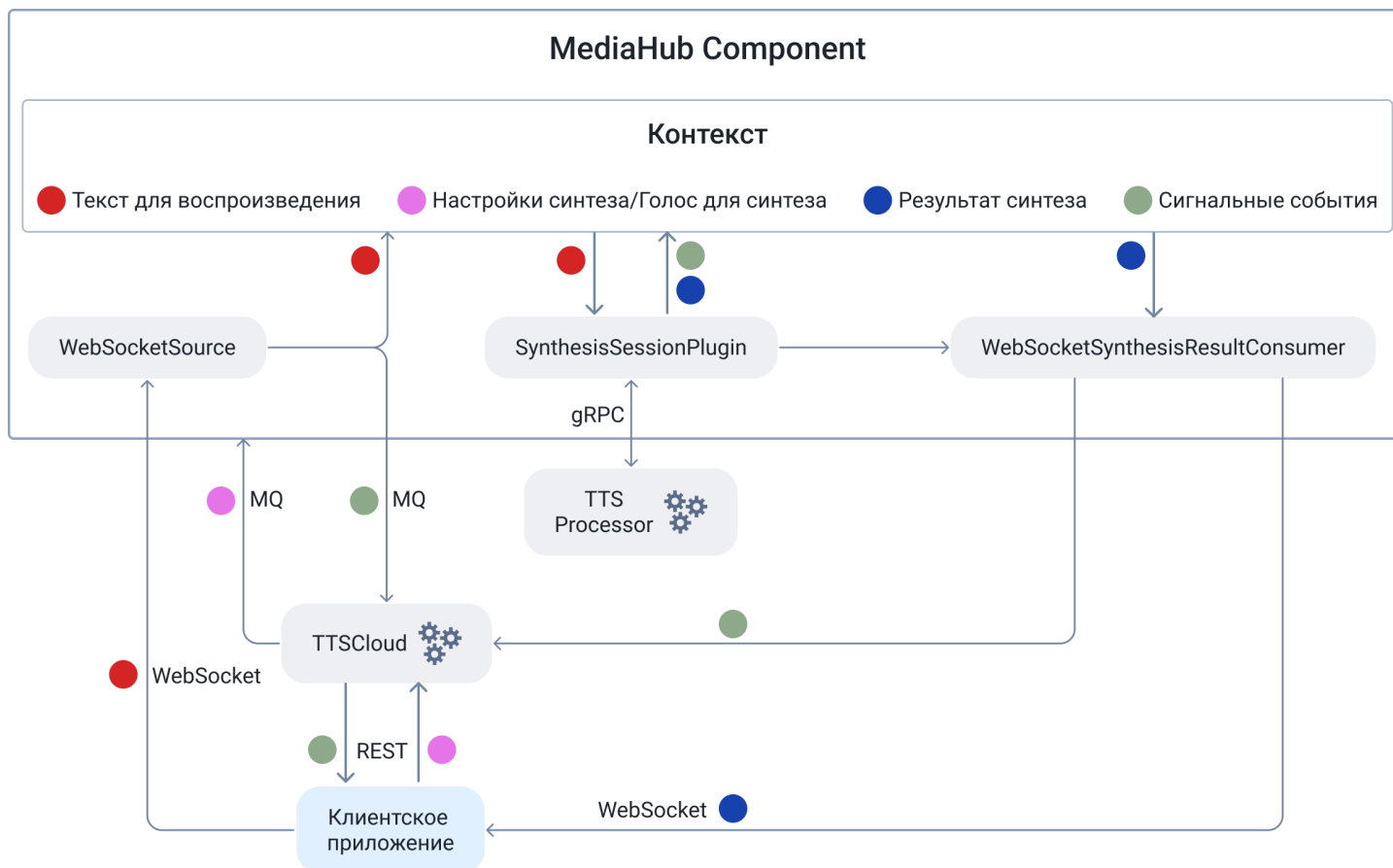


Рисунок 5 — Схема взаимодействия при работе через WebSocket

Клиент отправляет запрос в **TTSCloud Application** для открытия транзакции. Происходит инициализация внутренних компонентов **MediaHub Component**. Клиенту возвращается ссылка для подключения по протоколу WebSocket.

VoiceKey.IVR TTS

Текст от клиента передаётся по WebSocket в **MediaHub Component**, а затем в **TTS Processor**, где происходит обработка. По WebSocket клиенту возвращается звук в байтах по мере готовности.

Клиент отправляет запрос в **TTSCloud Application** для закрытия транзакции.

Для улучшения пропускной способности может быть несколько **TTS Processor**.

Передача по протоколу HTTP

На рисунке ниже приведена схема взаимодействия компонентов при работе в режиме передачи звука по протоколу HTTP.

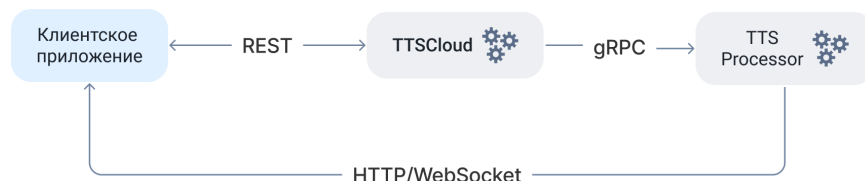


Рисунок 6 — Схема взаимодействия при работе через REST API

По REST API передаются все данные в **TTSCloud Application**. **TTSCloud Application** отправляет в **TTS Processor** конфигурацию и текст на синтез. Синтезированная речь целиком возвращается клиенту по HTTP.

При использовании буферной транзакции взаимодействуют те же компоненты, изменяется только алгоритм действий. Клиентское приложение отправляет настройки (порядок байтов и голос) без текста для синтеза по REST API в **TTSCloud Application**. **TTSCloud Application** сохраняет настройки и возвращает клиенту уникальный идентификатор транзакции. В следующем запросе клиент передаёт идентификатор транзакции, идентификатор сессии и текст по REST API в **TTSCloud Application**. **TTS Processor** синтезирует звук полностью и возвращает его клиенту по WebSocket. Клиент закрывает транзакцию.

Для улучшения пропускной способности может быть несколько **TTS Processor**.

4.4.1 Компоненты, необходимые для решения

Для корректной работы **VoiceKey.TTS** необходимы следующие компоненты:

Компонент	Комментарий
SessionService Application	
TTSCloud Application	
Configurator Component	
Router Component	
Security Component	
stc-net-license	
Monitoring Component	
MediaHub Component	возможна совместная установка нескольких экземпляров компонентов для масштабирования нагрузки

VoiceKey.IVR TTS

TTS Processor	возможна совместная установка нескольких экземпляров компонентов для масштабирования нагрузки
VoiceKey Portal	устанавливается при необходимости
Chronicler Component	устанавливается при необходимости
Reporter Component	устанавливается при необходимости

4.5 VoiceKey.IVR ASR gRPC

VoiceRecognizer Processor реализует преобразование речи в текст. Если распознанный текст (даты, телефонные номера и т.д.) необходимо преобразовать к нужному формату, тогда используется **TextTransform Processor**. Взаимодействие с **VoiceRecognizer Processor** и **TextTransform Processor** осуществляется по протоколу gRPC.

Компонентов **VoiceRecognizer Processor** и **TextTransform Processor** может быть установлено несколько. Для балансировки нагрузки используется **LoadBalancer Component** или Nginx. ASR gRPC

4.5.1 Компоненты, необходимые для решения

Для корректной работы **ASR gRPC** необходимы следующие компоненты:

Компонент	Комментарий
Configurator Component	
Router Component	
Security Component	
stc-net-license	
Monitoring Component	
TextTransform Processor	возможна совместная установка нескольких экземпляров компонентов для масштабирования нагрузки при этом все экземпляры должны иметь одинаковый набор моделей
VoiceRecognizer Processor	возможна совместная установка нескольких экземпляров компонентов для масштабирования нагрузки при этом все экземпляры должны иметь одинаковый набор моделей
Chronicler Component	устанавливается при необходимости
Reporter Component	устанавливается при необходимости
VoiceKey Portal	устанавливается при необходимости

4.6 VoiceKey.IVR TTS gRPC

TTS gRPC реализует синтез речи. Взаимодействие с **TTS Processor** осуществляется по протоколу gRPC. Компонентов **TTS Processor** может быть установлено несколько. Для балансировки нагрузки используется **LoadBalancer Component** или **Nginx**.

4.6.1 Компоненты, необходимые для решения

Для корректной работы **TTS gRPC** необходимы следующие компоненты:

Компонент	Комментарий
Configurator Component	
Router Component	
Security Component	
stc-net-license	
Monitoring Component	
TTS Processor	возможна совместная установка нескольких экземпляров компонентов для масштабирования нагрузки
Chronicler Component	устанавливается при необходимости
Reporter Component	устанавливается при необходимости
VoiceKey Portal	устанавливается при необходимости