基于freeswitch构建企业级呼叫系统

一、构建freeswitch 的docker容器

1、启动cenos7容器服务

```
docker run -it -d  --name  freeswitch_1.10.8  centos:7
```

2、去freeswitch的官网申请token

```
官网地址：https://voice9.signalwire.com/dashboard

#将用户名和token写入文件
echo "voice9" > /etc/yum/vars/signalwireusername
echo "PT699d3a10bf366b0294d4b8e318ff5885df92dae5beed5dcd" > /etc/yum/vars/signalwiretoken
```

3、更新centos7的源

```
yum install -y  wget

wget -O /etc/yum.repos.d/CentOS-Base.repo http://mirrors.aliyun.com/repo/Centos-7.repo
yum clean all
yum makecache
yum -y update

yum install -y subversion autoconf automake libtool gcc-c++ ncurses-devel make
yum install -y http://files.freeswitch.org/repo/yum/centos-release/freeswitch-release-repo-0-1.noarch.rpm epel-release
yum install -y yum-plugin-ovl centos-release-scl rpmdevtools yum-utils git
yum install -y alsa-lib-devel autoconf automake bison broadvoice-devel bzip2 centos-release-scl cmake3 curl-devel devtoolset-7 devtoolset-7
```

4、安装noarch

```
cd /usr/local/src/
wget http://files.freeswitch.org/freeswitch-release-1-6.noarch.rpm
yum install -y freeswitch-release-1-6.noarch.rpm


yum install -y libatomic
yum install -y git alsa-lib-devel autoconf automake bison broadvoice-devel bzip2 curl-devel libdb4-devel e2fsprogs-devel erlang flite-devel
```

5、安装cmake

```
yum remove cmake
wget https://cmake.org/files/v3.14/cmake-3.14.0.tar.gz
tar -zxvf cmake-3.14.0.tar.gz
cd cmake-3.14.0
./configure
make && make install
```

6、安装libks

```
cd /usr/local/src/
git clone https://github.com/signalwire/libks.git
cd libks
cmake .
make && make install
```

7、安装 signalwire-c

```
cd /usr/local/src/
git clone https://github.com/signalwire/signalwire-c.git
cd signalwire-c/
cmake .
make && make install
ln -sf /usr/local/lib64/pkgconfig/signalwire_client.pc /usr/lib64/pkgconfig/signalwire_client.pc
```

8、安装x264

```
cd /usr/local/src/
git clone http://git.videolan.org/git/x264.git
cd x264
./configure  --disable-asm
make && make install
```

9、安装mod_av

```
wget -c http://files.freeswitch.org/downloads/libs/libx264.tar.bz2
tar -jxvf libx264.tar.bz2
cd libx264
./configure --enable-static --enable-shared --prefix=/usr
make && make install
cp /usr/lib/pkgconfig/x264.pc /usr/lib64/pkgconfig/
cp /usr/lib/libx264.so /usr/lib64/
cp /usr/lib/libx264.a /usr/lib64/

# download and install libav
wget -c http://files.freeswitch.org/downloads/libs/libav-12.tar.bz2
tar -jxvf libav-12.tar.bz2
cd libav
./configure --enable-pic --enable-shared  --enable-libx264 --enable-gpl --extra-libs="-ldl" --extra-cflags=-I/usr/include --extra-ldflags=-
make && make install    # make CXXFLAGS="-fPIC"

cp /usr/local/lib/pkgconfig/libavcodec.pc /usr/local/lib/pkgconfig/libavdevice.pc /usr/local/lib/pkgconfig/libavfilter.pc /usr/local/lib/pk
# 执行刷新，以让FreeSWITCH运行时可以找到库
ldconfig
```

10、安装libpng

```
git clone https://freeswitch.org/stash/scm/sd/libpng.git
 cd libpng
./configure
make && make install
cp /usr/local/lib/pkgconfig/libpng* /usr/lib64/pkgconfig/
```

10、安装opus

```
git clone https://freeswitch.org/stash/scm/sd/opus.git
cd opus
./autogen.sh
./configure --libdir=$PWD/tmp
make && make install
```

11、安装sofia-sip

```
git clone https://github.com/freeswitch/sofia-sip
cd sofia-sip
./bootstrap.sh
./configure
make && make install
```

12、安装spandsp

```
git clone https://github.com/freeswitch/spandsp
cd spandsp
./bootstrap.sh
./configure
make && make install
export PKG_CONFIG_PATH=/usr/local/lib/pkgconfig
```

13、安装libopus-devel rmp包

```
vim /etc/yum.repos.d/linuxtech.repo
[linuxtech]
name=LinuxTECH
baseurl=http://pkgrepo.linuxtech.net/el6/release/
enabled=1
gpgcheck=1
gpgkey=http://pkgrepo.linuxtech.net/el6/release/RPM-GPG-KEY-LinuxTECH.NET

# 创建仓库,重新安装
 yum install -y libopus-devel
```

14、安装freeswitch

```
cd /usr/local/src/
wget http://files.freeswitch.org/freeswitch-1.10.8.-release.tar.gz
tar vzxf freeswitch-1.10.8.-release.tar.gz
cd freeswitch-1.10.8.-release
./configure --prefix=/app/freeswitch
make && make install
ln -sf /app/freeswitch/bin/freeswitch /usr/bin/
ln -sf /app/freeswitch/bin/fs_cli /usr/bin/
```

在load mod_av时，如果出现 *libavformat.so.57 file not found

```
echo "/usr/local/lib" >> /etc/ld.so.conf
ldconfig
```

在进入freeswitch控制台

```
#在宿主机上进入fs
docker exec -it freeswitch_x86  /usr/local/freeswitch/bin/fs_cli -H 172.17.0.2 -P 7400 -p voice9.com

#在容器里面进入fs
fs_cli -H 127.0.0.1 -P 7400 -p voice9.com

#启动、停止服务
freeswitch -nc -rp
freeswitch -stop
```

启动docker容器服务

```
docker run -it -d --name freeswitch_x86 \
-v /usr/local/freeswitch/conf:/usr/local/freeswitch/conf \
-v /usr/local/freeswitch/log:/usr/local/freeswitch/log \
-v /app/freeswitch/record:/app/freeswitch/record \
-v /usr/local/freeswitch/storage:/usr/local/freeswitch/storage \
-v /usr/local/freeswitch/sounds:/usr/local/freeswitch/sounds \
--network=host  registry.cn-hangzhou.aliyuncs.com/voice9_x86/freeswitch:1.1.0
```

二、基于Java程序打出去第一个电话

1、使用java esl 程序建立socket连接

```
public class FsListen {

      public void start() {
              for (int i = 0; i < threadNum; i++) {
                  ThreadFactory threadFactory = new ThreadFactoryBuilder().setNameFormat("fs-pool-" + i).build();
                  ThreadPoolExecutor executor = new ThreadPoolExecutor(1, 1, 0L, TimeUnit.MILLISECONDS, new LinkedBlockingQueue<Runnable>()
                  executorMap.put(i, executor);
              }

              Map<String, Object> params = new HashMap<>();
              params.put("applicationType", 4);
              params.put("applicationGroup", group);
              List<Station> fsStations = stationMapper.selectListByMap(params);
              for (Station station : fsStations) {
                  if (station.getStatus() == 1) {
                      connect(station.getApplicationHost(), station.getApplicationPort(), station.getPwd());
                  }
              }
              //重连检测
              checkFsThread.scheduleAtFixedRate(() -> {
                  try {
                      checkConnect();
                  } catch (Exception e) {
                      logger.error(e.getMessage(), e);
                  }
              }, 2, 1, TimeUnit.MINUTES);
          }

  private void connect(String host, Integer port, String password) {
        Client client = new Client();
        try {
            SocketAddress socketAddress = new InetSocketAddress(host, port);
            logger.info("Connecting to {} passwd:{}", socketAddress, password);
            client.connect(socketAddress, password, 3);
            if (localAddress == null) {
                InetSocketAddress address = (InetSocketAddress) client.getChannel().localAddress();
                localAddress = Constant.HTTP + address.getAddress().getHostAddress() + Constant.CO + localPort + Constant.SK + Constant.FS_
                //cacheService.setHost(localAddress);
            }
        } catch (Throwable e) {
            logger.error(e.getMessage(), e);
            return;
```

```
        }
        fsClient.put(host + ":" + port, client);
        client.setEventSubscriptions(IModEslApi.EventFormat.PLAIN, "all");
        IEslEventListener listener = new IEslEventListener() {....}

     }

}
```

2、坐席SDK外呼，桥接客户端channel

```
1、websocket
2、long polling

先呼坐席侧，再呼用户侧，最后桥接两个channel,这里是通过deviceId来记录的；
```

3、坐席和客户处于通话中，咨询另外一个坐席

```
通过前面已经拿到的一个callId,2个deviceId，现在要发起第三个人呼叫
坐席和第一个客户需要先从桥接状态拆线， 客户侧处于hold状态

结束咨询需要先把客户的hold状态break，再和坐席桥接。
```

4、坐席和客户通话中，转接给另外一个坐席

```
也是先发起呼叫，设备接起之后，才把坐席自己拆线，桥接剩下的2个设备
这里客户是没有放hold音的。
```

5、多方会议

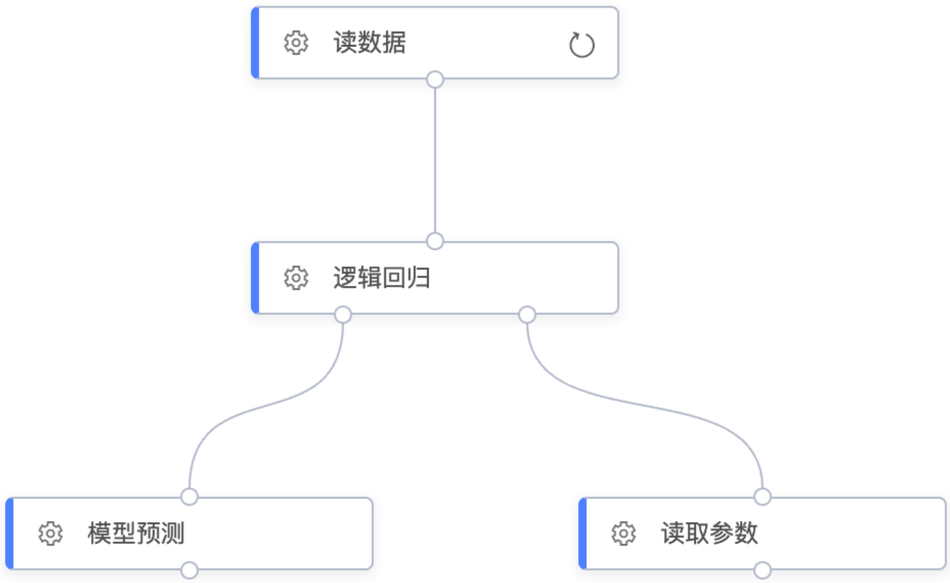
```
1、先把第三个人呼起来，这里一般用的是咨询，再拉入会议，
2、加入第4个人，咨询接听之后直接入会。
```

三、基于Janus构建webrtc音视频电话

```
# janus 注册sip  https://janus.conf.meetecho.com/siptest.html
sip:110.40.188.252:6685
sip:950106@110.40.188.252:6685
950106
vGtlXzG1

sip:18612983191@110.40.188.252:6685
```

四、基于scxml实现通话流程控制



构建一个最简单的scxml

```
<scxml xmlns="http://www.w3.org/2005/07/scxml" xmlns:v9="https://www.voice9.com" version="1.0" datamodel="jexl"
      initialstate="1">
   <datamodel>
       <data id="1" expr="开始_1"/>
       <data id="2" expr="结束_2"/>
       <data id="4" expr="5级评价收号_4"/>
       <data id="51" expr="满意_51"/>
       <data id="-5" expr="不满意_51"/>
       <data id="3" expr="评价结束语_3"/>
       <data id="-7" expr="一般_51"/>
       <data id="-8" expr="非常满意_51"/>
       <data id="-9" expr="非常不满意_51"/>
   </datamodel>
   <state id="1">
       <onentry>
           <v9:init/>
       </onentry>
       <transition event="evt.end" target="end"/>
       <transition event="evt.next" target="4"/>
       <transition event="evt.next" cond="_event.data=='null'"/>
       <transition event="evt.next" cond="_event.data=='end'" target="end"/>
       <transition event="evt.next" cond="_event.data=='4'" target="4"/>
       <transition event="evt.next" cond="_event.data=='51'" target="51"/>
       <transition event="evt.next" cond="_event.data=='-5'" target="-5"/>
       <transition event="evt.next" cond="_event.data=='3'" target="3"/>
       <transition event="evt.next" cond="_event.data=='-7'" target="-7"/>
       <transition event="evt.next" cond="_event.data=='-8'" target="-8"/>
       <transition event="evt.next" cond="_event.data=='-9'" target="-9"/>
   </state>
   <final id="end">
       <onentry>
           <v9:end hangup="1"/>
       </onentry>
   </final>
   <state id="4" initial="4_0">
       <state id="4_0">
           <onentry>
               <v9:menu stateId="4_0" playType="1" dtmfInterrupt="1" dtmfArray="1,2,3,4,5"  loopTimes="3" singleLoop="1"
                                 intervalTime="2" digitsTime="2" min="1" max="1" terminator=""
```

```
                                ossId="1005.wav"/>
            </onentry>
            <transition event="evt.end" target="end"/>
            <transition event="evt.next" cond="_event.data=='1'" target="-8"/>
            <transition event="evt.next" cond="_event.data=='2'" target="51"/>
            <transition event="evt.next" cond="_event.data=='3'" target="-7"/>
            <transition event="evt.next" cond="_event.data=='4'" target="-5"/>
            <transition event="evt.next" cond="_event.data=='5'" target="-9"/>
            <transition event="evt.dtmf_error_key" cond="_event.data&gt;'0'" target="4"/>
            <transition event="evt.dtmf_error_key" cond="_event.data=='0'" target="end"/>
            <transition event="evt.dtmf_timeout" cond="_event.data&gt;'0'" target="4"/>
            <transition event="evt.dtmf_timeout" cond="_event.data=='0'" target="end"/>
            <transition event="evt.dtmf_error" target="end"/>
        </state>
    </state>
    <state id="51" initial="51_0">
        <state id="51_0">
            <onentry>
                <v9:assign stateId="51" param="eval_key" value="2" type="1"/>
            </onentry>
            <transition event="evt.end" target="end"/>
            <transition event="evt.next" target="51_1"/>
        </state>
        <state id="51_1">
            <onentry>
                <v9:assign param="eval_key_val" value="满意" type="1"/>
            </onentry>
            <transition event="evt.end" target="end"/>
            <transition event="evt.next" target="3"/>
        </state>
    </state>
    <state id="-5" initial="-5_0">
        <state id="-5_0">
            <onentry>
                <v9:assign stateId="-5" param="eval_key" value="4" type="1"/>
            </onentry>
            <transition event="evt.end" target="end"/>
            <transition event="evt.next" target="-5_1"/>
        </state>
        <state id="-5_1">
            <onentry>
                <v9:assign param="eval_key_val" value="不满意" type="1"/>
            </onentry>
            <transition event="evt.end" target="end"/>
            <transition event="evt.next" target="3"/>
        </state>
    </state>
    <state id="3" initial="3_0">
        <state id="3_0">
            <onentry>
                <v9:play stateId="3_0" playType="1" dtmfInterrupt="0" ossId="thanks_goodbye.wav"/>
            </onentry>
            <transition event="evt.end" target="end"/>
            <transition event="evt.next" target="end"/>
            <transition event="evt.playend_break" target="end"/>
            <transition event="evt.playend_error" target="end"/>
        </state>
    </state>
    <state id="-7" initial="-7_0">
        <state id="-7_0">
            <onentry>
                <v9:assign stateId="-7" param="eval_key" value="3" type="1"/>
            </onentry>
            <transition event="evt.end" target="end"/>
            <transition event="evt.next" target="-7_1"/>
        </state>
        <state id="-7_1">
            <onentry>
                <v9:assign param="eval_key_val" value="一般" type="1"/>
            </onentry>
            <transition event="evt.end" target="end"/>
            <transition event="evt.next" target="3"/>
        </state>
    </state>
    <state id="-8" initial="-8_0">
        <state id="-8_0">
            <onentry>
                <v9:assign stateId="-8" param="eval_key" value="1" type="1"/>
            </onentry>
            <transition event="evt.end" target="end"/>
            <transition event="evt.next" target="-8_1"/>
```

```
        </state>
        <state id="-8_1">
            <onentry>
                <v9:assign param="eval_key_val" value="非常满意" type="1"/>
            </onentry>
            <transition event="evt.end" target="end"/>
            <transition event="evt.next" target="3"/>
        </state>
    </state>
    <state id="-9" initial="-9_0">
        <state id="-9_0">
            <onentry>
                <v9:assign stateId="-9" param="eval_key" value="5" type="1"/>
            </onentry>
            <transition event="evt.next" target="-9_1"/>
            <transition event="evt.end" target="end"/>
        </state>
        <state id="-9_1">
            <onentry>
                <v9:assign param="eval_key_val" value="非常不满意" type="1"/>
            </onentry>
            <transition event="evt.next" target="3"/>
            <transition event="evt.end" target="end"/>
        </state>
    </state>
</scxml>
```

基于百度的NLP实现电话机器人对话流程

```
<scxml xmlns="http://www.w3.org/2005/07/scxml" xmlns:v9="https://www.voice9.com" version="1.0" datamodel="jexl"
       initialstate="1">
    <datamodel>
        <data id="1" expr="开始_1"/>
        <data id="2" expr="结束_2"/>
        <data id="3" expr="loopNlp_3"/>
        <data id="4" expr="playAndAsr_4"/>
    </datamodel>
    <state id="1">
        <onentry>
            <v9:init record="1" asrId="6" ttsId="5" asrName="ali_asr" ttsName="ali_tts"/>
        </onentry>
        <transition event="evt.end" target="end"/>
        <transition event="evt.next" target="3"/>
        <transition event="evt.next" cond="_event.data=='null'"/>
        <transition event="evt.next" cond="_event.data=='end'" target="end"/>
        <transition event="evt.next" cond="_event.data=='3'" target="3"/>
        <transition event="evt.next" cond="_event.data=='4'" target="4"/>
    </state>
    <final id="end">
        <onentry>
            <v9:end/>
        </onentry>
    </final>

    <!-- 百度NGP接口  -->
    <state id="3">
        <onentry>
            <v9:loopNlp stateId="3" url="https://api-ngd.baidu.com/core/v3/start" method="POST"
                        contentType="application/json"
                        header='[{"Authorization":"NGD 12ba71ba-a3bf-46b6-a992-bb2f2e637024"}]'
                        body='{"sessionId": "#{[callId]}","ext": {"uid": "0000001","username": "ngd"}}'
                        response="welcome" timeout="百度NLP开白场接口请求超时"/>
        </onentry>
        <transition event="evt.end" target="end"/>
        <transition event="evt.next" target="5"/>
        <transition event="evt.timeout" cond="_event.data==timeout" target="end"/>
    </state>

    <state id="4">
        <onentry>
            <v9:loopNlp stateId="4" url="https://api-ngd.baidu.com/core/v3/query" method="POST"
                        contentType="application/json"
                        header='[{"Authorization":"NGD 12ba71ba-a3bf-46b6-a992-bb2f2e637024"}]'
                        body='{"queryText": "#{[asrResp]}","sessionId": "#{[callId]}"}'
                        response="answerText" response2="voice" timeout="百度NLP接口请求超时"/>
        </onentry>
```

```
        <transition event="evt.end" target="end"/>
        <transition event="evt.next" target="5"/>
        <transition event="evt.timeout" cond="_event.data==timeout" target="end"/>
    </state>

    <state id="5">
        <onentry>
            <v9:playAndAsr stateId="5" playType="2" ossId="#{[nlpResp]}"/>
        </onentry>
        <transition event="evt.end" target="end"/>
        <transition event="evt.next" target="4"/>
        <transition event="evt.timeout" cond="_event.data==timeout" target="end"/>
    </state>
</scxml>
```

五、实时订阅语音流

curl –X POST "{host}/fs–api/call/stream" –H "token: "{token}" –H "Content–Type: application/json"

request body:

JSON 复制代码

```json
{
  "callId":413672522823761920,    --通话中的callId
  "deviceId":"3005434257867544", --通话中的设备号
  "host":"172.17.0.2",           --媒体流接收端地址
  "port":12000                   --媒体流接受端口
}
```

先通过程序监听到呼叫平台的电话应答或者桥接成功，再发送订阅语音流拿到udp数据包。

实时asr识别结果

```
2023-04-18 22:22:02.351 - http-nio-7210-exec-2 - INFO - com.voice9.cc.fs.stream.UdpStream->83: callId:436290237354737664 deviceId:7238139693523427 received stream on 172.17.0.2:18838
2023-04-18 22:22:02.358 - http-nio-7210-exec-2 - INFO - com.voice9.cc.fs.stream.UdpStream->130: callId:436290237354737664, deviceId:7238139693523427, port:18838
2023-04-18 22:22:02.361 - http-nio-7210-exec-2 - INFO - com.voice9.cc.fs.esl.internal.AbstractEslClientHandler->158: send:/172.17.0.2:7400 message:bgapi uuid_rst  7238139693523427 172.17.0.2 18838
2023-04-18 22:22:02.382 - nioEventLoopGroup-8-1 - INFO - com.voice9.cc.fs.stream.UdpStream$1->97: callId:436290237354737664 deviceId 7238139693523427 received udp stream INV
2023-04-18 22:22:02.429 - fs-pool-28 - INFO - com.voice9.cc.fs.handler.FsMediaBugStartHandler->29: callId:436290237354737664 deviceId:7238139693523427 MEDIA_BUG_START
2023-04-18 22:22:02.538 - nioEventLoopGroup-8-1 - INFO - com.voice9.cc.service.impl.AiSpeechServiceImpl->279: get token: 1bbc2a5839fc4de3898e118507d849e4 , expire time:
2023-04-18 22:22:02.815 - nioEventLoopGroup-9-1 - INFO - com.voice9.cc.service.impl.AiSpeechServiceImpl$2->165: task_id: 982fc42814a84ec88c57c96b79ee3656, name: TranscriptionStarted, status: 20000000
2023-04-18 22:22:02.815 - nioEventLoopGroup-8-1 - INFO - com.voice9.cc.service.impl.AiSpeechServiceImpl->219: transcriber start deviceId:7238139693523427 taskId:982fc42814a84ec88c57c96b79ee3656
2023-04-18 22:22:02.878 - nioEventLoopGroup-9-1 - INFO - com.voice9.cc.service.impl.AiSpeechServiceImpl$2->170: task_id: 982fc42814a84ec88c57c96b79ee3656, name: SentenceBegin, status: 20000000
2023-04-18 22:22:03.038 - nioEventLoopGroup-9-1 - INFO - com.voice9.cc.service.impl.AiSpeechServiceImpl$2->189: resule:所，time:360, callId:436290237354737664, deviceId:7238139693523427 982fc42814a84ec88c57c96b79ee3656

2023-04-18 22:22:03.325 - nioEventLoopGroup-9-1 - INFO - com.voice9.cc.service.impl.AiSpeechServiceImpl$2->189: resule:所有的，time:720, callId:436290237354737664, deviceId:7238139693523427 982fc42814a84ec88c57c96b79ee3656

2023-04-18 22:22:03.803 - nioEventLoopGroup-9-1 - INFO - com.voice9.cc.service.impl.AiSpeechServiceImpl$2->189: resule:所有的平台都，time:1200, callId:436290237354737664, deviceId:7238139693523427 982fc42814a84ec88c57c96b79ee3656


2023-04-18 22:22:04.277 - nioEventLoopGroup-9-1 - INFO - com.voice9.cc.service.impl.AiSpeechServiceImpl$2->189: resule:所有的平台都希望，time:1680, callId:436290237354737664, deviceId:7238139693523427 982fc42814a84ec88c57c96b79ee36
2023-04-18 22:22:04.753 - nioEventLoopGroup-9-1 - INFO - com.voice9.cc.service.impl.AiSpeechServiceImpl$2->189: resule:所有的平台都希望说那个，time:2160, callId:436290237354737664, deviceId:7238139693523427 982fc42814a84ec88c57c96b
656
2023-04-18 22:22:05.381 - nioEventLoopGroup-9-1 - INFO - com.voice9.cc.service.impl.AiSpeechServiceImpl$2->189: resule:所有的平台都希望说那个就，time:2800, callId:436290237354737664, deviceId:72381396
e3656
2023-04-18 22:22:05.708 - nioEventLoopGroup-9-1 - INFO - com.voice9.cc.service.impl.AiSpeechServiceImpl$2->189: resule:所有的平台都希望说那个就是呃，time:3120, callId:436290237354737664, deviceId:7238139693523427 982fc42814a84ec88c
b79ee3656
2023-04-18 22:22:06.197 - nioEventLoopGroup-9-1 - INFO - com.voice9.cc.service.impl.AiSpeechServiceImpl$2->189: resule:所有的平台都希望说那个就是呃帮你解，time:3600, callId:436290237354737664, deviceId:7238139693523427 982fc42814a8
c57c96b79ee3656
2023-04-18 22:22:06.903 - nioEventLoopGroup-9-1 - INFO - com.voice9.cc.service.impl.AiSpeechServiceImpl$2->189: resule:所有的平台都希望说那个就是呃帮你解决这个事，time:4120, callId:436290237354737664, deviceId:7238139693523427 982f
4a84ec88c57c96b79ee3656
2023-04-18 22:22:07.164 - nioEventLoopGroup-9-1 - INFO - com.voice9.cc.service.impl.AiSpeechServiceImpl$2->189: resule:所有的平台都希望说那个就是呃帮你解决这个事情就是还，time:4560, callId:436290237354737664, deviceId:72381396935234
82fc42814a84ec88c57c96b79ee3656
2023-04-18 22:22:08.247 - nioEventLoopGroup-9-1 - INFO - com.voice9.cc.service.impl.AiSpeechServiceImpl$2->175: result:所有的平台都希望说那个就是呃帮你解决这个事情，就是还非常的。 confidence:0.847 begin_time:0 time:4980 callId:4362
354737664 deviceId:7238139693523427
```