新架构ivr交互查询

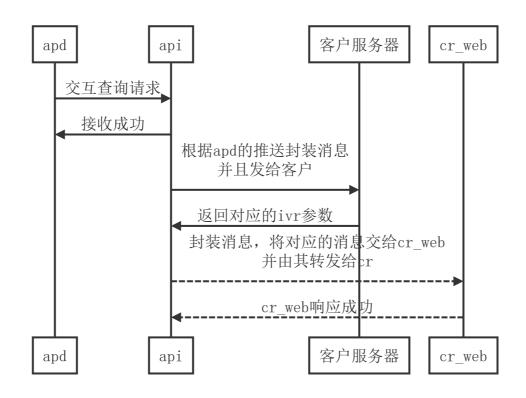
1需求描述

新架构ivr交互查询兼容老架构98,99查询被叫协议,并在此基础上新增了三种协议,分别是95,96,97。

通用的修改需求:

- 1. 新架构企业,通过ccgeid获取企业信息
- 2. 新架构企业,确定用户请求和用户响应格式
- 3. 新架构企业,通过CR-Web提供的查询响应接口返回数据
- 1.1 95(交互收键模式)
- 1.2 96(被叫查询模式)
- 1.3 97(通用交互模式)
- 1.4 98(AMNB或者AMXNB模式)
- 1.5 99(AMB或者AMXB模式)

2 交互流程



3 消息定义

3.1 apd->api

3.1.1 95(交互收键模式)

```
1
    {
 2
        "type": 95,
 3
        "ccgeid": 1576,
        "callId": "apixxxxxxxxxxxxx",
 4
        "ccNumber": "13260278209conf0_1519439781636",
        "callType" : 0,
 6
 7
        "ivrFlowId" : 245,
        "ivrQueryId" : 1,
 8
 9
        "caller": "13260278209",
10
        "callerType" : 1,
        "switchNumber" : "02566699794",
11
12
        "called" : "1001",
13
        "calledType" : 2,
        "timestamp": "1519439787",
15
        "userQueryId" : "id_0000001",
        "inputKeys" : "1000",
16
        "variables" : [
17
            {"id_number" : "110108198703127621" },
18
19
            {"name" :"" },
20
            {"address":"" }
        ]
21
    }
22
23
```

3.1.2 96(被叫查询模式)

```
1
    {
 2
        "type": 96,
 3
        "ccgeid": 1576,
 4
        "callId": "apixxxxxxxxxxxxx",
 5
        "ccNumber": "13260278209conf0_1519439781636",
 6
        "callType" : 0,
        "ivrFlowId" : 245,
 7
        "ivrQueryId" : 1,
 9
        "caller": "13260278209",
        "callerType" : 1,
10
11
        "switchNumber": "02566699794",
        "called" : "1001",
12
        "calledType" : 2,
13
14
        "timestamp" : "1519439787",
        "userQueryId" : "id_0000001",
15
        "inputKeys" : "1000",
16
        "variables" : [
17
            {"id_number" : "110108198703127621" },
18
19
            {"name": "" },
            {"address": "" }
20
21
        ]
22 }
```

3.1.3 97(通用交互模式)

```
1
 2
        "type": 97,
 3
        "ccgeid": 1576,
        "callId": "apixxxxxxxxxxxxx",
 4
        "ccNumber": "13260278209conf0_1519439781636",
 5
 6
        "callType": 0,
        "ivrFlowId" : 245,
 7
 8
        "ivrQueryId" : 1,
 9
        "caller": "13260278209",
10
        "callerType" : 1,
        "switchNumber" : "02566699794",
11
        "called" : "1001",
12
13
        "calledType" : 2,
14
        "timestamp": "1519439787",
        "userQueryId" : "id_0000001",
15
        "inputKeys" : "1000",
16
17
        "variables" : [
            {"id_number" : "110108198703127621" },
18
19
            {"name" :"" },
            {"address":"" }
20
21
        ]
22 }
```

3.1.4 98(AMNB或者AMXNB模式)

```
1 | {
 2
        "type" : 98,
 3
        "ccgeid": 1576,
 4
        "callId": "apixxxxxxxxxxxx",
        "ccNumber" : "13260278209conf0_1519439781636",
 5
        "ivrFlowId" : 34,
 6
 7
        "ivrQueryId" : 35,
        "switchNumber": "02566699794",
9
        "useNumber": "02566699794",
        "caller": "13260278209",
10
11
        "callerType" : 1,
        "timestamp": "1519439787",
12
        "path" : "1000"
13
14 }
```

3.1.5 99(AMB或者AMXB模式)

```
1
   {
 2
        "type" : 99,
        "ccgeid" : 1576,
 3
        "callId" : "apixxxxxxxxxxxxx",
 4
 5
        "ccNumber": "13260278209conf0_1519439781636",
        "ivrFlowId" : 34,
 6
 7
        "ivrQueryId" : 35,
        "switchNumber": "02566699794",
 8
 9
        "useNumber": "02566699794",
        "caller": "13260278209",
10
11
        "callerType" : 1,
12
        "timestamp" : "1519439787",
        "path" : "1000"
13
14 }
```

3.2 api内部消息转换

根据前面的需求描述

1. 根据ccgeid从企业表中获取企业信息

根据ccgeid获取企业信息,我们不需要通过callid或者总机号才能获取到企业信息,需要注意的是。后面处理时还会尝试获取一次企业的详细信息,对于新架构,我们需要将企业的信息发送过去,以减少一次数据库操作(但是enterprise结构体的大小对于推送的消息队列来说过于庞大了)。

2. 新架构的通用ivr交互新增了部分通用字段,有些字段api必须转发给客户。需要新增一个结构体用来存储这些信息

| 通话类型 | IVR流程 | callType | caller | callerType | called | calledType |
|----------|------------|----------|-----------|------------|-----------|------------|
| 呼入 | 正常流程 | 5 | 客户号码 | 1 | - | - |
| 呼入 | 子IVR流程 | 5 | 客户号码 | 1 | 坐席话 机号 | 1/2 |
| 呼出 | 子IVR流程 | 0或7 | 坐席话机 号 | 1/2 | 客户号 码 | 1 |
| 语音通 知 | 语音通知 流程 | 3 | 总机号码 | 1 | 客户号 码 | 1 |

```
1 typedef struct __push_post_data_ivr_moudle__
2 {
     DB_CALL_RECORD_CALL_TYPE calltype; //联合calltype caller
3
  called,可以得知在此次ivr请求中,哪个是客户,哪个是坐席
4
     | 通话类型 | IVR流程 | callType | caller | callerType | called
5
     | calledType |
    6
   ----- | ------ |
    | 呼入 | 正常流程 | 5 | 客户号码 | 1 | -
   | 呼入 | 子IVR流程 | 5 | 客户号码 | 1 | 坐席话机号
8
   | 1/2 |
   | 呼出
           | 子IVR流程 | O或7 | 坐席话机号 | 1/2
                                            | 客户号码
   | 语音通知 | 语音通知流程 | 3 | 总机号码 | 1 | 客户号码
  | 1 |
11
     unsigned long ivrFlowId; //ivr流程id
12
13
     unsigned long ivrQueryId; //ivr查询节点id
     char userQueryId[64]; //用于向客户服务器确定id对应下一步的操作是什
14
     char inputKeys[64]; //推送上次输入的按键
15
16
     char
              *variables;
17
     用户自定义查询请求变量值集合,是{"variables": [{"id_number":
18
   "110108198703127621"}, {:"name" :""}, {"address":""} ]},
19
```

```
20
    }ModCallPushPostDataIVRMoudle;
21
22
    typedef struct __push_post_data__
23
24
        unsigned char
                                      callid[CALL_RECORD_CALLID_MAX_LEN];
25
        unsigned char
                                      ccNumber[CALL_RECORD_CALLID_MAX_LEN];
        unsigned char
26
                                      caller[PHONE_NUMBER_MAX_LEN];
27
        unsigned char
                                      called[PHONE_NUMBER_MAX_LEN];
28
        int
                                      xferTimes;
29
        BOOL
                                      extCaller;
        BOOL
30
                                      extCalled;
31
        BOOL
                                      isCaller;
32
        DB_CALL_RECORD_CALL_TYPE
                                      type;
        DB_CALL_RECORD_STATUS
33
                                      status;
34
        unsigned char
                                      useNumber[PHONE_NUMBER_MAX_LEN];
35
        unsigned char
                                      switchNumber[PHONE_NUMBER_MAX_LEN];
36
        unsigned char
                                      subNumber[PHONE_NUMBER_MAX_LEN];
37
        unsigned char
                                     virtNumber[PHONE_NUMBER_MAX_LEN];
        unsigned long
38
                                      enterpriseId;
39
        unsigned long
                                      ccgeid;
        unsigned long
                                      ringTime;
40
41
        unsigned long
                                      startTime;
42
        unsigned long
                                      endTime;
43
        unsigned long
                                      timestamp;
44
        unsigned long
                                      duration;
        unsigned long
45
                                      reason;
46
        unsigned long
                                      gid;
47
        unsigned long
                                     pbxCallLogId;
48
        char
                                      feedback[CALL_RECORD_FEEDBACK_MAX_LEN];
49
        ModCallPushPostDataIVRMoudle
                                          *ivr_argv;
                                                          //新架构通用ivr扩展字段
50
51
        BOOL
                                      realtimeData;
        BOOL
52
                                      isCheckData:
53
        BOOL
                                      isCommonEnterprise;
54
        BOOL
                                     hangup2calling;
55
        int
                                      index;
56
        int
                                      lwpid;
57
        int
                                      mes_type;
58
        int
                                      failed_delay_time;
59
        unsigned char
                                     path[CALL_RECORD_FEEDBACK_MAX_LEN];
                                                                             //99
    协议 , 按键 (可能包含二级按键, 比如:2-9)
60
61
        unsigned long
                                      app_id;
        unsigned long
62
                                      provinceId;
63
        char
                                      number[USER_NUMBER_MAX_LEN];
        char
64
                                      mobile[PHONE_NUMBER_MAX_LEN];
65
        char
                                      destNumber[USER_NUMBER_MAX_LEN];
66
        unsigned long
                                      ngnReason;
     //201708-N02细化通话失败和挂断原因
67
        char
                                      batchCallId[CALL_RECORD_CALLID_MAX_LEN];
        char
68
                                      batchCalluserData[USER_DATA_MAX_LEN];
69
        char
                                      batchCallTaskId[BATCH_TASK_ID_MAX_LEN];
    } ModCallPushPostData;
70
71
72
    typedef struct __app_server_callback_argument__
73
74
        BOOL
                 isPiccCallbackArg;
75
        BOOL
                 isCommonEnterprise;
```

```
BOOL packageBalanceEnough;
 76
 77
         int
                callbackTimes;
 78
         int
                appCallbackDataFormat:
 79
         int
                call_detail_num;
 80
         unsigned long app_id;
 81
         unsigned long provinceId;
 82
         unsigned long enterpriseId;
 83
 84
         unsigned long curTimestamp;
 85
         unsigned long timestamp;
         unsigned char keyFeedback[CALL_RECORD_FEEDBACK_MAX_LEN];
 86
 87
         unsigned char destNumber[USER_NUMBER_MAX_LEN];
 88
         PiccCallBackArgument *piccArgs;
 89
 90
        ModCallPushPostDataIVRMoudle
                                        *ivr_argv;
                                                       //新架构通用ivr扩展字段,记
     得释放
 91
        DB_CALL_RECORD_CALL_TYPE
                                               //此次推送的通话类型,根据ivr文档描
                                    type;
     述call_records和call_detail中的type已经不能标识这次的推送通话类型了
 92
 93
         APP_CALLBACK_TYPE
                            callbackType;
 94
         char AppServerUrl[CALLBACK_URL_MAX_LEN];
 95
         // MYSQL *pDb;
                                                      // Only for local
     database access
96
 97
         // ModCallPushPostData *post_data;
98
         EMICALLDEV_DB_CALL_RECORDS *call_record;
99
         EMICALLDEV_DB_CALL_DETAILS *call_details;
100
         EMICALLDEV_DB_CALL_DETAILS *cur_call_detail;
101
        EMICALLDEV_DB_ACCOUNTS
                                   *mainAccount;
102
         EMICALLDEV_DB_APPLICATIONS *appInfo;
103
         EMICDEV_DB_COMMON_ENTERPRISES *comm_enterprise;
104
         EMICDEV_DB_ENTERPRISES *enterprise;
105
         ENTERPRISE_BASE_INFO
                                    *eInfo;
106
         EMICALLDEV_DB_CALL_RECORD_CC_EXTENDS *call_record_extends;
107
108
         char downloadurl[HTTP_URL_MAX_LEN];
109
     } AppServerCallbackArgument;
```

3.2 api->用户服务器

请求格式和用户响应格式由客户进行配置,可以是json或者xml

3.2.1 请求客户

1.99/9899和98向客户的请求信息保持不变

json

```
1
    {
 2
         'appId': 'b23abb6d451346efa13370172d1921ef',
 3
         'callid':'api1234059445aDbbJxIdbT',
 4
         'accountSid':'c5dc4b87f33ef2ef37c8e974793ad8e5',
 5
         'caller': '18769874345',
 6
        'path': '1-2',
 7
         'type':0,
 8
        'callType':99,
 9
         'useNumber': '02566687987',
         'switchNumber': "02566699794",
10
11
        'userData':"FE87D3"
12
    }
```

2.95,96和97消息需要使用新架构新的消息格式

新架构的查询场景不再限于呼入场景,api客户可能需要知道主叫或被叫是坐席还是客户,callType需要 传给客户

```
1
    {
 2
        "type": 95/96/97,
 3
        //"initialCallType": 1,
 4
        //一开始发起的通话类型
 5
        "callType": 1,
        "accountSid": "c5dc4b87f33ef2ef37c8e974793ad8e5",
 6
        "subAccountSid": "c5dc4b87f33ef2ef37c8e974793ad8e5",
 7
 8
        'appId':'b23abb6d451346efa13370172d1921ef',
 9
10
        "callId": "apixxxxxxxxxxxxx",
        "caller": "13260278209",
11
12
        "called" : "1001",
13
        "useNumber": "02566699794",
        "userQueryId": "id_0000002",
14
15
        "inputKeys" : "1000",
        "variables" : [
16
           {"id_number" : "110108198703127621" },
17
            {"name" :"" },
18
19
            {"address":"" }
20
        ],
21
        'userData':"FE87D3"
22
23
24
    说明:在上例中,有3个全局变量:id_number、name和address,id_number已经赋值,name和
    address未赋值,需要用户服务器返回,并在IVR其它节点中引用。
```

```
1
    <request>
 2
        <callType>5</callType>
 3
        <type>97</type>
 4
        <accountSid>42f7de84ff2ea9b4d71c2aa667455249</accountSid>
 5
        <subAccountSid>51d17d870e0c1da85869580195b31f1d/subAccountSid>
 6
        <appId>68a87b1250011b35cfb244f04a27ca9d</appId>
 7
        <callId>1582511927.526072</callId><caller>15861800293</caller>
 8
        <called>0252133</called>
 9
        <useNumber>02566687671</useNumber><usrQueryId>1</usrQueryId>
10
        <inputKeys>1</inputKeys>
11
        <userData></userData>
12
        <variables>
```

3.2.2 客户响应

3.2.2.1 98/99返回数据

```
1
   {
 2
       "response":{
 3
           //0:允许通话,否则失败
 4
           "retcode": 0,
           //返回被叫
 5
 6
           "action": 0,
           "called": "1****211",
 7
           "number": "1002",
8
9
           "workNumber": "test",
10
           "waitTime": 20,
11
           "outNumber": "02566687**1",
           "reason": "原因描述",
12
           "userdata": "用户数据"
13
14
       }
15
   //其中number>workNumber>called,这三个都是被叫选项,三者选一。被叫如果传多个,会按顺序
16
   拨打直至拨打完毕
```

3.2.2.2 95/96/97

通用格式

```
1
   {
2
       "response":{
3
          "retcode": 0,
4
          //客户响应数据都将透传给cr, 具体含义见api->cr_web的定义
          //可以为空,也可以和请求保重的userQueryId相同,也可以不同
6
          //不为空时,下次的交互查询节点会直接赋值userQueryId
7
          "userQueryId":"id_0000002",
8
9
          //variables是请求包中variables的子集,需要赋值的参数必须要通过这个列表中返回
          "variables": [
10
11
             {"id_number" : "110108198703127621" },
             {"name" :"" },
12
13
             {"address":"" }
14
          //虚拟键值,交互收键模式时的响应参数
15
16
          "virtualKey":"1111",
          //用户返回的下一步参数,这个数据应当由api透传给cr,现cr已经定义好响应数据,所以
17
   和需求文档中的格式略有不同,具体数据参数见api->cr_web的消息定义
          "nextAction" : {
18
             "action" : 1,
19
20
             "params" : {
                 "voiceId": "播放语音文件id",
21
                 "voiceName": "播放语音文件唯一名称",
22
                 "allowBreak": "是否允许打断: 0-不允许 1-允许"
23
```

```
<?xml version="1.0" encoding="utf-8"?>
 2
    <response>
 3
        <retcode>0</retcode>
 4
        <userQueryId>1</userQueryId>
 5
        <userData>FE87D3</userData>
        <variables>
 6
 7
            <a725>1582726762</a725>s
 8
            <as725>1582726762</as725>
9
        </variables>
10
11
        <nextAction>
12
            <action>2</action>
13
            <params>
14
                <voiceId>1</voiceId>
15
                <voiceName>dwad</voiceName>
16
                 <allowBreak>1</allowBreak>
17
            </params>
18
        </nextAction>
19
   </response>
```

3.2.2.2.1 nextAction放音

json

```
1
   "nextAction" : {
2
       "action" : 1,
3
       "params" : {
4
          //下面四个参数选择其中之一
          "voiceId": "播放语音文件id",
5
6
          "voiceName": "播放语音文件唯一名称",
7
          "voiceTempId": "播放语音模板id",
          "voiceTempName": "播放语音模板名称",
8
          //语音模板参数,多个参数见用英文","号隔开,当选择voiceTempId或者
9
   voiceTempName时需要传入
10
          "voiceTempParams": "语音模板参数",
11
          "allowBreak": "是否允许打断: 0-不允许 1-允许"
12
13
       }
14 }
```

xml

3.2.2.2.2 nextAction放音收键

json

```
1
   "nextAction" : {
       "action" : 2,
2
3
       "params" : {
4
          //下面四个参数选择其中之一
5
          "voiceId": "播放语音文件id",
          "voiceName": "播放语音文件唯一名称",
6
7
          "voiceTempId": "播放语音模板id",
8
          "voiceTempName": "播放语音模板名称",
          //语音模板参数,多个参数见用英文","号隔开,当选择voiceTempId或者
9
   voiceTempName时需要传入
10
          "voiceTempParams": "语音模板参数",
11
12
          "allowBreak": "是否允许打断: 0-不允许 1-允许",
13
          "getKeyNumber": "获取按键位数",
          "getKeyTimeout": "收键超时时间",
14
          "endwithHashKey": "是否以#号键结束, 0-不是, 1-是"
15
16
      }
17 }
```

3.2.2.2.3 nextAction转技能组

```
"nextAction" : {
       "action": 3,
2
3
       "params" : {
          "acdId": "技能组id",
4
5
          "acdName": "技能组名称",
          "useAcdValue": "0-不使用技能组配置 1-使用技能组配置",
6
7
          "queueTime": "排队超时时长",
          "switchTimes": "坐席流转次数",
8
          "ringTimeout": "坐席振铃超时时长",
9
10
          "customerMemory": "0-不记忆 1-优先熟客记忆 2-强制熟客记忆"
       }
11
12 }
```

3.2.2.2.4 nextAction转坐席

```
"nextAction" : {
1
2
      "action": 4,
3
      "params" : {
          "workNumber" : "1001,1002,1003",
4
5
          "number" : "1001,1002,1003",
6
          "queueTime": "坐席忙时排队时长",
7
          "ringTimeout": "多坐席情况下,坐席振铃超时时长",
8
     }
9 }
```

3.2.2.2.5 nextAction转外线

```
1 "nextAction": {
2     "action": 5,
3     "params": {
4         "called": "外线被叫号码",
5         "outNumber": "呼出总机号码"
6     }
7 }
```

3.2.2.2.6 nextAction转其他IVR流程

```
1 "nextAction": {
2     "action": 6,
3     "params": {
4         "ivrFlowId": "IVR流程id",
5         "ivrFlowName": "IVR流程名称"
6     }
7 }
```

3.2.2.2.7 nextAction结束IVR流程

```
1 "nextAction" : {
2     "action" : 7
3 }
```

3.3 api->cr_web

新架构cr采用统一的json格式,所以不管是95、96、97、98还是99,api需要将客户返回的消息转换成cr需要的格式,而不需要校验。api需要在对外文档中列出这些响应数据对应的格式。

以下是cr根据action定义的响应消息内容,需要根据这些重新生成给cr的json响应,然后通过cr_web透传给cr

为方便路由,要求将ccgeid和cc_number作为头域参数

action

```
typedef enum
2
3
    0, invalid
    1, 放音响应
     2, 放音按键响应
    3, 转技能组响应
    4, 转坐席响应
7
8
    5, 转外线响应
     6, 转其他IVR流程响应
9
    7, 流程结束响应
10
11 }IvrActionType;
```

3.3.1 交互收键模式响应消息

```
1 {
 2
       "eid" : "00011",
3
       "ccgeid": "111222",
4
       "request_type": "1", //标识是通话控制请求还是交互式ivr响应
       "class_type" : "1", //
 5
       "ccNumber" : "",
 6
 7
       "data" :
8
9
       {
          "rspCode" : 0,
10
           "userQueryId":"id_0000002",
11
12
          "virtualKey" : "5",
          "variables" : [
13
               { "id_number" : "110108198703127621" },
14
               { "name":"张三"},
15
               { "address":"江苏省南京市江宁区" }
16
17
           ]
           "reason" : "test",
18
           "userdata" : "test"
19
20
       }
21
22 }
```

3.3.2 被叫查询和通用交互模式响应消息

3.3.2.1 放音响应

json to cr

```
1 {
       "eid" : "00011",
 2
       "ccgeid" : "111222",
 3
4
       "request_type": "1", //标识是通话控制请求还是交互式ivr响应
       "class_type" : "1", //
 5
       "ccNumber" : "",
 6
       "data":"
7
8
           "rspCode" : 0,
9
           "userQueryId":"id_0000002",
10
          "variables" : [
               { "id_number" : "110108198703127621" },
11
```

```
{ "name" :"张三" },
12
13
              { "address": "江苏省南京市江宁区" }
14
           ],
           "nextAction" : {
15
              "action" : 1,
16
17
              "params" : {
18
                  //下面四个参数选择其中之一
                  "voiceId": "播放语音文件id",
19
20
                  "voiceName": "播放语音文件唯一名称",
21
                  "voiceTempId": "播放语音模板id",
                  "voiceTempName": "播放语音模板名称",
22
23
                  //语音模板参数,多个参数见用英文","号隔开,当选择voiceTempId或者
   voiceTempName时需要传入
24
                  "voiceTempParams": "语音模板参数",
25
                  "allowBreak": "是否允许打断: 0-不允许 1-允许"
26
27
              }
28
           },
29
           "reason" : "test",
           "userdata" : "test"
30
31
32 }
```

3.3.2.2 放音收键响应

ison to cr

```
1
   {
 2
       "eid" : "00011",
 3
       "ccgeid": "111222",
       "request_type" : "1",
 4
                            //标识是通话控制请求还是交互式ivr响应
 5
       "class_type" : "1",
                             //
 6
       "ccNumber" : "",
                             //
 7
       "data":"
           "rspCode" : 0,
           "userQueryId":"id_0000002",
9
           "variables" : [
10
11
              { "id_number" : "110108198703127621" },
               { "name" :"张三" },
12
               { "address": "江苏省南京市江宁区" }
13
14
           ],
           "nextAction" : {
15
               "action" : 2,
16
17
               "params" : {
                  //下面四个参数选择其中之一
18
19
                  "voiceId": "播放语音文件id",
                  "voiceName": "播放语音文件唯一名称",
20
                  "voiceTempId": "播放语音模板id",
21
22
                  "voiceTempName": "播放语音模板名称",
                  //语音模板参数,多个参数见用英文","号隔开,当选择voiceTempId或者
23
    voiceTempName时需要传入
24
                  "voiceTempParams": "语音模板参数",
25
                  "allowBreak": "是否允许打断: 0-不允许 1-允许",
26
27
                  "getKeyNumber": "获取按键位数",
28
                  "getKeyTimeout": "收键超时时间",
                  "endwithHashKey": "是否以#号键结束, 0-不是, 1-是"
29
```

3.3.2.3 转技能组响应

json to cr

```
1
   {
       "eid" : "00011",
 2
 3
       "ccgeid": "111222",
       "request_type" : "1",
 4
                             //标识是通话控制请求还是交互式ivr响应
 5
       "class_type" : "1",
                              //
       "ccNumber" : "",
 6
                              //
 7
       "data":"
 8
           "rspCode" : 0,
           "userQueryId":"id_0000002",
 9
           "variables" : [
10
               { "id_number" : "110108198703127621" },
11
               { "name" :"张三" },
12
13
               { "address":"江苏省南京市江宁区" }
14
           ]
           "nextAction" : {
15
               "action" : 3,
16
               "params" : {
17
18
                   "acdId": "技能组id",
19
                   "acdName": "技能组名称",
                   "useAcdValue": "0-不使用技能组配置 1-使用技能组配置",
20
                   "queueTime": "排队超时时长",
21
                   "switchTimes": "坐席流转次数",
22
23
                   "ringTimeout": "坐席振铃超时时长",
24
                   "customerMemory": "0-不记忆 1-优先熟客记忆 2-强制熟客记忆"
25
               }
26
           }
           "reason" : "test",
27
           "userdata" : "test"
28
29
30 }
```

3.3.2.4 转座席响应

json to cr

```
1
   {
 2
        "eid" : "00011",
 3
        "ccgeid": "111222",
 4
        "request_type" : "1",
                              //标识是通话控制请求还是交互式ivr响应
 5
        "class_type" : "1",
        "ccNumber" : "",
 6
                               //
        "data":"
 7
 8
            "rspCode" : 0,
            "userQueryId":"id_0000002",
9
            "variables" : [
10
                { "id_number" : "110108198703127621" },
11
                { "name" :"张三" },
12
```

```
{ "address":"江苏省南京市江宁区" }
13
14
           ]
           "nextAction" : {
15
               "action" : 4,
16
               "params" : {
17
                   "workNumber" : "1001,1002,1003",
18
                   "number" : "1001,1002,1003",
19
                   "queueTime": "坐席忙时排队时长",
20
21
                   "ringTimeout": "多坐席情况下,坐席振铃超时时长"
22
               }
23
           },
24
           "reason" : "test",
           "userdata" : "test"
25
26
27 }
```

3.3.2.5 转外线响应

json to cr

```
1
   {
        "eid" : "00011",
 2
 3
        "ccgeid": "111222",
        "request_type" : "1",
 4
                              //标识是通话控制请求还是交互式ivr响应
        "class_type" : "1",
 5
                               //
        "ccNumber" : "",
 6
                               //
        "data":"
 7
 8
            "rspCode" : 0,
 9
            "userQueryId":"id_0000002",
            "variables" : [
10
               { "id_number" : "110108198703127621" },
11
                { "name" :"张三" },
12
               { "address":"江苏省南京市江宁区" }
13
14
            ]
            "nextAction" : {
15
               "action" : 5,
16
                "params" : {
17
18
                   "called": "外线被叫号码",
                   "outNumber": "呼出总机号码"
19
20
               }
21
            "reason" : "test",
22
           "userdata" : "test"
23
24
25 }
```

3.3.2.6 转其他IVR流程响应

json to cr

```
1
  {
2
      "eid" : "00011",
3
      "ccgeid": "111222",
4
      "request_type" : "1",
                            //标识是通话控制请求还是交互式ivr响应
5
      "class_type" : "1",
                             //
      "ccNumber" : "",
6
                             //
       "data":"
```

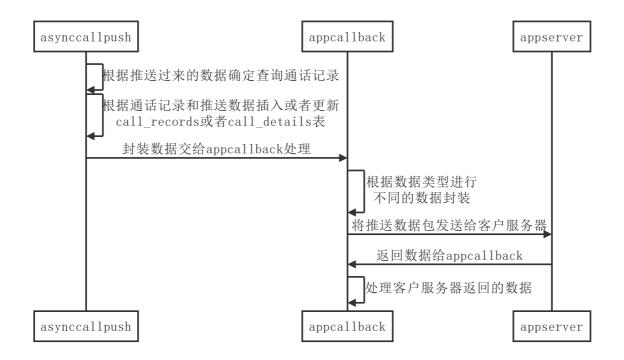
```
"rspCode" : 0,
8
9
            "userQueryId":"id_0000002",
            "variables" : [
10
               { "id_number" : "110108198703127621" },
11
               { "name" :"张三" },
12
13
               { "address":"江苏省南京市江宁区" }
14
15
            "nextAction" : {
               "action": 6,
16
                "params" : {
17
                   "ivrFlowId": "IVR流程id",
18
                   "ivrFlowName": "IVR流程名称"
19
20
               }
21
            }
            "reason" : "test",
22
           "userdata" : "test"
23
24
25 }
```

3.3.2.7 流程结束响应

```
1
   {
       "eid" : "00011",
 2
 3
        "ccgeid": "111222",
        "request_type" : "1",
                             //标识是通话控制请求还是交互式ivr响应
 4
 5
        "class_type" : "1",
                              //
 6
        "ccNumber" : "",
                              //
       "data":"
 7
           "rspCode" : 0,
 8
           "userQueryId":"id_0000002",
9
           "variables" : [
10
11
               { "id_number" : "110108198703127621" },
12
               { "name":"张三"},
               { "address": "江苏省南京市江宁区" }
13
14
           ]
           "nextAction" : {
15
               "action" : 7
16
17
           "reason" : "test",
18
19
           "userdata" : "test"
20
21 }
```

4 处理流程

• api callpush模块时序图



asynccallpush到appcallback只需要将ivr的特有数据透传就行。

appcallback到appserver因为type和calltype的含义已经变化,在原有的结构上进行扩展已经不适合了。下面是appcallback的新逻辑。

1 //

经过考察我们主要的修改是:

- 1. 发向appcallback队列的数据会进行扩展,存储ivr扩展信息。
- 2. 对于交互类型的数据,不能更新通话记录
- 3. 回调接口扩展,新增回调函数
- 4. 查询结果封装并返回,通过cr web接口

疑问:

- 1.98和99或者96和以前不同,会推送振铃请求
- 2. 目前type和calltype的定义比较混乱
- 3. 不支持重传

4.1 95(交互收键模式)

如果已有通话记录,则不对通话记录进行任何修改;如果没有通话记录,则该通话一定是呼入通话,需要在call_records和call_details表中生成原始通话记录,通话类型等于请求数据中的callType

4.2 96(被叫查询模式)

被叫查询api会直接根据消息数据入库,和之前的处理方式保持一致即可。

4.3 97(通用交互模式)

如果已有通话记录,则不对通话记录进行任何修改;如果没有通话记录,则该通话一定是呼入通话,需要在call_records和call_details表中生成原始通话记录,通话类型等于请求数据中的callType

4.4 98(AMNB或者AMXNB模式)

被叫查询api会直接根据消息数据入库,和之前的处理方式保持一致即可。

4.5 99(AMB或者AMXB模式)

被叫查询api会直接根据消息数据入库,和之前的处理方式保持一致即可。

5 涉及代码