IVR (Voice Messaging System) Home

Space Content:

Recent space activity

- **Dagmar Timler**
    - Support updated Jan 21, 2015 • view change

- **Achmat Samsodien**
    - Verboice Setup updated Oct 30, 2014 • view change

- **Dagmar Timler**
    - Verboice Setup commented Oct 30, 2014

- **Achmat Samsodien**
    - Verboice Setup commented Oct 30, 2014

- **Diné Bennett**
    - Verboice Setup updated Oct 30, 2014 • view change

Space contributors

- Dagmar Timler (1114 days ago)
- Achmat Samsodien (1198 days ago)
- Diné Bennett (1198 days ago)
- Raiyaan Smith (1351 days ago)

IVR REST API

Sections:

- IVR REST API - Contacts

IVR REST API - Campaigns

> **Important Note**
> For all these services, please set Content-Type:application/json

Add Campaign

```
HTTP POST {baseUrl}/service/campaigns
```

Note: Call Flow Name, Channel Name, Schedule Name and Verboice Project ID **MUST BE THE EXACT NAMES AND ID AS IT IS ON THE VERBOICE SERVER.**

Example request:

```
[
    {
        "name": "Stellenbosch Campaign",
        "description": "Main Campaign for Stellies",
        "duration": "5",
        "callFlowName": "Main Call Flow",
        "channelName": "Stellies Skype",
        "scheduleName": "Stellies Schedule",
        "verboiceProjectId": "10"
    }
]
```

Example response:

```
[
    {
        "id": 37,
        "name": "Stellenbosch Campaign",
        "description": "Main Campaign for Stellies",
        "duration": "5",
        "callFlowName": "Main Call Flow",
        "channelName": "Stellies Skype",
        "scheduleName": "Stellies Schedule",
        "verboiceProjectId": "10"
    }
]
```

Get Campaigns

```
HTTP GET {baseUrl}/service/campaigns
```

Example response:

```
[
  {
    "id": 3,
    "name": "Stellenbosch Campaign",
    "description": "Main Campaign for Stellies",
    "duration": 5,
    "callFlowName": "Main Call Flow",
    "channelName": "Stellies Skype",
    "scheduleName": "Stellies Schedule",
    "verboiceProjectId": 10
  }
]
```

Update Campaign

```
HTTP PUT {baseUrl}/service/campaigns/{campaignId}
```

Example request

```
[
    {
        "name": "Stellenbosch Campaign - Updated Name",
        "description": "Main Campaign for Stellies",
        "duration": "5",
        "callFlowName": "Main Call Flow",
        "channelName": "Stellies Skype",
        "scheduleName": "Stellies Schedule",
        "verboiceProjectId": "10"
    }
]
```

Set Messages for Campaign

NB: This will override and delete any previous campaign messages!!

```
HTTP POST {baseUrl}/service/campaigns/<campaignId>/campaignMessages
```

Example request:

```
[
    {
        "verboiceMessageNumber": 1,
        "messageTimeOfDay": "10:30",
        "messageDay": 1
    },
    {
        "verboiceMessageNumber": 2,
        "messageTimeOfDay": "14:00",
        "messageDay": 1
    },
    {
        "verboiceMessageNumber": 3,
        "messageTimeOfDay": "09:00",
        "messageDay": 2
    },
    {
        "verboiceMessageNumber": 4,
        "messageTimeOfDay": "14:00",
        "messageDay": 2
    },
    {
        "verboiceMessageNumber": 5,
        "messageTimeOfDay": "09:00",
        "messageDay": 3
    }
]
```

Get Messages in Campaign

```
HTTP GET {baseUrl}/service/campaigns/<campaignId>/campaignMessages
```

Example Response


```
[
    {
        "verboiceMessageNumber": 1,
        "messageTimeOfDay": "10:30",
        "messageDay": 1
    },
    {
        "verboiceMessageNumber": 2,
        "messageTimeOfDay": "14:00",
        "messageDay": 1
    },
    {
        "verboiceMessageNumber": 3,
        "messageTimeOfDay": "09:00",
        "messageDay": 2
    },
    {
        "verboiceMessageNumber": 4,
        "messageTimeOfDay": "14:00",
        "messageDay": 2
    },
    {
        "verboiceMessageNumber": 5,
        "messageTimeOfDay": "09:00",
        "messageDay": 3
    }
]
```


IVR REST API - Contacts

Important Note
For all these services, please set Content-Type:application/json

Get All Contacts in a Campaign

```
HTTP GET {baseUrl}/service/campaigns/{campaignId}/contacts
```

Example response:


```
[
    {
        "id": 284,
        "password": "2702",
        "msisdn": "27724194158",
        "progress": 0,
        "campaignId": 287,
        "verboiceContactId": null,
        "voided": false
    },
    {
        "id": 285,
        "password": "8075",
        "msisdn": "27768198075",
        "progress": 0,
        "campaignId": 287,
        "verboiceContactId": null,
        "voided": false
    },
    {
        "id": 286,
        "password": "6045",
        "msisdn": "27734276045",
        "progress": 0,
        "campaignId": 287,
        "verboiceContactId": null,
        "voided": false
    }
]
```


Add Contact

```
HTTP POST {baseUrl}/service/campaigns/{campaignId}/contacts
```

Example request:


```
[
    {
        "password": "2702",
        "msisdn": "27724194170"
    }
]
```

Example response:

```
{
    "id": 290,
    "password": "2702",
    "msisdn": "27724194171",
    "progress": 0,
    "campaignId": 287,
    "verboiceContactId": null,
    "voided": false
}
```

Update Contact

```
HTTP PUT {baseUrl}/service/campaigns/{campaignId}/contacts/{contactId}
```

Example request

```
{
    "password": "2702",
    "progress": 0
}
```

Delete (Void) Contact

```
HTTP DELETE {baseUrl}/service/campaigns/{campaignId}/contacts/{contactId}
```

IVR Test Matrix and execution results

ivr test matrix.xlsx

Test that have failed do have work arounds and have to be noted to the client. The web interface with the work around is ready for the pilot phase with the client.

Set Up

- Set Up Procedure for Stellenbosch VMMC IVR
- Set Up Procedure for a new VMMC campaign.
- Verboice Setup

Set Up Procedure for a new VMMC campaign.

Contents

1 Creating an Account on the Verboice Server

The server can be found here: http://196.26.21.45/ (Achmat might change this in the near future).

Decide what account you want to use, and make sure you enter the login details in the application.properties file on the vmmc server.

The vmmc dev server (Sol) account currently uses the following details:

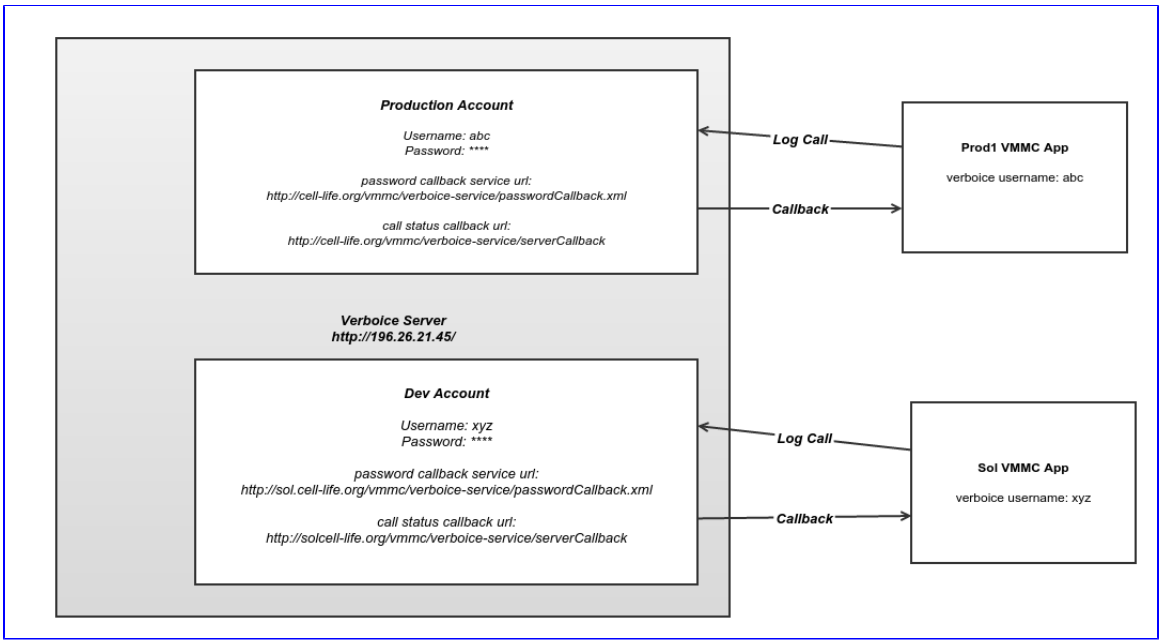
- verboice.username=dine@cell-life.org
- verboice.password=centuryscareddescribetwenty
- verboice.base_url=http://196.26.21.45/

You can choose to use a different account, but then ***you MUST edit the application.properties file accordingly***.

2 Creating your project on the Verboice Server

1. Log in to the Verboice server (http://196.26.21.45) with the account you decided on in step 1.
2. Create a new project with your project name, eg. 'Test 123 Project' OR you can use an existing project. Note the project Id.
3. Create new columns 'password', 'message' and 'password_entered' in the phonebook, if they don't already exist. (The names must be exactly those, or it won't work).
4. Under 'External Services', you need to add the callback service from the VMMC server.
    - If you are using the production server, this will be: http://cell-life.org/vmmc/verboice-service/passwordCallback.xml
    - For Sol it will be http://sol.cell-life.org/vmmc/verboice-service/passwordCallback.xml.
    - **NOTE:** The VMMC app needs to be running on the server for this to work.
5. Under 'Settings' add the callback URL
    - Production server: http://cell-life.org/vmmc/verboice-service/serverCallback
    - Dev server: http://sol.cell-life.org/vmmc/verboice-service/serverCallback
6. Create new call flow, or find an existing call flow you want to use eg. 'Test 123 Call Flow'
7. Create new Skype Channel or find the Skype Channel you want to use
8. Create a new call new schedule or find the schedule you want to use

Diagram of integration between Verboice server and VMMC server.



3 Set up the campaign on the VMMC server

Note: The complete REST service is documented HERE.

1. Created a new campaign with the following command.

Note that the verboice project ID should be the one you noted down in step 2.1 above. And the call flow, channel name and schedule name should be the ones from 2.6, 2.7 and 2.8 above.

```
POST {baseUrl}/service/campaigns
```

Request Body

```
[
    {
        "name": "Test 123 Campaign",
        "description": "A campaign to test.",
        "duration": "5",
        "callFlowName": "Test 123 Call Flow",
        "channelName": "My Skype Channel",
        "scheduleName": "My Schedule",
        "verboiceProjectId": "10"
    }
]
```

2. Set campaign messages with:

```
POST {baseUrl}/service/campaigns/{campaignId}/campaignMessages
```

Request Body

```
[
    {
        "verboiceMessageNumber": "1",
        "messageTimeOfDay": "9:00",
        "messageDay": 1
    },
    {
        "verboiceMessageNumber": "2",
        "messageTimeOfDay": "9:00",
  "messageDay": 2
    },
    {
        "verboiceMessageNumber": "3",
        "messageTimeOfDay": "9:00",
  "messageDay": 3
    },
    {
        "verboiceMessageNumber": "4",
        "messageTimeOfDay": "9:00",
  "messageDay": 4
    },
    {
        "verboiceMessageNumber": "5",
        "messageTimeOfDay": "9:00",
  "messageDay": 5
    }
]
```

4 Adding your contacts via the web interface

Add a CSV with the contacts.

NOTE:

1. The CSV should have two columns: msisdn and password. Column headers must be 'msisdn' and 'password'.
2. The msisdn should be in the format 27724194158.

5 Troubleshooting

5.1 "Error enqueuing call to Verboice. Call will be logged as 'waiting'."

If you see this in the log, it likely means that you are trying to use a Verboice project that does not belong the the account in Step 1.

Try and log into the verboice server with the account you decided on in Step 1. Check that the project you are trying to access indeed belongs to this account.

If it doesn't, use a project that does belong to this account.

5.2 My report does not reflect the call statuses.

This probably means the callbacks aren't working.

Make sure you have the callbacks set up correctly on the Verboice server, as described above.

Check the access log on Prod1/Sol to see if Verboice is sending callbacks to the correct machine when a call is made.

5.3 The log says "Reason: HTTP/1.1 404 Not Found"

If Verboice cannot find the channel, call flow name or shcedule you specified, it will send back 404 Not Found. Check that you have typed the correct names when you set up the campaign.

Set Up Procedure for Stellenbosch VMMC IVR

This is what I have done so far.

1. On Verboice server:

(http://196.26.21.45/projects/), under account 'developers@cell-life.org'

1. Created a new project with name Stellenbosch IVR.
2. Created new columns 'password', 'message' and 'password_entered' in the phonebook. (The names must be exactly those, or it won't work).
3. Under 'External Services', add the service as follows: http://sol.cell-life.org/vmmc/verboice-service/passwordCallback.xml
    a. **NOTE 1:** Obviously you need to replace the server with the correct name, not sol.
    b. **NOTE 2:** The VMMC app needs to be running on the server for this to work.
4. Under 'Settings' add the callback URL in this format: http://sol.cell-life.org/vmmc/verboice-service/serverCallback (again, obviously change sol to the correct server)
5. Created new call flow 'Main Call Flow.'
6. Created new Skype Channel *Stellies Skype*.
7. Created new schedule *Stellies Schedule*.

2. On Cell Life IVR app (via REST Client):

Note: The complete REST service is documented HERE.

1. Make sure you have the correct Verboice server url, username and password in the properties file.
2. Created a new campaign with:

```
POST {baseUrl}/service/campaigns
```

Request Body

```
[
    {
        "name": "Stellenbosch Campaign",
        "description": "Main Campaign for Stellies",
        "duration": "5",
        "callFlowName": "Main Call Flow",
        "channelName": "Stellies Skype",
        "scheduleName": "Stellies Schedule",
        "verboiceProjectId": "10"
    }
]
```

3. Set campaign messages with:

```
POST {baseUrl}/service/campaigns/3/campaignMessages
```

Request Body

```
[
    {
        "verboiceMessageNumber": "1",
        "messageTimeOfDay": "9:00",
        "messageDay": 1
    },
    {
        "verboiceMessageNumber": "2",
        "messageTimeOfDay": "9:00",
  "messageDay": 2
    },
    {
        "verboiceMessageNumber": "3",
        "messageTimeOfDay": "9:00",
  "messageDay": 3
    },
    {
        "verboiceMessageNumber": "4",
        "messageTimeOfDay": "9:00",
  "messageDay": 4
    },
    {
        "verboiceMessageNumber": "5",
        "messageTimeOfDay": "9:00",
  "messageDay": 5
    }
]
```

3. Via the web interface:

Add a CSV with the contacts.

NOTE:

1. The CSV should have two columns: msisdn and password. Column headers must be 'msisdn' and 'password'.
2. The msisdn should be in the format 27724194158.

Verboice Setup

Content

Each IVR/VMMC campaign requires: a project, a channel, call schedule and call flow to be set up. I will attempt to document this setup here.

Also see official Verboice documentation here: https://bitbucket.org/instedd/verboice/wiki/Home

Server Installation

For installation on Ubuntu 12.04 LTS please refer to:

https://bitbucket.org/instedd/verboice/wiki/Installation.md

Note: Cloud server providers only support Ubuntu 14.04 LTS release and there are some package differences which hinders the running of the Ruby installation

Setting up a project

Each project in Verboice can contain multiple call flows and multiple call schedules.

1. Log in and go to the projects page: http://196.26.21.45/projects
2. Select "Create New"
3. Enter the project name, timezone and language.
4. Select "Built-in" for text-to-speech engine.
5. Enter the URL to send call status updates to as well as the authentication details. (See 'Callbacks' below).
6. Next you need to set up a call schedule for the project. This is discussed below.

Callbacks

The callback URL is for the Verboice Call Status Notification. This is the notification Verboice sends to our server when the status of a call (e.g. in-progress, completed, no-answer etc.) changes. You need to enter the URL where you would like Verboice to send this notification to.

For VMMC, the following URLs can consume callbacks:

- http://www.cell-life.org/vmmc/verboice-service/serverCallback (for production)
- http://sol.cell-life.org/vmmc/verboice-service/serverCallback (for development)

- See CallLogController.updateCallState() in the IVR project for details on how this gets processed in the IVR jar.
- See Verboice documentation for more technical info: https://bitbucket.org/instedd/verboice/wiki/Connecting%20Calls.wiki#!call-status-notification

Setting up a call schedule

A call schedule sets the hours during which the Verboice server is allowed to make calls. If no schedule is set up, the server will make calls at any time.

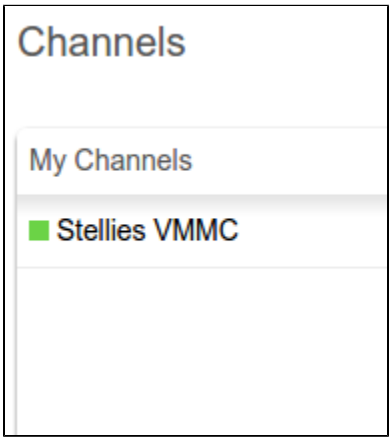
1. Select the project and click "Add Schedule".
2. Fill in the following:
    a. Name: the name for your call schedule
    b. Days: select the days you want calls to go out
    c. Time (Optional): You can enter a starting and ending time that you want calls to go out for. Calls will not be made outside of these hours. They will stay queued until the next day.
    d. Retries (Optional): This is the delays you want before a failed call retries. eg. if you fill in 1,2,3 the call will retry after one hour, again after two hours and again after three hours.

Setting up a channel

This is the Skype/Sip/Twilio channel that Verboice will use to actually place the calls to cellphone subscribers.

1. Log in to Verboice and go to the channels page: http://196.26.21.45/channels

2. Choose Create New -> Skype Channel
3. Give the channel a name.
4. Enter a username and password. A new username and password can be purchased from https://manager.skype.com/features/sip. (Login details in the kdb).
5. 'Number' and 'limit' can be left blank.
6. Save the channel, and check that the indicator turns green (see picture below).
7. If the indicator doesn't turn green after a few minutes, there is something wrong.



External services

External services are used to add new call flow elements (the little squares in the call flow) to Verboice. If you want to use any external services you have to add them under this tab.

See the official documentation on how to create external services here: https://bitbucket.org/instedd/verboice/wiki/External%20Services.md#!how-to-use-existing-external-services

The Password Callback

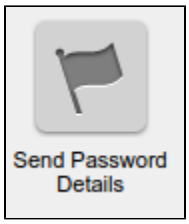
Currently I am using one external service for the Stellenbosch project, which allows Verboice to send the password entered by the user to our VMMC server.

You can see the service here: http://www.cell-life.org/vmmc/verboice-service/passwordCallback.xml or expand the source below.

```
<verboice-service>
<name>Cell Life</name>
<global-settings>
 <variable name="service_domain" display-name="Service Domain" type="string"/>
</global-settings>
<steps>
 <step name="password_details" display-name="Send Password Details" icon="flag"
type="callback"
callback-url="https://www.cell-life.org/vmmc/verboice-service/passwordCallback">
  <settings>
   <variable name="passwordEntered" display-name="Password Entered" type="string"/>
  </settings>
  <response type="none"/>
 </step>
</steps>
</verboice-service>
```

This makes available the "Send Password Details" callflow element, which can be used to send the user's password back to the VMMC server. The only configuration necessary in Verboice is adding the password variable. See "Sending the user's password"

below.



Setting up a call flow

Next you need to create a call flow. I have uploaded a sample call flow here, which you can import and play around with.

I will try and document some important notes about our current call flows and the elements used.

The branch element

This is kind of like a case statement in the call flow. The call flow can branch off into different sub-flows, depending on the value of a variable.

In our case we use the "message" variable to branch to different audio content.

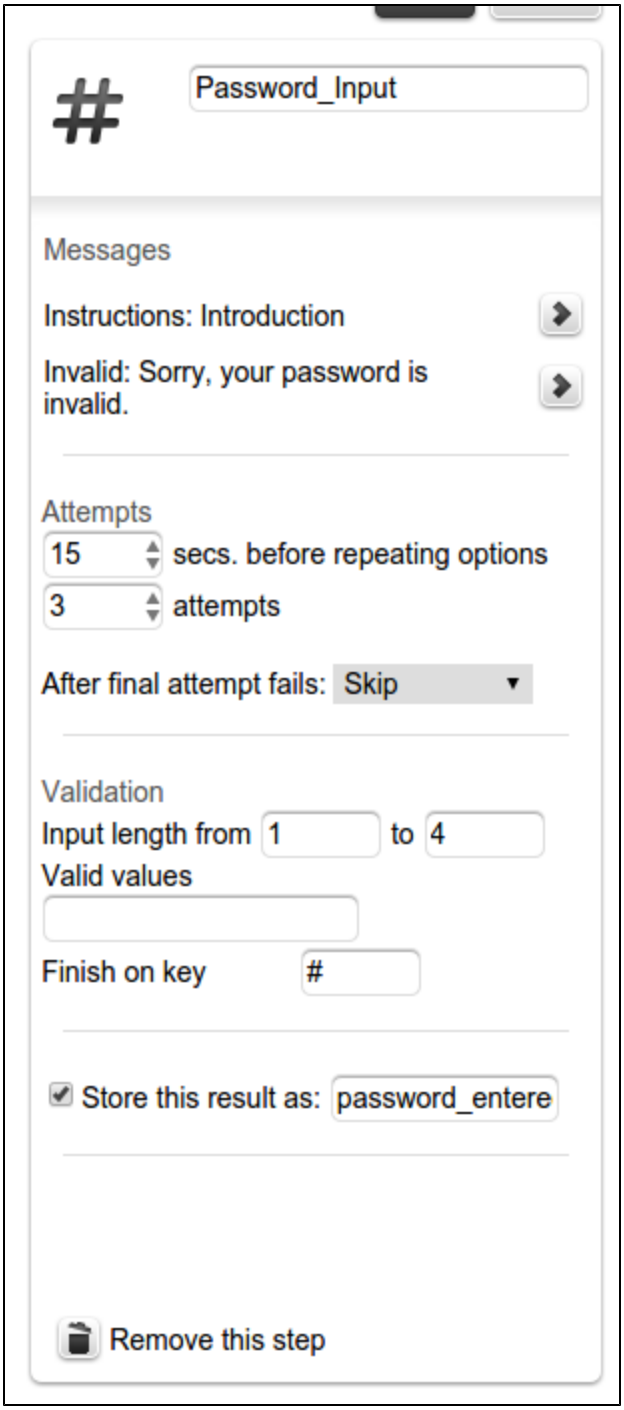
- The 'branch' function looks at the 'message' variable to determine what message to play.
- Calls are logged to Verboice via a REST URL of the following format: http://196.26.21.45/api/call?hannel=Stellies+VMMC&address=27734276045&call_flow=Main+Call+Flow&schedule=Stellies+Schedule&vars[message]=8&vars[password]=6045.
- The message number (8, in this case) will determine which message is branched to.
- All messages in an IVR/VMMC "campaign" needs to be added to the branch in order to be available for playback.

Saving the user's password entry with the 'input' element

The input function is used to get input from they user in the form of keys pressed on the cellphone.

It can be set up as below for a four digit password.

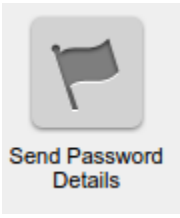
It is important to store the input (or 'result') to a variable. In this instance we are storing it to 'password entered'.



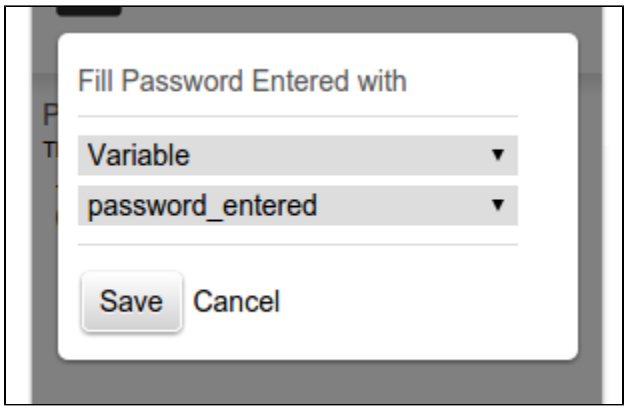
Sending the user's password entry back to the VMMC server

This can be done with the external service we created (see 'External Services' above).

If our external service has been added, we can add the call flow element 'Send Password Details'.



Then select the variable "password_entered" to be sent back to the VMMC server (see picture below).



The 'impersonate' element

This element is used to authenticate a user. It is used when multiple users share the same phone number, but with different pins. The pin is more like an ID, and not really like a password. See discussion here: https://groups.google.com/forum/#!searchin/instedd-tech/impersonate$20function/instedd-tech/ds1DG2sPl6c/Gmhtn2l9DQsJ

We didn't end up making use of this, but I am noting it in case you want to revisit it in future.

Support

This page contains information on how to support and administer our VoIP system.

- Skype
    - Skype Channel
    - Skype Credits
    - Skype Number
    - Reports
- Verboice
    - How to see if Verboice is online
    - Settings
    - Troubleshooting

Skype

Skype is a VoIP (Voice over IP) provider. We have an account with them in order to make phone calls.

You will need to have a Skype login in order to configure the Skype account. Login at https://manager.skype.com/

Read more about Skype here: Skype Manager

Skype Channel

A Skype SIP channel is used to make calls. This SIP channel has a monthly subscription and is configured via the Skype account.

Skype Credits

You need to purchase credits in the Skype account in order to make calls.

Skype Number

You can purchase a Skype number to associate with your SIP channel in order your number to appear in caller ID. If you don't have a Skype Number then all calls will be displayed as "Private number".

Reports

You can run reports from Skype in order to see the calls made.

https://manager.skype.com/reports

See Allocations (SIP channel and Skype number subscriptions) and Usage tabs (to see call logs).

Verboice

Verboice is the IVR (Interactive Voice Response) system we use. Verboice is responsible for integrating with Skype in order to place the call and performing whatever actions are defined in the call flow.

Read more:

- Our Verboice setup.
- Verboice mailing list
- Verboice documentation

How to see if Verboice is online

Go to http://196.26.21.45/ and login.

View the channels for the call flow (for Stellies it is currently http://196.26.21.45/channels/10) and then you can see "No errors" in the right most box.

Settings

Once you are viewing a Channel in Verboice, click the Settings tab (currently it is http://196.26.21.45/channels/10/edit) and then modify the SIP username and password..

Troubleshooting

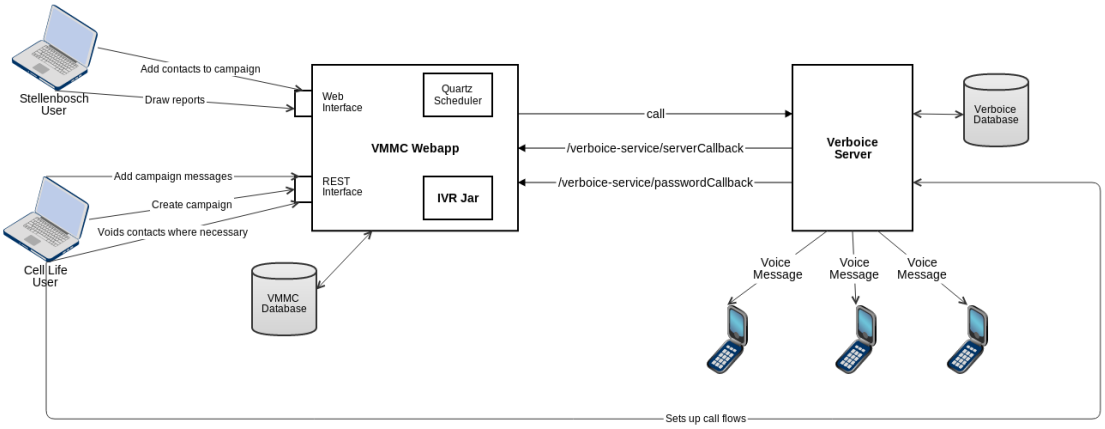
Login to Riaan (196.26.21.45) and view the log file at /var/log/verboice

System Design

System Diagram

The diagram below shows how the voice messaging system is set up. Some things to note:

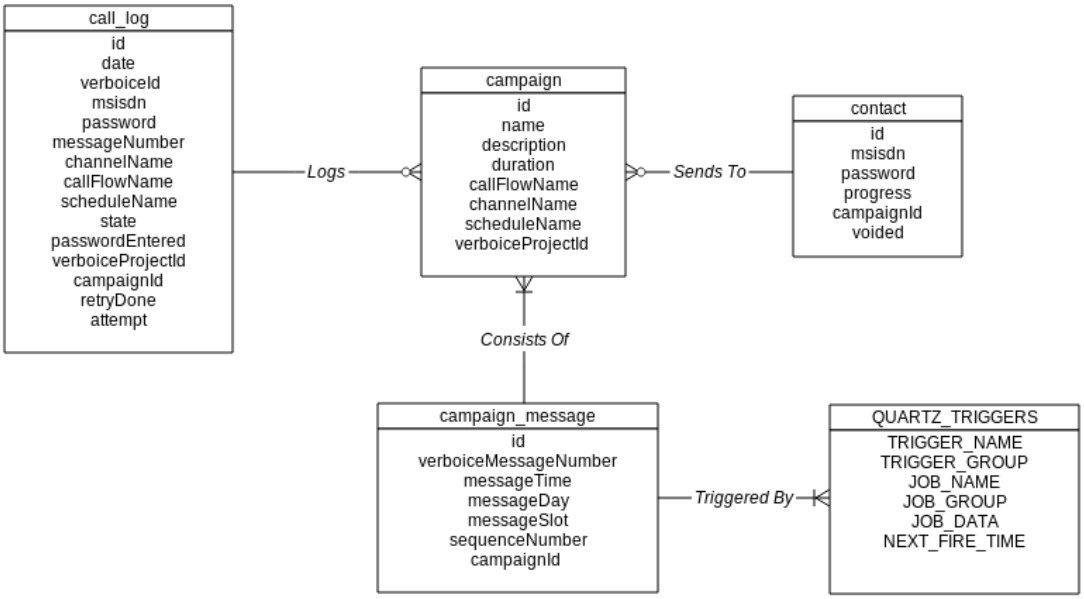
- The VMMC webapp logs campaign calls to the Verboice server, which then places the calls to the end user.
- Once the call is logged, the Verboice server sends the status of the call back to the VMMC app every time the status changes.
- The Verboice server also sends the password entered by the user to the VMMC app, should they enter something.
- The VMMC webapp relies on the IVR core jar, a separate Cell Life project.
- The VMMC web interface can be accessed at https://www.cell-life.org/vmmc.
- The rest interface is documented HERE.



Database (ERD) Diagram

There are four main entities in the voice messaging system, as shown in the diagram below.

Each instance of an IVR app can contain multiple instances of a campaign.



Quartz: Message scheduling and other quartz jobs.

The message scheduling is handled by the quartz scheduler component: http://quartz-scheduler.org/documentation/quartz-1.x/configuration/. (We are using version 1.8.6 at the moment).

When a new message is added to a campaign, a corresponding quartz "trigger" is created for the message time. eg. A message scheduled for 10:00am will create a quartz trigger for 10:00, UNLESS a quartz trigger is already scheduled for that time. The quartz framework thus takes care of sending campaign messages at the correct time.

There is also a quartz job for finding failed calls and retrying them, every hour. Various other "background" jobs may be created as needed.

Releasing new versions of VMMC and IVR

Since VMMC uses IVR as it's core functionality, I have been working on both simultaneously. That means that when the time comes to make a release, I have to first release IVR and then release VMMC.

I am documenting the steps to do this:

1. First make a new release of IVR (https://www.cell-life.org/jenkins/job/ivr/)
2. Note the version of the new release eg. 1.0.4. (The jar should also be uploaded here: https://www.cell-life.org/nexus/content/repositories/releases/org/celllife/ivr/celllife-ivr-core/)
3. Edit the parent pom.xml of the vmmc project and set the version of celllife-ivr-core to the version in step 2.

Where to set the new version:

```
<!-- Cell Life Ivr -->
            <dependency>
                <groupId>org.celllife.ivr</groupId>
                <artifactId>celllife-ivr-core</artifactId>
                <version>1.0.4</version>
            </dependency>
```

4. Commit this pom.xml file.
5. One jenkins has registered the change, make a release of VMMC (https://www.cell-life.org/jenkins/job/vmmc/).