Square O

CC Callback for UCCE

##### Document history

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Person | Version no. | Description |
| 2023-05-20 | Marek Słomiński | 1.0 | First version of the document |
| 2023-05-21 | David Macias | 1.1 | Review |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Table of Contents

[Table of Contents 1](#_Toc135456258)

[Introduction 2](#_Toc135456259)

[Solution Overview 2](#_Toc135456260)

[Solution HA 2](#_Toc135456261)

[Callback dataflow 3](#_Toc135456262)

[Callback set by CVP VXML Application 3](#_Toc135456263)

[Callback set by Finesse Gadget 4](#_Toc135456264)

[Admin portal 5](#_Toc135456265)

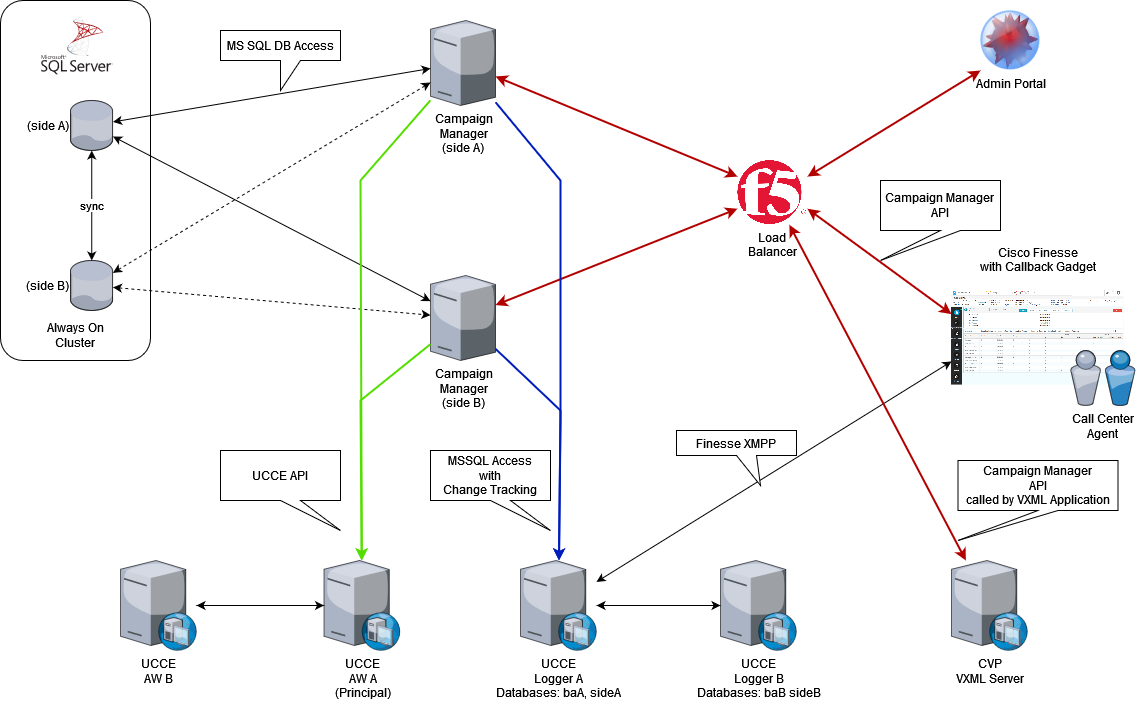
[CUIC Reports 6](#_Toc135456266)

# Introduction

Below document describes the solution of custom callback functionality that can be added to UCCE system. The solution is created at top of the UCCE dialer functionality for making outbound calls to the customers and additional 3rd party elements (delivered as a part of solution).

# Solution Overview

The below diagram presents the solution high level overview (in HA mode).



The solution is built with the following components:

* Campaign Manager service – component needed to deliver API and to manage scheduled callbacks.
* Database – MS SQL working in Always On mode – this option will be much better than using standard replication mechanisms as it is free from version 2016 and uses less resources.
* Load balancer – to distribute the traffic across Campaign Manager servers to provide application HA.
* Admin portal – written in Angular JS – portal that allows to manage admin settings of the Campaign Manager. At this point the portal will allow to:
  + Manage access to admin portal.
  + Configure Timeslots for callbacks.
* CUIC reports – reports that will show how callbacks are working.

## Solution HA

The solution will be delivered as HA solution. To achieve this goal the solution will use the following mechanisms:

1. MS SQL Database – database will be deployed as Always On Cluster which means that only one database is available during regular operation. Secondary DB is offline but replicates all the data that arrives at primary database.
2. Campaign Manager – the system will have 2 working Campaign Managers. Those servers will:
   1. Always deliver APIs for Finesse Gadget and Admin portal
   2. Only one side will communicate with UCCE – one of the Campaign Mangers will have master role that will allow it to communicate with UCCE. If Master Campaign Manager will go down, the master role will be passed to backup server.
3. Load balancer – will be used to distribute CVP VXML Application, Finesse Gadget and Admin Portal request to working Campaign Manager servers.

## Callback dataflow

The solution will support 2 types of callback flows:

* Callback set by IVR (CVP Application) – this scenario will be used to schedule client callbacks presented by IVR in case no agents are available, or the estimate wait time is very high and the customer decides to wait on the phone.
* Callback set by Finesse Gadget – this scenario will be used to schedule client callbacks when customer is already connected to agent but during conversation client will require to end conversion and reschedule it for the future.

The below 2 chapters describe in detail both mentioned scenarios.

### Callback set by CVP VXML Application

1. Customer calls the main IVR number.
2. IVR delivers customer treatment. At some point IVR checks agent availability and estimated wait time. No agents are available, and the wait time is higher than expected.
3. IVR reaching out to Campaign Manager to validate if it’s online and if there are callback slots available in the next X days.
4. Assuming yes to the above, IVR proposes the callback to customer.
5. Customer decides to have a callback at specific time.
6. IVR calls the schedule callback API and passes the following info to Campaign Manager:
   1. Customer phone number.
   2. Callback date and time.
   3. Skill group name.
   4. Customer ID if known or empty to autogenerate it for anonymous clients.
   5. Callback campaign name.
   6. Additional call data.
7. When request is caught by Campaign Manager, all entered data are validated and checked if system will be able to make callback at the following time (check against Campaign Manager configuration). The checks that must be implemented:

* Validity of the phone number.
* Validity of the set date
  + Check if entered date is not a past date.
  + Check if skill is opened during the callback time.
  + Check if date is in the range of global rules set on the Campaign Manager.
  + Check against holidays.
  + Campaign Manager validates the request checking the configured timeslots and number of calls already set for callback.

If all conditions are met, call back is written to database and waits for dial out.

If conditions are not met, ex: the limits are already reached, info is passed back to IVR with the information that callback should be rescheduled.

1. One of the key services deployed on the Campaign Manager is Scheduler. During Callback dataflow it is responsible to pull scheduled callbacks and transfer them to dialer. The callback checks are performed on scheduled period (ex. 1 min). Once the callback time conditions are met – information is pulled from Campaign Manager Database.
2. Based on the pulled data Scheduler passes this data to Web Client
3. Web client makes a UCCE API Rest call to import new contact to callback campaign.
4. Dialer begins to dial. Dialing process is a standard Dialer Outbound IVR dialing process where dialer exchanges information between Voice Gateway or Cube.
5. If the customer answers the call, Dialer will redirect it to ICM Script.
6. IVR Script triggers CVP script which communicates with Campaign Manager and pulls the customer data (Call Variables) based on Account Number.
7. Web server checks the Campaign Manager Database and pulls all information stored in the internal Database (agent name, skill). Information is returned to script.

At this point 2 scenarios may happen:

* Call can be queued based on the information about agent and skill (if agent name is available then system will try to queue the call to agent – personal callback).
* Call requires further processing (ex. gather additional data about client from internal systems).

1. Based on the data IVR delivers call to selected skill group.
2. Callback set by IVR flow ends.

### Callback set by Finesse Gadget

1. Agent logs to Finesse application.
2. At the beginning Finesse loads from the Finesse server basic list of setting and all gadgets defined for an agent. One of those gadgets is Callback Gadget. When UI is loaded, agent change status to ready.

During the day agent receives a customer call (originated by dialer via outbound IVR campaign). During conversation customer requests callback. Agent goes to Callback Gadget and enters parameters:

* Customer phone number – number will be auto populated but if customer wants to change it agent will have ability to enter new number.
* Callback date and time.

1. Callback Gadget communicates with Campaign Manager via Callback REST API passing the following information:

* Customer phone number.
* Callback date and time.
* Skill name – for regular (skill based) callbacks.
* Customer First and Last Name.
* Callback campaign name.

1. When request is caught by Web server, all entered data are validated and checked if system will be able to make callback at the following time (check against Campaign Manager configuration). The checks that must be implemented:

* Validity of the phone number.
* Validity of the set date
  + Check if entered date is not a past date.
  + Check if skill is opened during the callback time.
  + Check if date is in the range of global rules set on the Campaign Manager.
  + Check against holidays.
  + Campaign Manager validates the request checking the configured timeslots and number of calls already set for callback.

If all conditions are met, call back is written to database and waits for redial.

If conditions are not met, ex: the limits are already reached, info is passed back to agent to with the information that callback should be rescheduled or info that something is wrong with the callback.

1. One of the key services deployed on the Campaign Manager is scheduler. During Callback dataflow it is responsible to pull scheduled callbacks and transfer them to dialer. The callback checks are performed on scheduled period (ex. 1 min). Once the callback time conditions are met – information is pulled from Campaign Manager Database.
2. Based on the pulled data Scheduler passes this data to Web Client
3. Web client makes a UCCE API Rest call to import new contact to callback campaign.
4. Dialer begins to dial. Dialing process is a standard Dialer Outbound IVR dialing process where dialer exchanges information between Voice Gateway or Cube.
5. If the customer answers the call, Dialer will redirect it to standard ICM Script.
6. IVR Script triggers CVP script which communicates with Campaign Manager and pulls the customer data (Call Variables) based on Account Number.
7. Web server checks the Campaign Manager Database and pulls all information stored in the internal Database (agent name, skill). Information is returned to script.

At this point 2 scenarios may happen:

* Call can be queued based on the information about agent and skill (if agent name is available then system will try to queue the call to agent – personal callback).
* Call requires further processing (ex. gather additional data about client from internal systems).

1. At the end call is queued to agent or a skill group. When agent becomes, available call is transferred to agent.
2. Callback set by Finesse Gadget flow ends.

# Admin portal

One of the elements of the solution is an Admin portal that will allow business administrators to manage the behavior of Campaign Manager. For the callback functionality the admin portal will have the following features:

* Login page – all users that have access to portal will need to authorize.
* User management – portal page that allows to assign access to users.
* Outbound campaign management – all callback campaigns that will be created by Campaign Manager via API. This means that campaign cannot be managed by ICM Configuration Manager. Admins to change parameters (like power dialing) will need to have access dedicated UI to change campaign parameters.
* Timeslots page – portal pages that allow to configure timeslots functionality. The page will have the following sub-pages:
  + Daily timeslot schedule – place where administrators can create a timeslots template for a day (assigned allowed callbacks within 30 minute interval)
  + Weekly schedules – place where administrators can assign daily timeslots schedule and create a timeslot template schedule for a week.
  + Skill timeslots – place where administrators can assign prepared weekly templates to UCCE skills.

# API Endpoints Available to IVR

* Available slots for skill. IVR passes skill name. Campaign Manager returns the count of available slots for the next 3 business days.
* Available slots for skill and dates. IVR passes skill name and up to 3 dates.

# CUIC Reports

As a part of the solution CUIC Reports will be delivered. The delivered reports will present:

* Real-time view of scheduled callbacks
* Callbacks scheduled.
* Callbacks made.
* Callbacks result.
* Report that shows how many times was a callback requested (this is not the same as scheduled) a callback can be requested, but no slots might be available, or the person doesn’t complete the callback process.