ADMINISTRATION AND CONFIGURATION (AVAYA AND CRM)

CDC SOFTWARE LLC

Version 1.5.1
November 17, 2016

The information presented in this Administration and Configuration guide Documentation was prepared by CDC Software. The sole purpose of this Documentation is to provide the recipient with information about Administration and Configuration. The recipient is encouraged to visit the Company's website at www.cdcsoftware.com for additional information.

While the Company believes that the information contained herein is accurate, it expressly disclaims any and all liability for representations or warranties, expressed or implied, contained in, or for omissions from, this Documentation or any other written or oral communication transmitted or made available.

As a condition to our furnishing you with this Documentation and any additional information we may provide, you agree to treat all information that is contained in the Documentation and all such additional information as confidential unless and until such information has been made generally available to the public or unless the Company consents in writing to any disclosure. This Documentation may not, in whole or part, be reproduced or distributed to others. You may disclose such information to your agents or directors, officers, employees or advisors if such representatives are informed by you of the confidential nature of such information and directed by you and agree to treat such information confidentially. Nothing herein shall limit or impair your right to disclose information contained in the Documentation if legally required to do so in any judicial, administrative or governmental proceeding, subject to your prior notification to the Company.

FOR FURTHER INFORMATION, CONTACT:

Support
Helpdesk, CDC Software LLC
240 Market Street, Suite 1B
Bloomsburg, PA 17815-1727

Office: 570.309.6827 Support Option
Email: support@cdc-tek.com
Online: http://support.cdc-tek.com

Table of Contents

1 CDC Platform Overview

The CDC Platform Server is designed to provide a highly extensible platform by utilizing configurable “Connectors”. Each “Connector” provides specific functionality enabling rapid and flexible integration. The CDC Platform server host connectors for a variety of telephony systems and for systems wishing to send/receive data from these telephony systems. The telephony connector is responsible for managing the connection to the telephony environment, standardizing the data and events generated by the individual systems, and publishing the standardized events and data to the CDC Platform Engine. Any other connector can then subscribe to events published by the telephony connectors and multiple connectors can act on a single published event. A single connector can both publish and subscribe to events while published events support both synchronous and asynchronous modes.

The CDC Platform engine supports filtering of events, data translation, and custom data mapping at both the point of ingress to the CDC Platform engine (event published) and at egress (event received by subscriber).

1.1 Description

The CDC Platform integrates your 3rd party telephony system to your Cloud based CRM environment. The CDC Platform supports searching for, displaying, and creating CRM items based on the call data associated with a call. Leverage the information collected within your IVR, Voice Portal, and ACD to "Screen Pop" the agent's CRM session with new or existing CRM users and tickets.

The communications enabled by the CDC Platform is bi-directional and provides the ability to update call data within your 3rd party telephony system with Zendesk data. This enables:

- CRM user and ticket data located or created during the existing call to be made available to any other agents that may handle the current call via transfer or conference.
- Enables access to CRM data that was located, displayed, or created during the call to your other contact center applications (reporting, call/screen recording applications, etc.) For example, you can now associate your call recording or screen recording to the call and to the CRM ticket automatically created by the call.

In summary the CDC Platform Server, through its connectors, allows the contact center to utilize data from their telephony systems in any system for which a connector is available and vice versa.

1.2 Implementation

CDC Platform presently supports the following telephony systems:

1. Avaya Aura Red (also called ACM or Avaya Communications Manager). Integration via TSAPI on AES and CCT via AACC
2. Avaya Aura Blue (also called CS1K). Integration via CCT.
3. Avaya IP Office. Integration via CTI Link Pro (TAPI).
4. Cisco UCCX (Integration via Finesse, CAD, or UCCXCTI)
5. Cisco UCCE (Integration via Finesse, CTIOS, or CTIServer)
6. Cisco CUCM (Integration via TAPI)
7. SwitchVOX (Premise or Cloud) (Integration via SwitchVOX API over HTTP)
8. Asterisk (Integration via AMI)
9. Vonage Business (Cloud) (Integration via Vonage API over HTTP)
10. InContact (Cloud) (Integration via InContact API over HTTP)
11. MiTel (Integration via Open Integration Gateway[OIG])
12. BroadSoft and BroadSoft based systems (Integration via Xtended Service Interface over HTTP)
13. Alcatel-Lucent OMNI Systems (Integration via TSAPI or TAPI. Requires Alcatel Premium TSAPI/TAPI server)
14. Interactive Intelligence (Integration via ICWS over HTTP)
15. TAPI compliant pbxs (AllWorx, some Mitel, many other pbxs) (Integration via TAPI)
16. Generic HTTP support. A PBX that can generate HTTP notifications for call events can be directly supported.

A CDC Platform implementation with CRM connector, due to the highly customized nature of most customer's telephony environments, requires an initial discovery session with customer to gather information about the customer's existing 3rd party telephony system. Information gathered during discovery session includes:

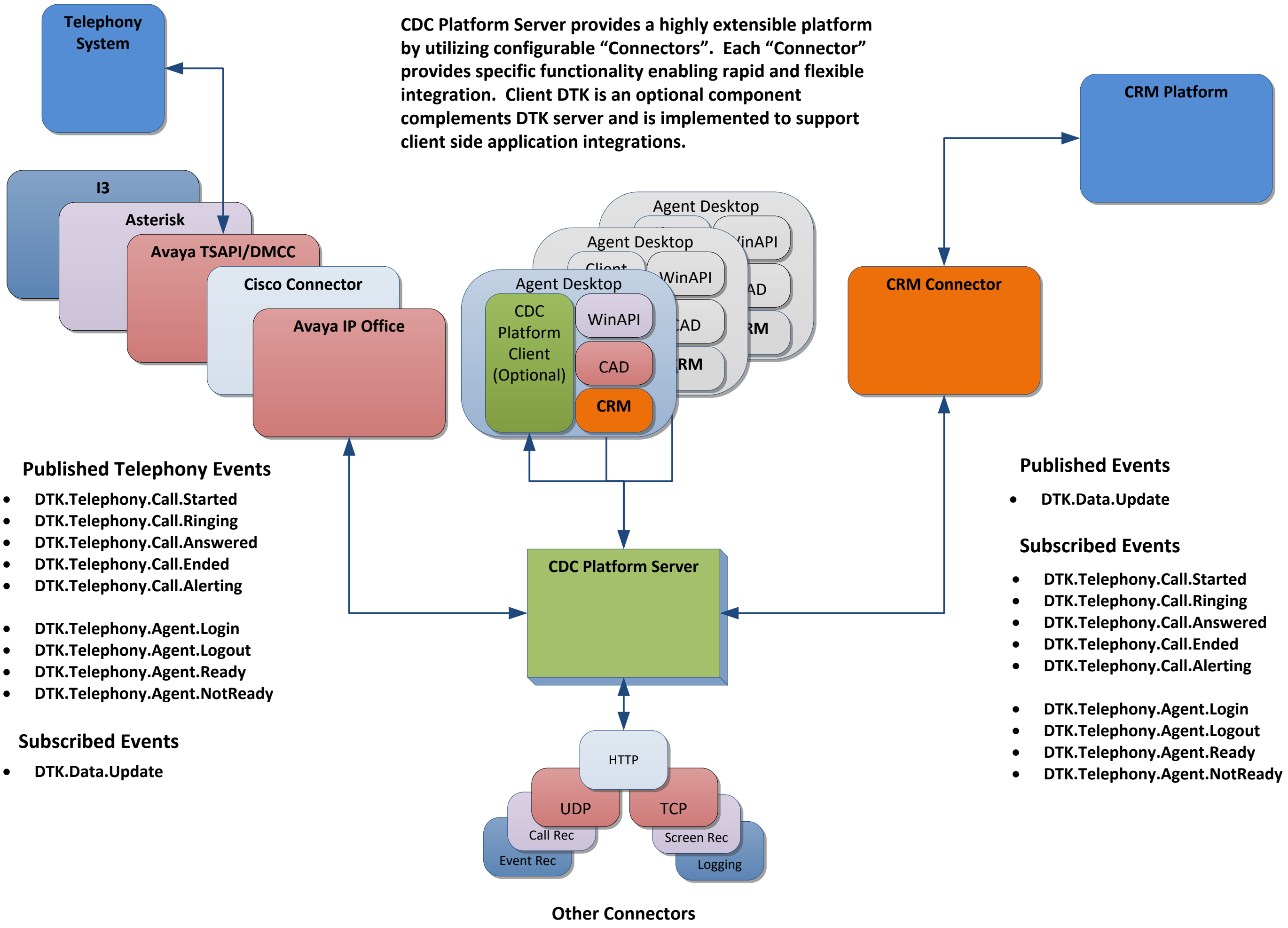
- Call data available in telephony system. E.g. what data is being collected about caller and where is it being stored?
- What behavior should occur in CRM session and when should it occur? E.g. Display a user and/or create a ticket?
- What, if any, data lookups are required to other systems (Customer or order systems for example)?

Once these details are gathered the CDC Platform will be configured by CDC Software technicians with the rules to enable the desired behavior and deployed within the customer's environment. Most integrations for "screen pop" capability are a server only solution so require no software to be deployed on the agent's desktop greatly simplifying deployment within the enterprise.

2 Example Call Flow and Architectures

The CDC Platform is a middleware solution that sits between the telephony system and the CRM system. The platform monitors the call activity, in real-time, and pushes data to the CRM user through the CRM’s cloud based Restful API over HTTPS. In short, phone calls drive the display of ticket data to the agent in their browser. Please see below figures for call flow and architecture diagrams.

CDC Platform and Connector Architecture



CDC Platform Server provides a highly extensible platform by utilizing configurable “Connectors”. Each “Connector” provides specific functionality enabling rapid and flexible integration. Client DTK is an optional component complements DTK server and is implemented to support client side application integrations.

Published Telephony Events

- DTK.Telephony.Call.Started
- DTK.Telephony.Call.Ringing
- DTK.Telephony.Call.Answered
- DTK.Telephony.Call.Ended
- DTK.Telephony.Call.Alerting

- DTK.Telephony.Agent.Login
- DTK.Telephony.Agent.Logout
- DTK.Telephony.Agent.Ready
- DTK.Telephony.Agent.NotReady

Subscribed Events

- DTK.Data.Update

Published Events

- DTK.Data.Update

Subscribed Events

- DTK.Telephony.Call.Started
- DTK.Telephony.Call.Ringing
- DTK.Telephony.Call.Answered
- DTK.Telephony.Call.Ended
- DTK.Telephony.Call.Alerting

- DTK.Telephony.Agent.Login
- DTK.Telephony.Agent.Logout
- DTK.Telephony.Agent.Ready
- DTK.Telephony.Agent.NotReady

2.1 Avaya AES

Figure 1 CDC Platform Call Flow

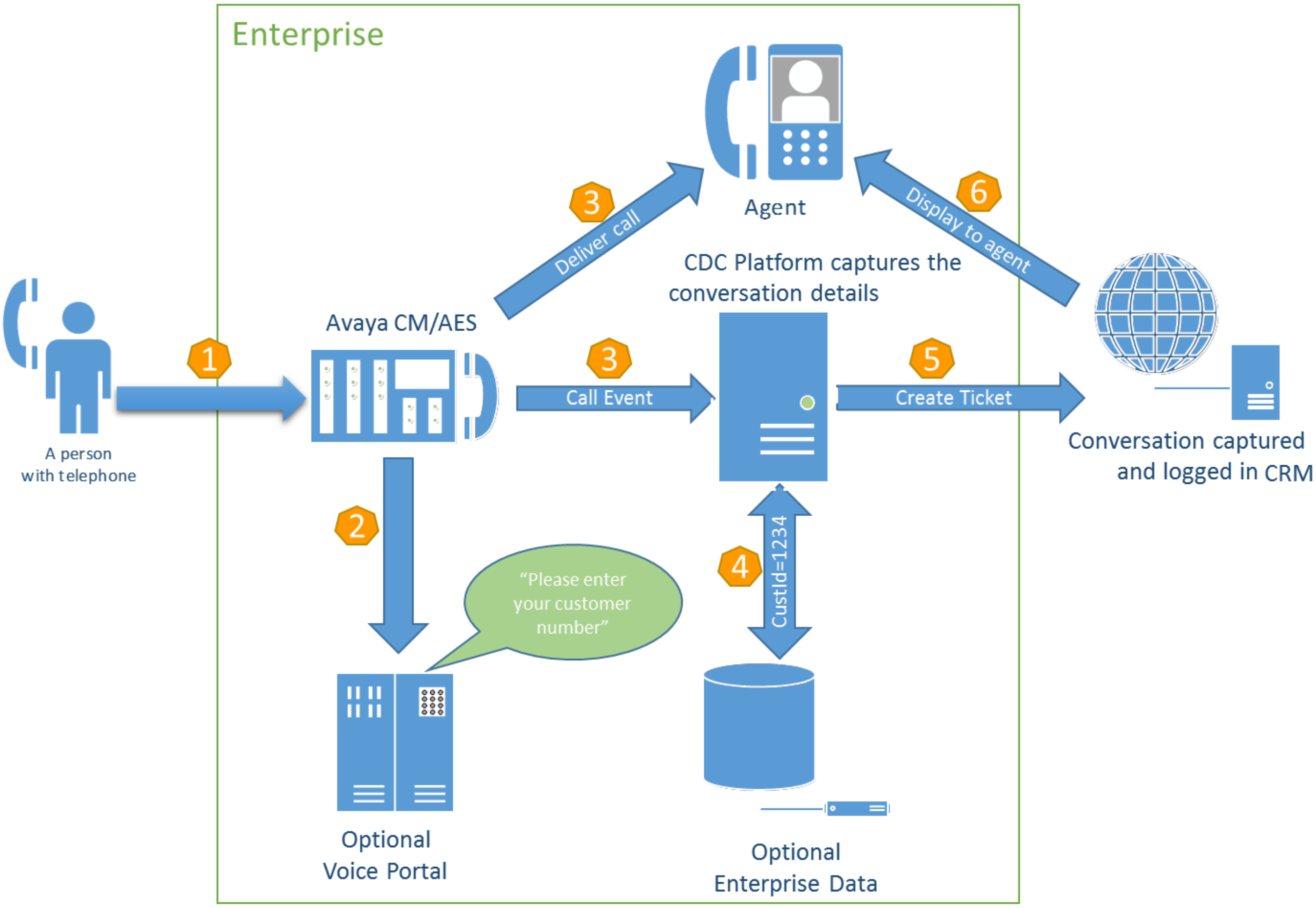
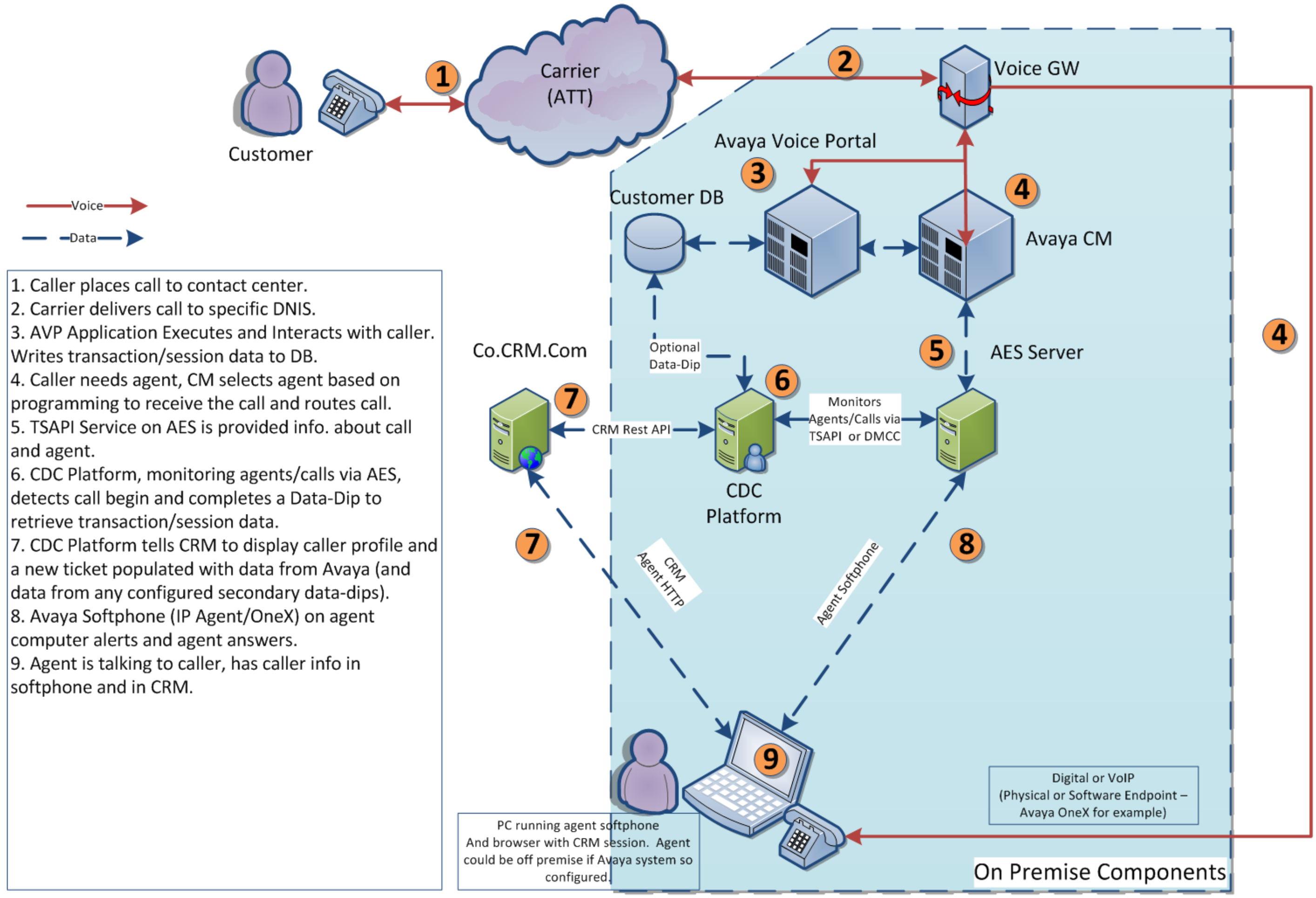


Figure 2 CDC Platform Physical Architecture


Customer
1
Carrier (ATT)
2
Voice GW
Avaya Voice Portal
3
4
Customer DB
Avaya CM
Voice
Data
Co.CRM.Com
Optional Data-Dip
5
AES Server
4
6
7
CRM Rest API
Monitors Agents/Calls via TSAPI or DMCC
CDC Platform
7
CRM Agent HTTP
Agent Softphone
8
9
Digital or VoIP (Physical or Software Endpoint – Avaya OneX for example)
PC running agent softphone And browser with CRM session. Agent could be off premise if Avaya system so configured.
On Premise Components

1. Caller places call to contact center.
2. Carrier delivers call to specific DNIS.
3. AVP Application Executes and Interacts with caller. Writes transaction/session data to DB.
4. Caller needs agent, CM selects agent based on programming to receive the call and routes call.
5. TSAPI Service on AES is provided info. about call and agent.
6. CDC Platform, monitoring agents/calls via AES, detects call begin and completes a Data-Dip to retrieve transaction/session data.
7. CDC Platform tells CRM to display caller profile and a new ticket populated with data from Avaya (and data from any configured secondary data-dips).
8. Avaya Softphone (IP Agent/OneX) on agent computer alerts and agent answers.
9. Agent is talking to caller, has caller info in softphone and in CRM.

Physical Avaya / AES

2.2 Avaya IP Office

Figure 3 CDC Platform Call Flow

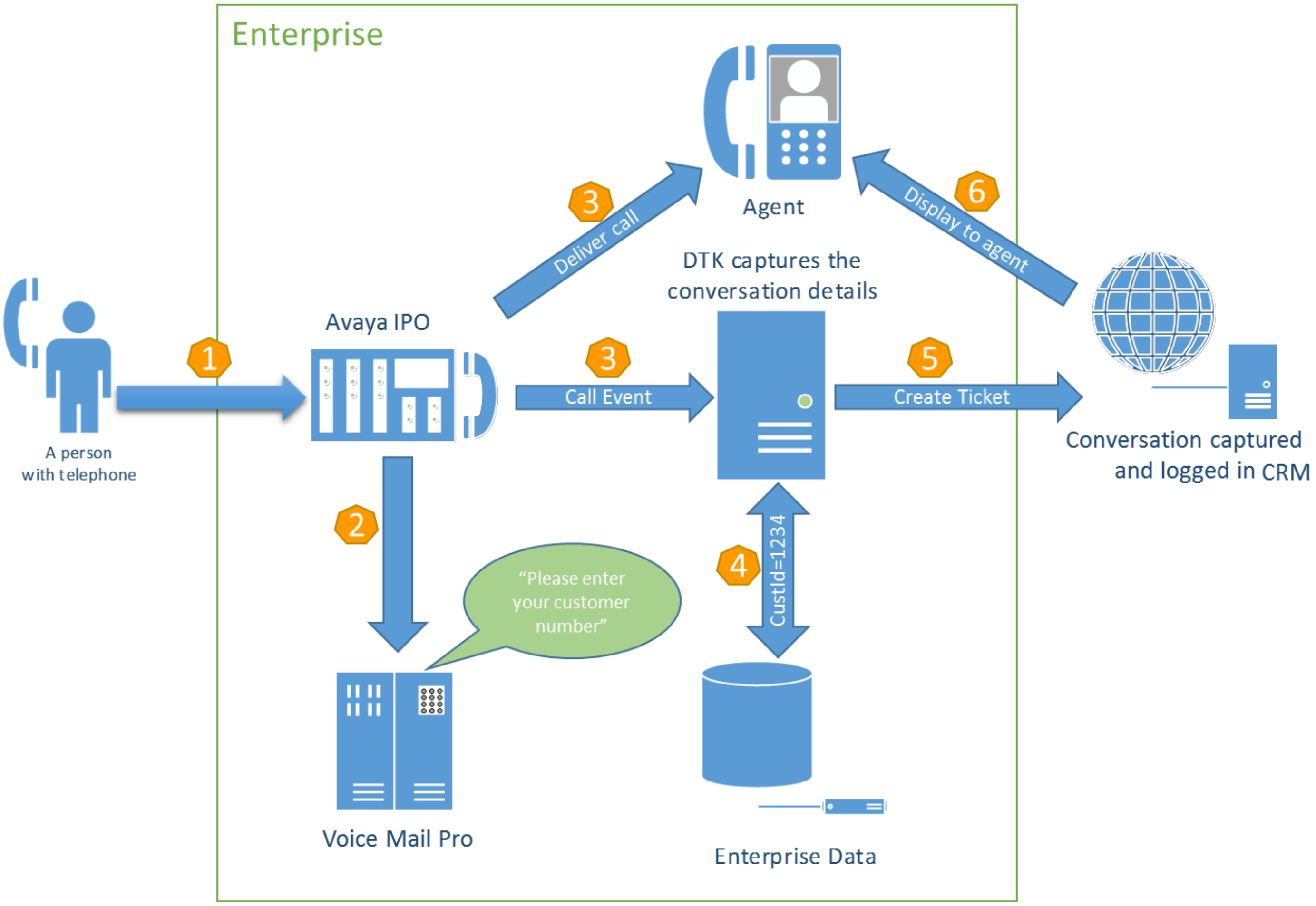
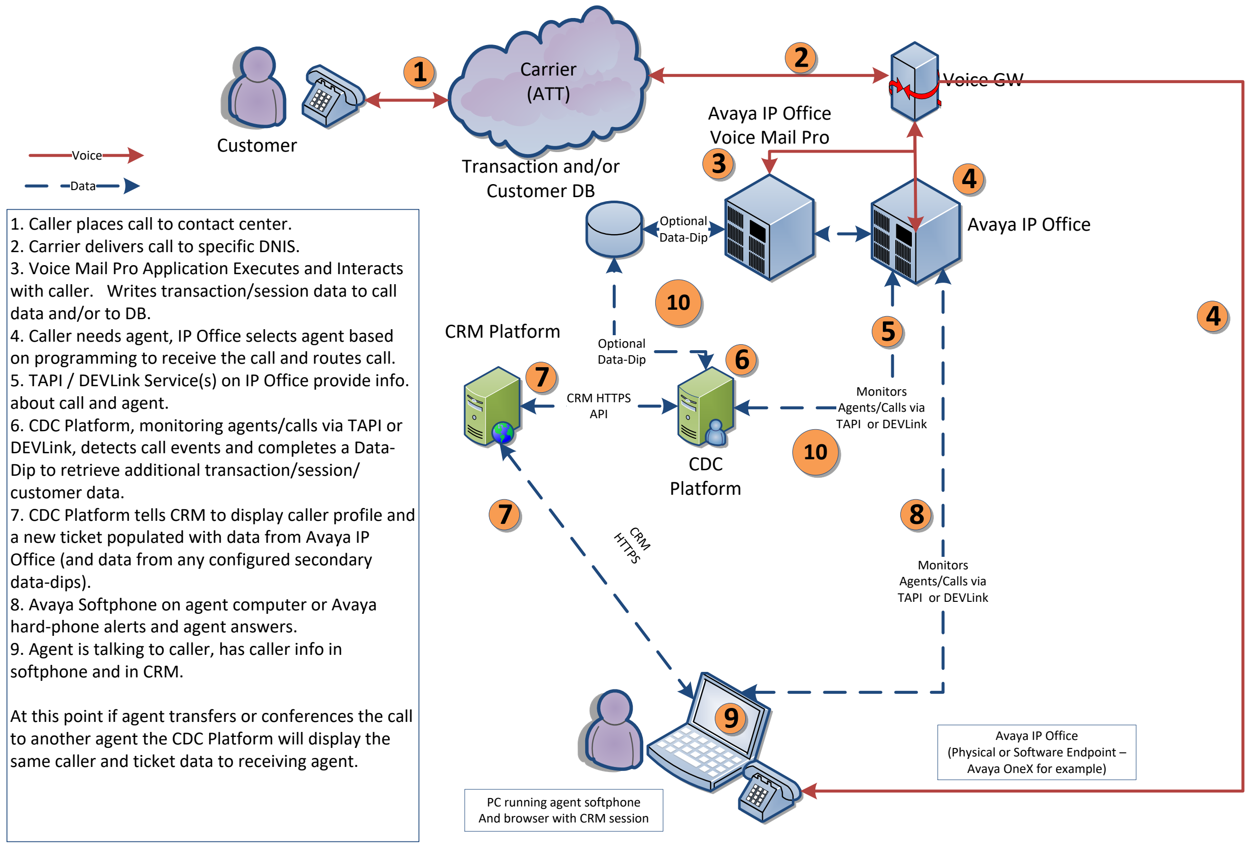


Figure 4 CDC Platform Physical Architecture



1. Caller places call to contact center.
2. Carrier delivers call to specific DNIS.
3. Voice Mail Pro Application Executes and Interacts with caller. Writes transaction/session data to call data and/or to DB.
4. Caller needs agent, IP Office selects agent based on programming to receive the call and routes call.
5. TAPI / DEVLink Service(s) on IP Office provide info. about call and agent.
6. CDC Platform, monitoring agents/calls via TAPI or DEVLink, detects call events and completes a Data-Dip to retrieve additional transaction/session/ customer data.
7. CDC Platform tells CRM to display caller profile and a new ticket populated with data from Avaya IP Office (and data from any configured secondary data-dips).
8. Avaya Softphone on agent computer or Avaya hard-phone alerts and agent answers.
9. Agent is talking to caller, has caller info in softphone and in CRM.

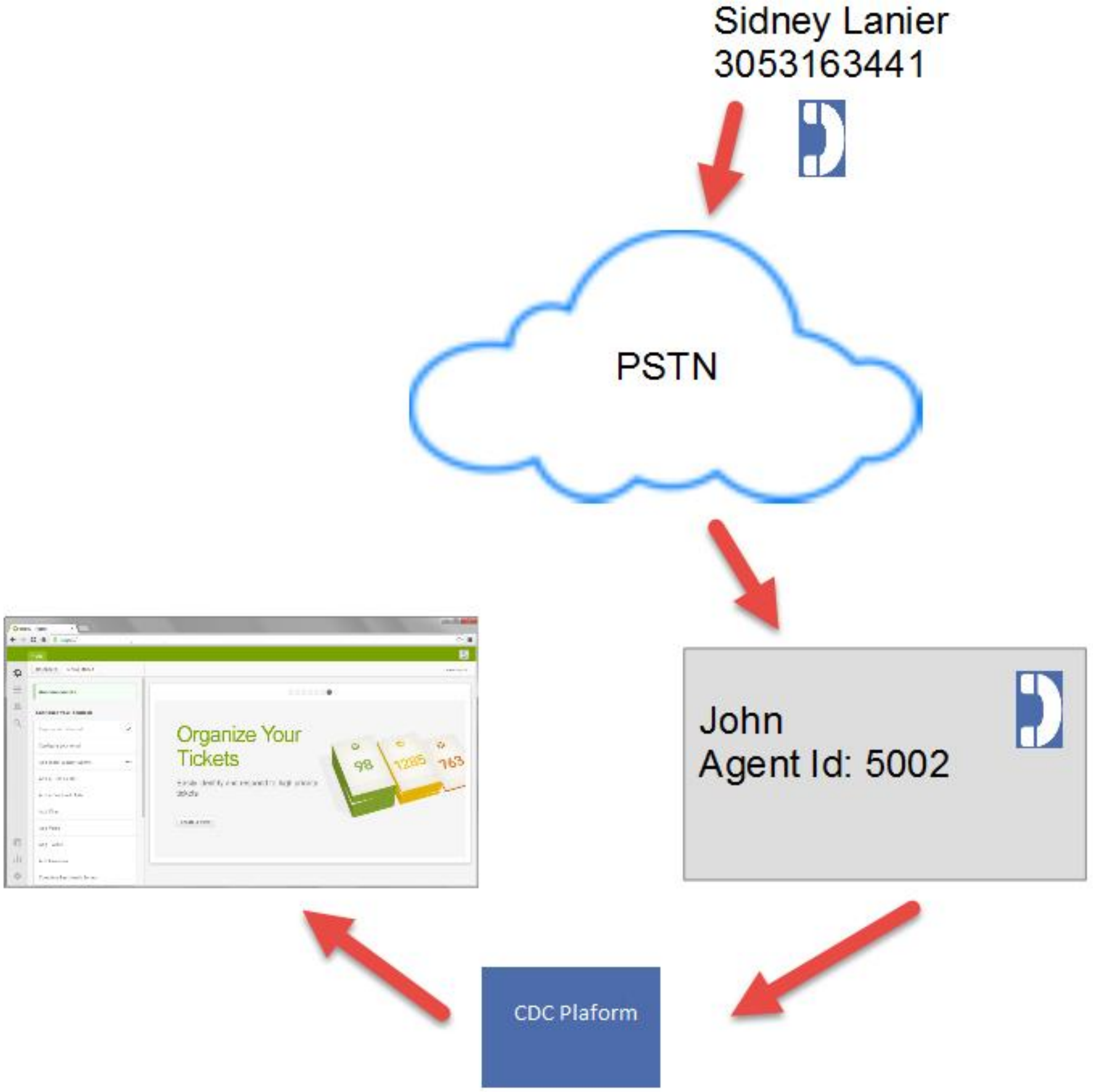
At this point if agent transfers or conferences the call to another agent the CDC Platform will display the same caller and ticket data to receiving agent.

Physical Avaya IP Office

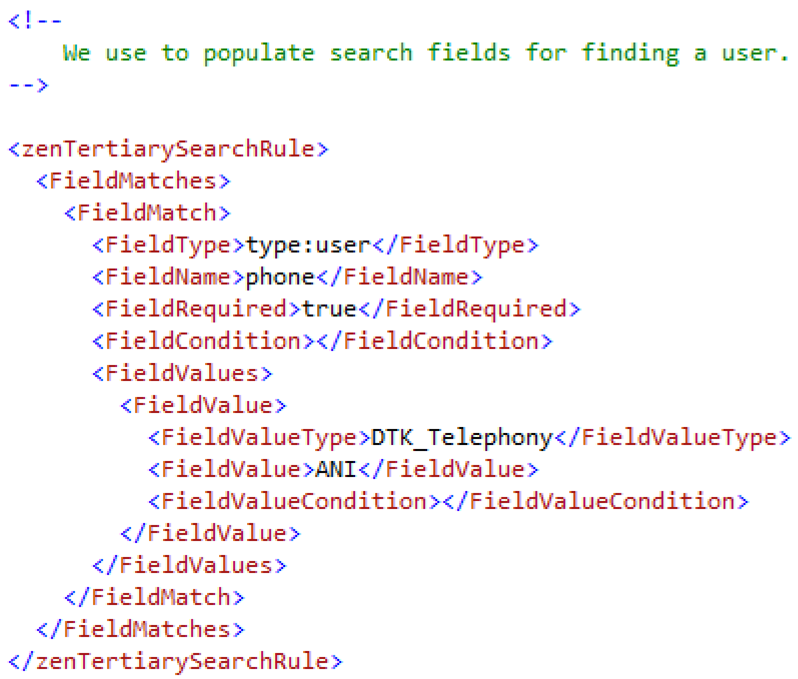
3 Automatic Number Identification (ANI) Example

These example images and configurations will show how ANI (phone number of the caller) can be used to locate a Zendesk user and display that user to the Agent receiving the call.

Each CDC Platform Telephony call event received by CRM Connector contains the ANI or telephone number of the caller. When the example end-user shown calls your organization, the event received by CDC Platform will contain a data element named ANI that contains the caller’s phone number. This in turn is used to locate the CRM user associated with this phone number and display that user to the agent.



The below search rule illustrates how the ANI example is configured. In short it states: “Search the CRM user field named phone for the value in the telephony’s events ANI field”.



```
<!--
    We use to populate search fields for finding a user.
-->

<zenTertiarySearchRule>
  <FieldMatches>
    <FieldMatch>
      <FieldType>type:user</FieldType>
      <FieldName>phone</FieldName>
      <FieldRequired>true</FieldRequired>
      <FieldCondition></FieldCondition>
      <FieldValues>
        <FieldValue>
          <FieldValueType>DTK_Telephony</FieldValueType>
          <FieldValue>ANI</FieldValue>
          <FieldValueCondition></FieldValueCondition>
        </FieldValue>
      </FieldValues>
    </FieldMatch>
  </FieldMatches>
</zenTertiarySearchRule>
```

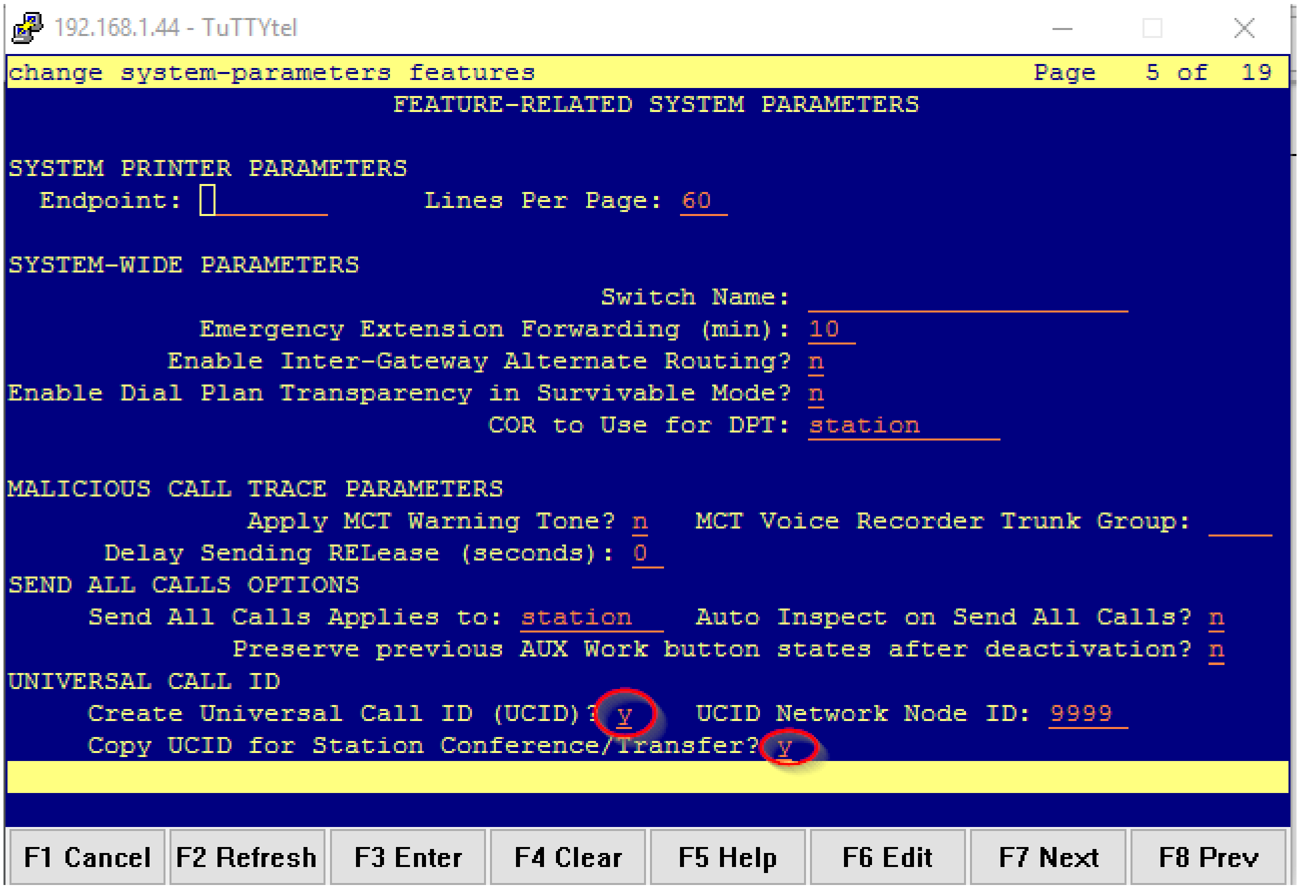
4 Telephony Connector Configuration

The telephony connectors must be configured to communicate to the telephony system and for the events required for the customer specific configuration which is based on the feature set implemented and customer requirements.

4.1 Avaya AES

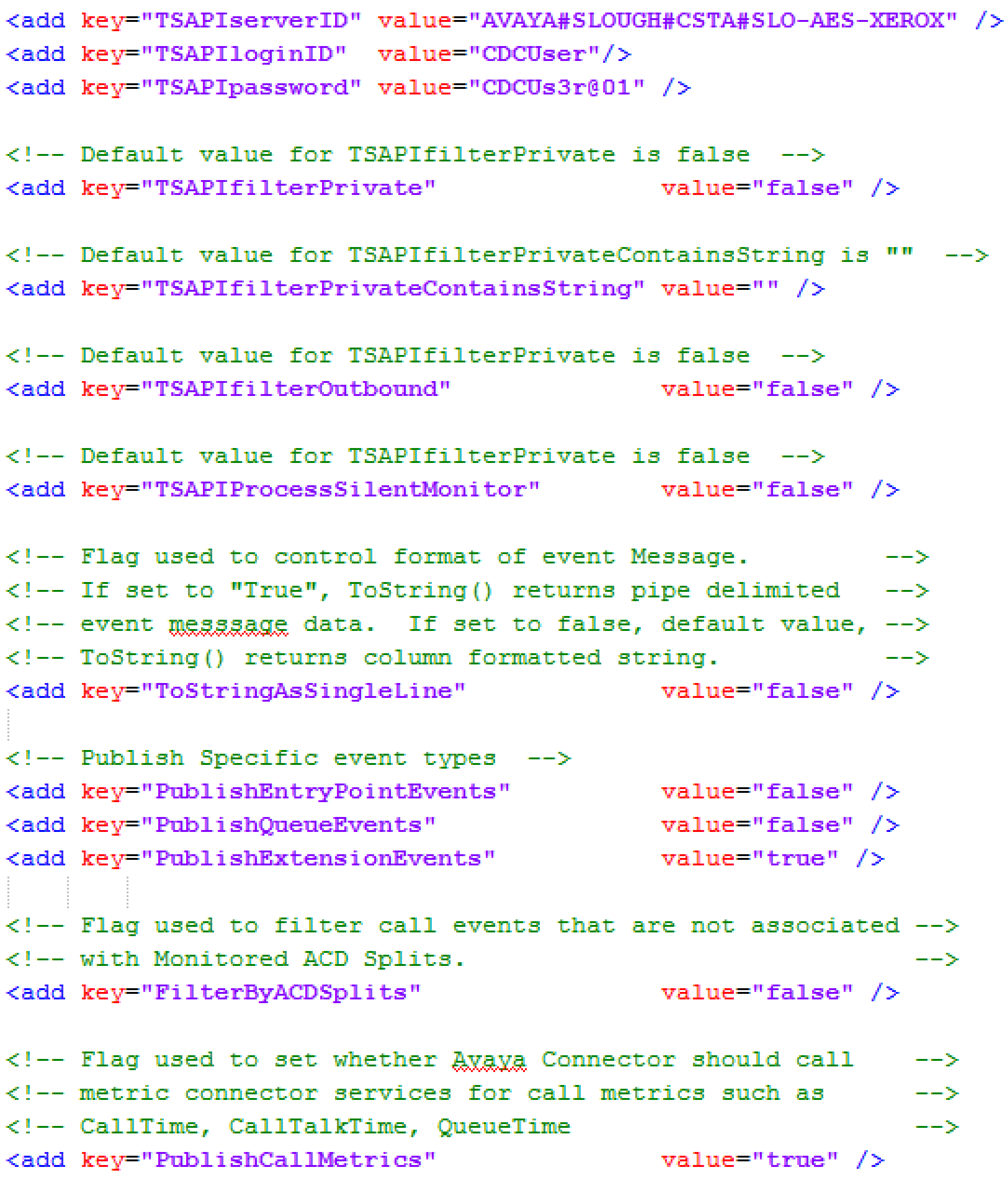
The CDC Platform connector for Avaya AES utilizes the Avaya TSAPI client. The TSAPI client must be installed prior to CDC Platform configuration using the Avaya TSAPI Client installer.

The CDC Platform requires the use of the UCID (Universal Call Identifier) to be enabled in Avaya along with enabling the copy of the UCID for station conference and transfer.



```
change system-parameters features                          Page   5 of  19
                        FEATURE-RELATED SYSTEM PARAMETERS

SYSTEM PRINTER PARAMETERS
  Endpoint:                 Lines Per Page: 60

SYSTEM-WIDE PARAMETERS
                                           Switch Name:
            Emergency Extension Forwarding (min): 10
          Enable Inter-Gateway Alternate Routing? n
Enable Dial Plan Transparency in Survivable Mode? n
                                  COR to Use for DPT: station

MALICIOUS CALL TRACE PARAMETERS
               Apply MCT Warning Tone? n   MCT Voice Recorder Trunk Group:
      Delay Sending RELease (seconds): 0
SEND ALL CALLS OPTIONS
     Send All Calls Applies to: station    Auto Inspect on Send All Calls? n
              Preserve previous AUX Work button states after deactivation? n
UNIVERSAL CALL ID
     Create Universal Call ID (UCID)? y    UCID Network Node ID: 9999
     Copy UCID for Station Conference/Transfer? y
```

At that point the settings shown below in the AvayaCTServer.config can be confirmed:

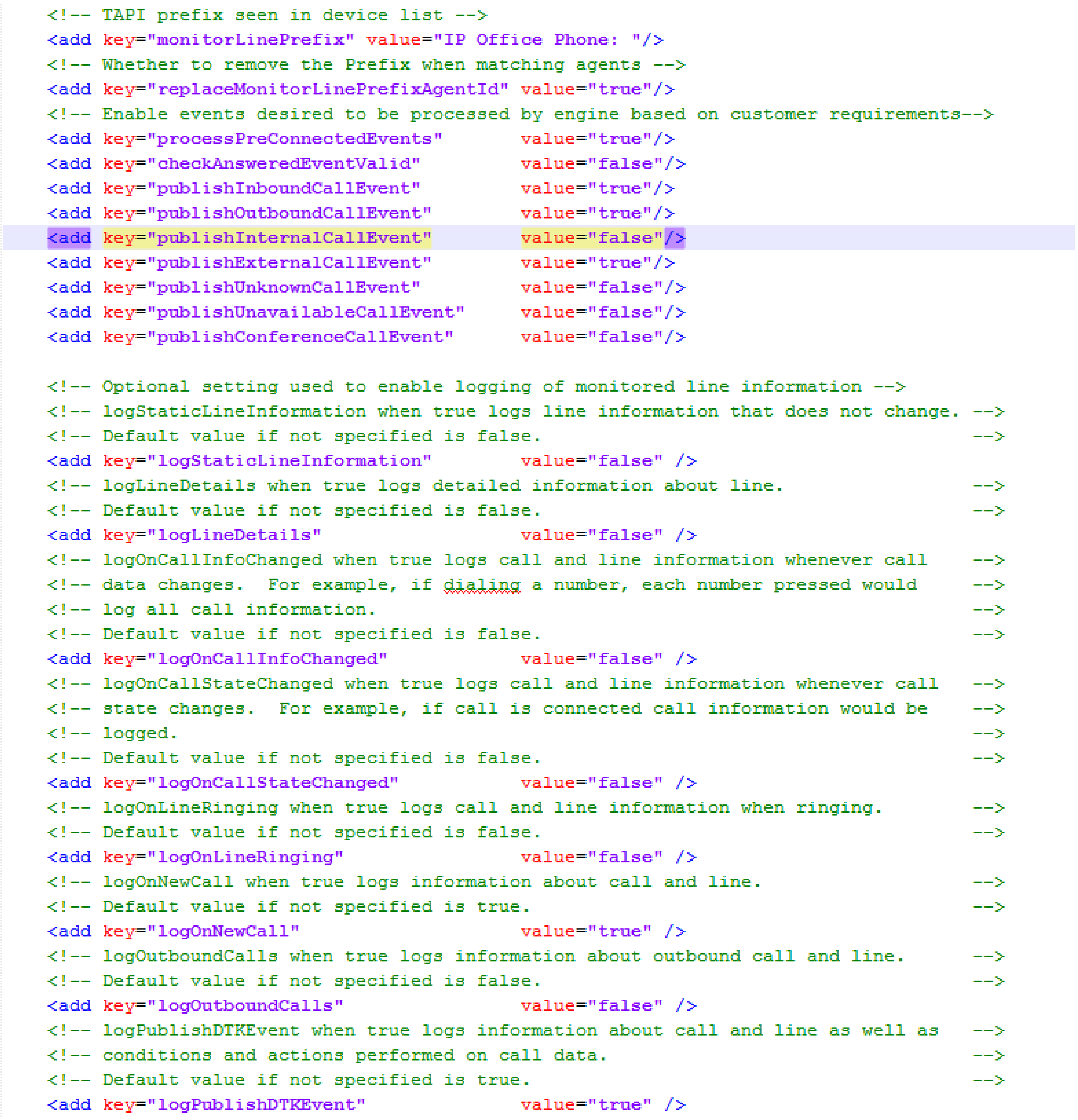


```
<add key="TSAPIserverID" value="AVAYA#SLOUGH#CSTA#SLO-AES-XEROX" />
<add key="TSAPIloginID"  value="CDCUser"/>
<add key="TSAPIpassword" value="CDCUs3r@01" />

<!-- Default value for TSAPIfilterPrivate is false  -->
<add key="TSAPIfilterPrivate"               value="false" />

<!-- Default value for TSAPIfilterPrivateContainsString is ""  -->
<add key="TSAPIfilterPrivateContainsString" value="" />

<!-- Default value for TSAPIfilterPrivate is false  -->
<add key="TSAPIfilterOutbound"              value="false" />

<!-- Default value for TSAPIfilterPrivate is false  -->
<add key="TSAPIProcessSilentMonitor"        value="false" />

<!-- Flag used to control format of event Message.         -->
<!-- If set to "True", ToString() returns pipe delimited   -->
<!-- event messsage data.  If set to false, default value, -->
<!-- ToString() returns column formatted string.           -->
<add key="ToStringAsSingleLine"             value="false" />

<!-- Publish Specific event types  -->
<add key="PublishEntryPointEvents"          value="false" />
<add key="PublishQueueEvents"               value="false" />
<add key="PublishExtensionEvents"           value="true" />

<!-- Flag used to filter call events that are not associated -->
<!-- with Monitored ACD Splits.                              -->
<add key="FilterByACDSplits"                value="false" />

<!-- Flag used to set whether Avaya Connector should call    -->
<!-- metric connector services for call metrics such as      -->
<!-- CallTime, CallTalkTime, QueueTime                       -->
<add key="PublishCallMetrics"               value="true" />
```

4.2 Avaya IP Office

The CDC Platform connector for Avaya IP Office utilizes the Avaya TAPI client. The TAPI client must be installed prior to CDC Platform configuration in 3rdy party mode with the ACD Queues option selected.

At that point the settings shown below in the AtapiMonitor.config can be confirmed:



```
    <!-- TAPI prefix seen in device list -->
    <add key="monitorLinePrefix" value="IP Office Phone: "/>
    <!-- Whether to remove the Prefix when matching agents -->
    <add key="replaceMonitorLinePrefixAgentId" value="true"/>
    <!-- Enable events desired to be processed by engine based on customer requirements-->
    <add key="processPreConnectedEvents"       value="true"/>
    <add key="checkAnsweredEventValid"         value="false"/>
    <add key="publishInboundCallEvent"         value="true"/>
    <add key="publishOutboundCallEvent"        value="true"/>
    <add key="publishInternalCallEvent"        value="false"/>
    <add key="publishExternalCallEvent"        value="true"/>
    <add key="publishUnknownCallEvent"         value="false"/>
    <add key="publishUnavailableCallEvent"     value="false"/>
    <add key="publishConferenceCallEvent"      value="false"/>

    <!-- Optional setting used to enable logging of monitored line information -->
    <!-- logStaticLineInformation when true logs line information that does not change. -->
    <!-- Default value if not specified is false.                                       -->
    <add key="logStaticLineInformation"        value="false" />
    <!-- logLineDetails when true logs detailed information about line.                 -->
    <!-- Default value if not specified is false.                                       -->
    <add key="logLineDetails"                  value="false" />
    <!-- logOnCallInfoChanged when true logs call and line information whenever call    -->
    <!-- data changes.  For example, if dialing a number, each number pressed would     -->
    <!-- log all call information.                                                      -->
    <!-- Default value if not specified is false.                                       -->
    <add key="logOnCallInfoChanged"            value="false" />
    <!-- logOnCallStateChanged when true logs call and line information whenever call   -->
    <!-- state changes.  For example, if call is connected call information would be    -->
    <!-- logged.                                                                        -->
    <!-- Default value if not specified is false.                                       -->
    <add key="logOnCallStateChanged"           value="false" />
    <!-- logOnLineRinging when true logs call and line information when ringing.        -->
    <!-- Default value if not specified is false.                                       -->
    <add key="logOnLineRinging"                value="false" />
    <!-- logOnNewCall when true logs information about call and line.                   -->
    <!-- Default value if not specified is true.                                        -->
    <add key="logOnNewCall"                    value="true" />
    <!-- logOutboundCalls when true logs information about outbound call and line.      -->
    <!-- Default value if not specified is false.                                       -->
    <add key="logOutboundCalls"                value="false" />
    <!-- logPublishDTKEvent when true logs information about call and line as well as   -->
    <!-- conditions and actions performed on call data.                                 -->
    <!-- Default value if not specified is true.                                        -->
    <add key="logPublishDTKEvent"              value="true" />
```

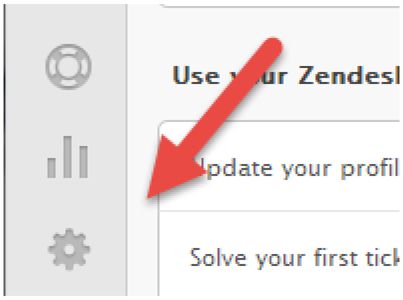
5 CRM Connector Configuration

The CRM system and the CRM connector must be configured to both allow communication between the CRM system and the CDC Platform and to associate a CRM Agent with their identifier within the telephony system. This process is similar on all CRM systems supported.

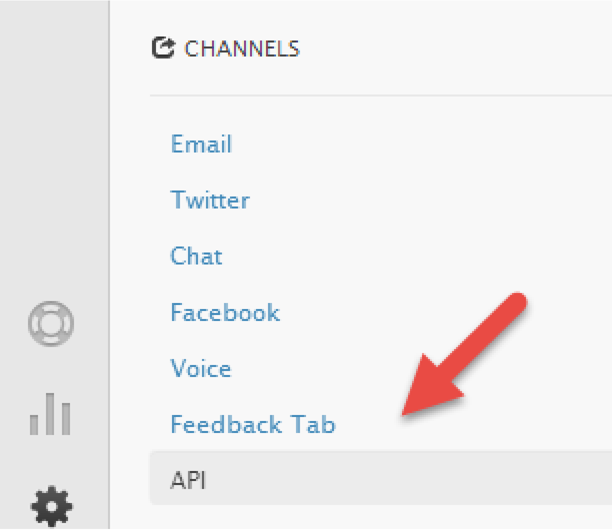
5.1 CRM configuration

Before you start configuring the CRM connector you need to configure your CRM account to work with the CDC Platform CRM Connector. The connector needs to be configured with proper API access credentials and in order for the connector to properly associate the telephony agent to their corresponding CRM agent session a CRM user field needs to be setup to contain the identifier used by the agent in their telephony system.
Steps to configure are:

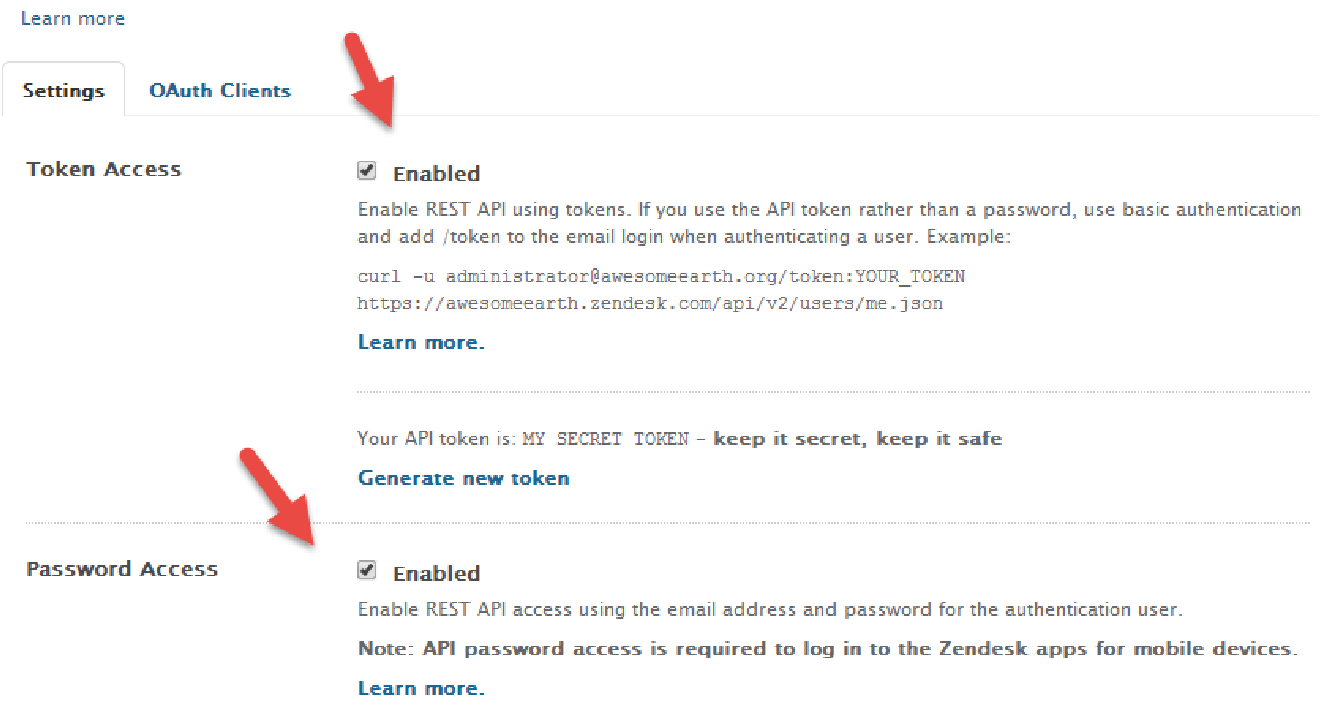
1. Log into your CRM account with the administrator user account.
2. Click on the gear icon to access the administration settings



3. Click on API under Channels



4. You can authenticate using the API token or password. This example will use token. Make sure Token Access is enabled like below.



Token Access

☑ Enabled

Enable REST API using tokens. If you use the API token rather than a password, use basic authentication and add /token to the email login when authenticating a user. Example:

```
curl -u administrator@awesomeearth.org/token:YOUR_TOKEN
https://awesomeearth.zendesk.com/api/v2/users/me.json
```

Learn more.

Your API token is: MY SECRET TOKEN – **keep it secret, keep it safe**

Generate new token

Password Access

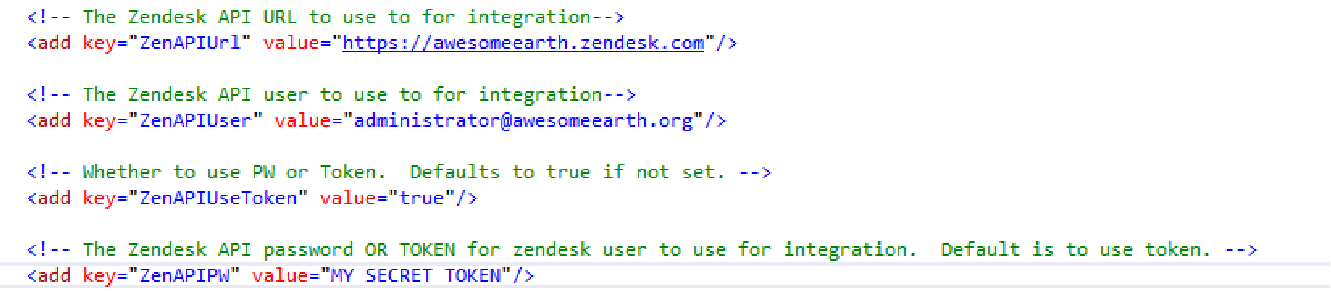
☑ Enabled

Enable REST API access using the email address and password for the authentication user.

Note: API password access is required to log in to the Zendesk apps for mobile devices.

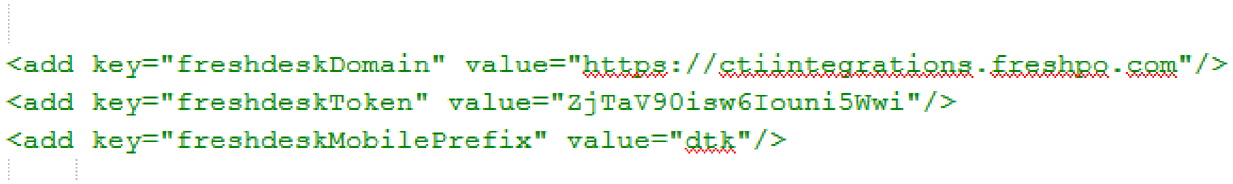
Learn more.

5. When using Token Access the configuration connector's configuration file will contain the following settings:



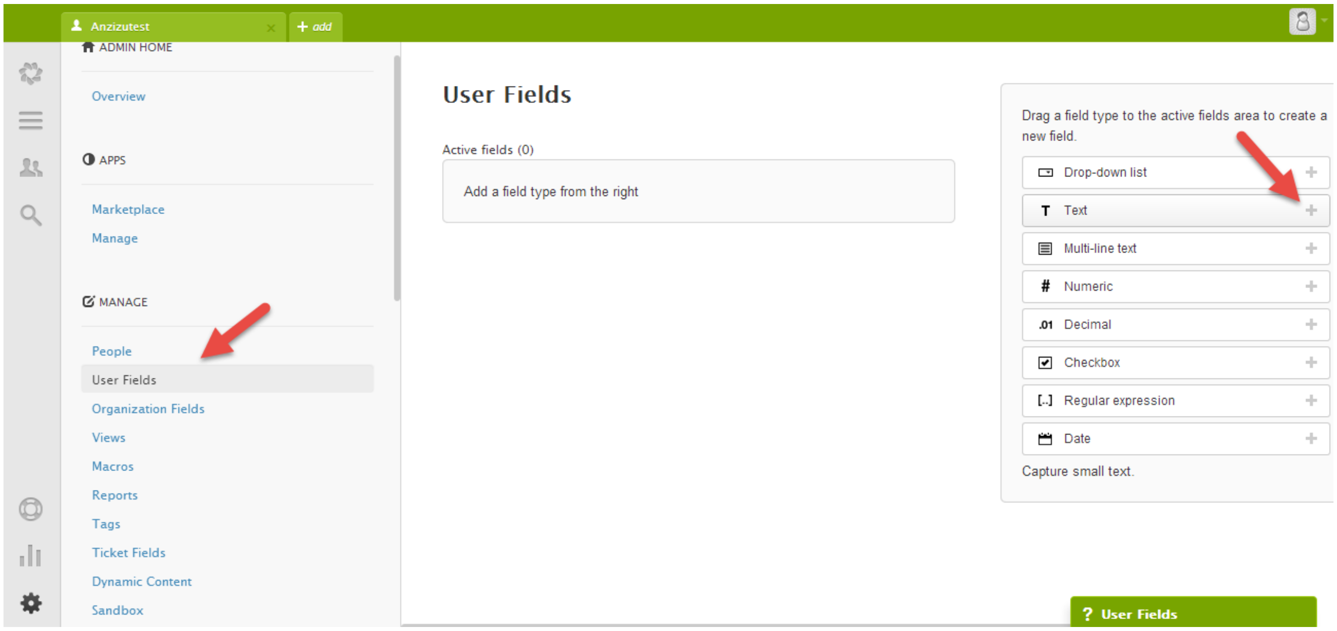
```
<!-- The Zendesk API URL to use to for integration-->
<add key="ZenAPIUrl" value="https://awesomeearth.zendesk.com"/>

<!-- The Zendesk API user to use to for integration-->
<add key="ZenAPIUser" value="administrator@awesomeearth.org"/>

<!-- Whether to use PW or Token.  Defaults to true if not set. -->
<add key="ZenAPIUseToken" value="true"/>

<!-- The Zendesk API password OR TOKEN for zendesk user to use for integration.  Default is to use token. -->
<add key="ZenAPIPW" value="MY SECRET TOKEN"/>
```

OR



```
<add key="freshdeskDomain" value="https://ctiintegrations.freshpo.com"/>
<add key="freshdeskToken" value="ZjTaV90isw6Iouni5Wwi"/>
<add key="freshdeskMobilePrefix" value="dtk"/>
```

6. Define a User Field to identify the telephony agent id with CRM agent



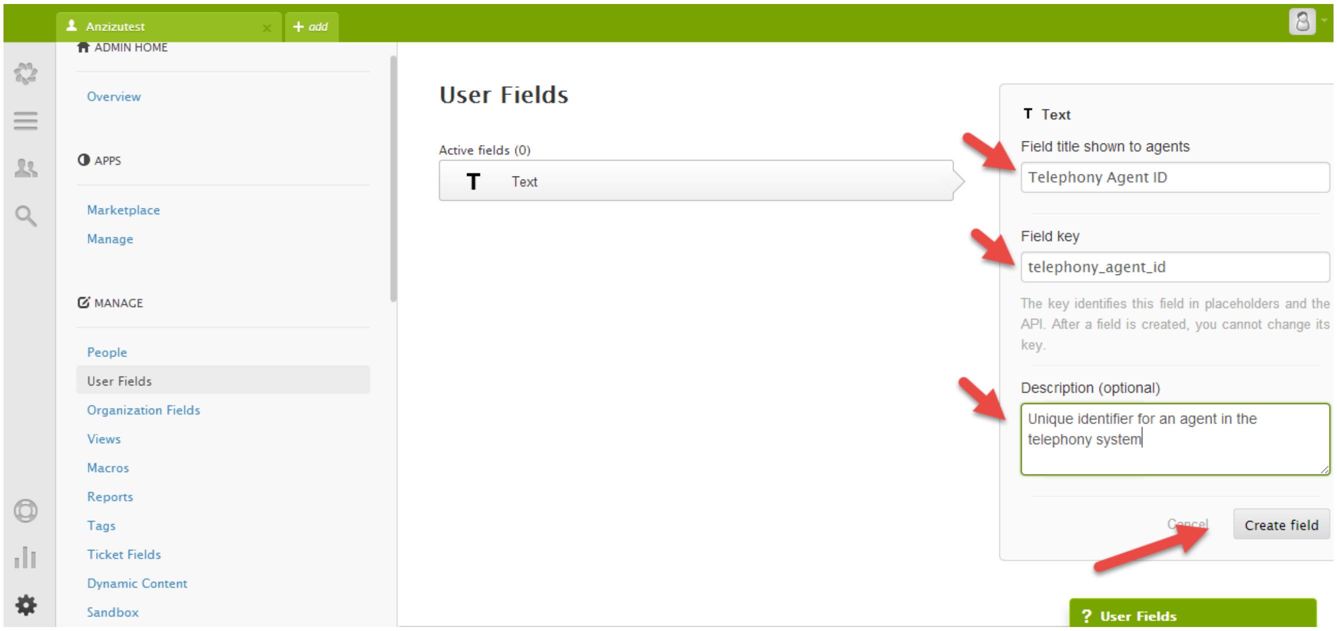
User Fields

Active fields (0)

Add a field type from the right

Drag a field type to the active fields area to create a new field.

Capture small text.



User Fields

Active fields (0)

T Text

T Text

Field title shown to agents: Telephony Agent ID

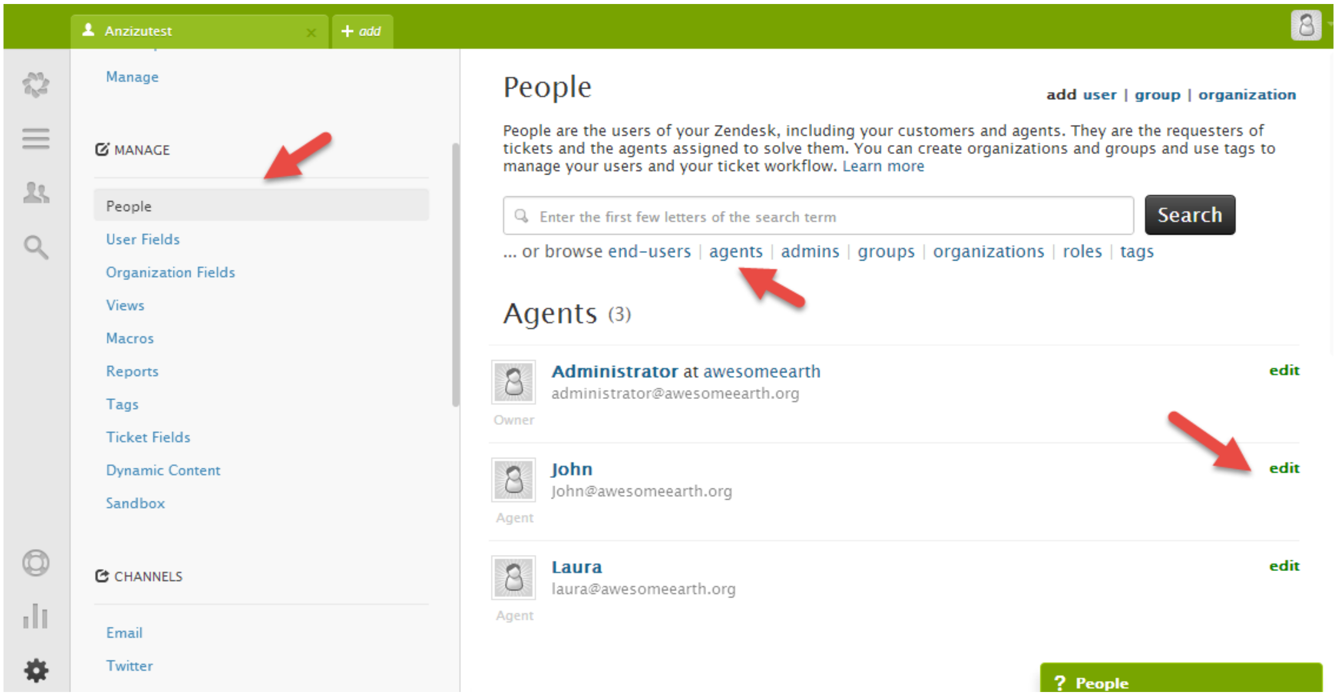
Field key: telephony_agent_id

The key identifies this field in placeholders and the API. After a field is created, you cannot change its key.

Description (optional): Unique identifier for an agent in the telephony system

The Field Key value "telephony_agent_id" will be used in this example CRM connector configuration to match events from the telephony system with your agents in your CRM account.

7. Updating your agents user field "Telephony Agent ID":
    a. Click on People, then agents to filter by agent role.
    b. For each required agent click edit to update. For larger deployments an API update tool can be utilized to set this value for each agent based on data exported from the telephony system or provided by the customer.


Anzizutest
+ add
Manage
MANAGE
People
User Fields
Organization Fields
Views
Macros
Reports
Tags
Ticket Fields
Dynamic Content
Sandbox
CHANNELS
Email
Twitter
add user | group | organization
Enter the first few letters of the search term
Search
... or browse end-users | agents | admins | groups | organizations | roles | tags
? People

People

People are the users of your Zendesk, including your customers and agents. They are the requesters of tickets and the agents assigned to solve them. You can create organizations and groups and use tags to manage your users and your ticket workflow. Learn more

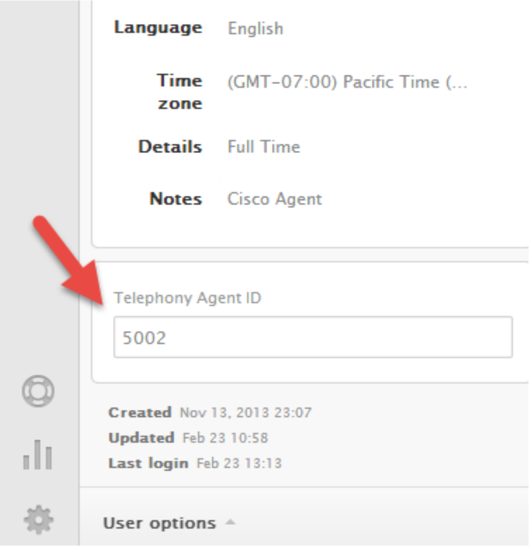
Agents (3)

Administrator at awesomeearth
administrator@awesomeearth.org
Owner
edit

John
John@awesomeearth.org
Agent
edit

Laura
laura@awesomeearth.org
Agent
edit

c. Update the Telephony Agent ID field.



Language English

Time zone (GMT-07:00) Pacific Time (...

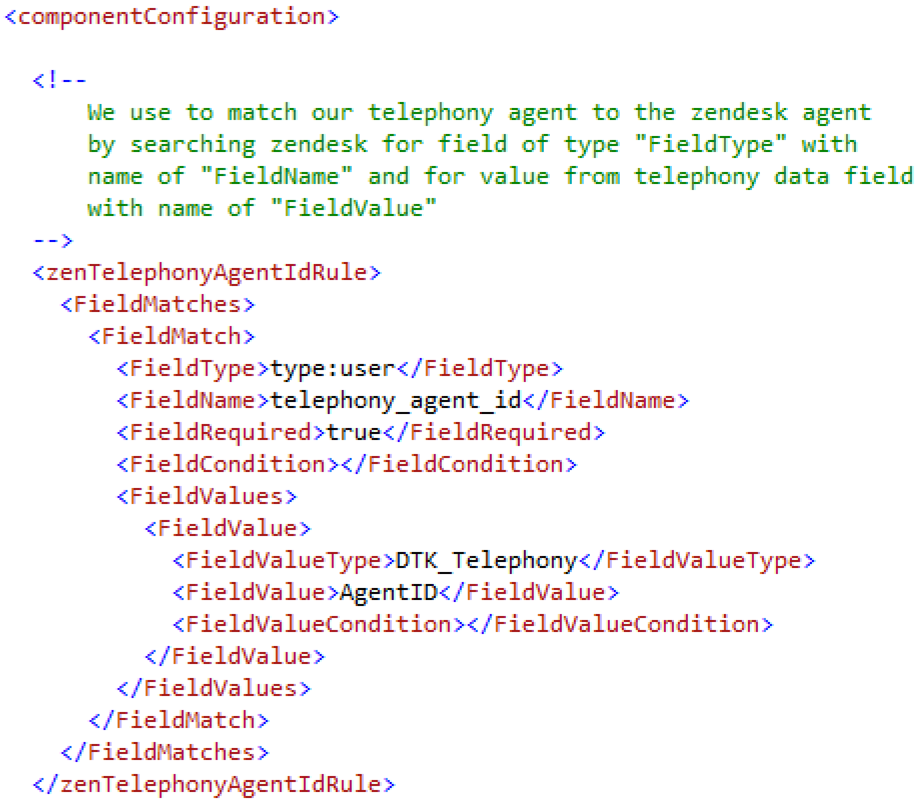
Details Full Time

Notes Cisco Agent

Telephony Agent ID
5002

Created Nov 13, 2013 23:07
Updated Feb 23 10:58
Last login Feb 23 13:13

8. The zenTelephonyAgentIdRule in the CRM connector configuration relates CRM agents with Telephony Agents. Below is the rule as defined for our example. “In short the rule states: “Match the CRM user field with name of telephony_agent_id to the telephony event field named AgentID”.



```xml
<componentConfiguration>

  <!--
      We use to match our telephony agent to the zendesk agent
      by searching zendesk for field of type "FieldType" with
      name of "FieldName" and for value from telephony data field
      with name of "FieldValue"
  -->
  <zenTelephonyAgentIdRule>
    <FieldMatches>
      <FieldMatch>
        <FieldType>type:user</FieldType>
        <FieldName>telephony_agent_id</FieldName>
        <FieldRequired>true</FieldRequired>
        <FieldCondition></FieldCondition>
        <FieldValues>
          <FieldValue>
            <FieldValueType>DTK_Telephony</FieldValueType>
            <FieldValue>AgentID</FieldValue>
            <FieldValueCondition></FieldValueCondition>
          </FieldValue>
        </FieldValues>
      </FieldMatch>
    </FieldMatches>
  </zenTelephonyAgentIdRule>
```