3.14 Synchronizing the MS Teams Presence with Openscape UC

The synchronization between MS Teams and OpenScape UC allows for changing the status of the MS Teams user to busy whenever the OpenScape UC presence status is set to busy as well. Whenever the status of the MS Teams user changes from busy to a different option, the previous status will be restored in OpenScape UC too. The opposite case will be also supported, thus, if the UC presence status is changed to busy then the MS Teams status will be changed as well.

Therefore, only the busy (in a call) status of the Microsoft Teams user will be integrated into the OpenScape UC user presence status.

This is achieved by integrating the MS Graph interface in a new edge service and associating the MS teams account with the external id.

The table below displays when the MS Teams/OpenScape UC Presence changes to busy in a call:

MS Teams presence status prior to making a call	MS Teams presence status during a call
"availability": "Available", "activity": "Available",	"availability": "Busy", "activity": "InACall",
"availability": "Busy", "activity": "Busy",	"availability": "Busy", "activity": "InACall",
"availability": "DoNotDisturb", "activity": "DoNotDisturb",	"availability": "DoNotDisturb", "activity": "DoNotDisturb",
"availability": "BeRightBack", "activity": "BeRightBack",	"availability": "BeRightBack", "activity": "BeRightBack",
"availability": "Away", "activity": "Away",	"availability": "Away", "activity": "Away",
"availability": "Offline", "activity": "OffWork",	"availability": "Offline", "activity": "OffWork",
"availability": "Busy", "activity": "InAMeeting",	"availability": "Busy", "activity": "InACall",

Tabelle 11 MS Teams/OpenScape UC Presence

NOTE: The user activity **"InAMeeting"** is fetched through Outlook synchronization, and it is shown when the user is not in a meeting call. The user activity is **"InACall"** regardless of whether the user is an MS meeting or not.

NOTE: As per MS Teams API restriction, the subscription limit for MS teams users presence is limited to 650 users per technical ser account.

3.14.1 Configuration in MS Azure Active Directory

Creation of the Azure application in Microsoft Azure Active Directory admin center:

1. To create an Azure application you need to login to Microsoft Azure Active Directory admin center with an administrator account and navigate to **Azure Active Directory > Add > App Registration**.
2. Perform a new Registration using the default settings to create an Azure application, as in the following example:

- Application name: UCPresenceApp;
- Accounts in this organizational directory only (Single tenant);
- There is no need to define a Redirect URI.

3. Go to the **Authentication** option after selecting the newly created application. Set the **Allow public client flows** option to Yes.
4. Go to the **Certificates & secrets** section of the application and define a New client secret. Remember the value of the secret as it will not be available in the future. Please note that the client secret expires, therefore an update will be required before expiry.
5. Go to the **API permissions** section of the application and add the following permissions for the **Microsoft Graph** application:
   - **Presence.Read.All** for **Delegated** usage.
   - **Presence.ReadWrite.All** for **Application** usage.
   - **User.Read.All** for **Application** usage.

   Admin consent will be required to apply the changes.

3.14.2 UC MS Graph Connector app in Facade node

1. Install the optional package for the UC MS Graph Connector app:

   osc-setup up osc-setup

   osc-setup in OpenScapeUC_Graph_Connector_Facade
2. Configure the UC Graph Connector in the Facade node.
3. Edit the file located under **/opt/uc-graph-connector/application.properties**
4. Fill in the following properties:
   - **azure.activedirectory.clientId**: the application client id of your Azure registered application.
   - **azure.activedirectory.tenantId**: the tenantId for your organization in Azure ADD.
   - **azure.activedirectory.clientSecret**: the application client secret defined in your application that was registered in Azure.
   - **azure.activedirectory.technical.user.username**: the username of the user that will be used as technical user to allow OpenScapeUC to use Microsoft Graph API.
   - **azure.activedirectory.technical.user.password**: the password of the technical user.

- **webhook.receiver**: the public URL in which UC Graph Connector can be accessed publicly and receive events from Microsoft Graph.

  Example:

  webhook.receiver = [https://[Public_FQDN/IP_Facade_node]:8443/graph/listen|https://[public_fqdn/IP_Facade_node]:8443/graph/listen]|

- **webhook.listener.http.url**: the UC FE endpoint that point to the FE that will receive the events from UC Graph Connector and will propagate them further to UC Backend. In case multiple FEs are defined they need to be comma separated.

  Example:

  webhook.listener.http.url = [https://[Public_FQDN/IP_Frontend_node]:8443/owc-servlets/user/graph/notifications|https://[private_fqdn/IP_Frontend_node]:8443/owc-servlets/user/graph/notifications]|

For more information about how to configure TLS versions, ciphers and customer certificates, refer to Configuring the Certificate for the UC Graph Connector.

3.14.3 Configuration of the callback uri in the HAproxy

The webhook.receiver public URL, used for Microsoft Graph event reception must be correspondingly configured in the reverse Proxy. For more information, refer to Configuring HAProxy.

3.14.4 Configuration of UC Teams Graph Connector in UC BE

1. Navigate to the config beans directory in BE (/opt/siemens/common/conf/config-beans) and locate the config bean for the new OSGi bundle (displayName="Teams Connector Service"). The internal URL of façade node has to be defined in this config bean to allow the communication between Presence BE and Graph Connector in the Facade node.
2. Edit the **GraphConnectorURL** property, as the the example below:

   ```
   <Property name="GraphConnectorURL" value="https://
   10.11.246.199:10443" writable="true"/>
   ```

NOTE: Please note that here we use the internal IP address of the Graph Connector since they will communicate within the intranet. The definition of this property will be the triggering point that will indicate the busy (in a call)