# Overview

Current Behavior:  
 - Keep the TFS session active for 1 hour. Try to access My Profile page after 1hr, User redirected to the IAM login page in the new tab.

- If TFS session logged out manually then My Profile page is remains blank on next action.

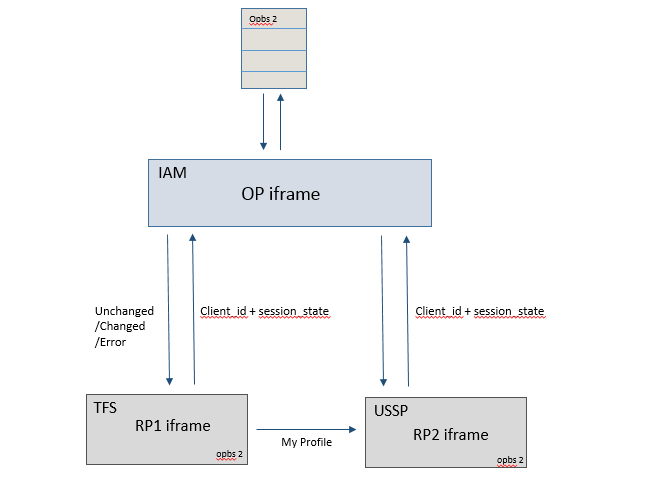
Expected Behavior and scenarios to be handled:  
 - If TFS session is active then even after 1 hour user shall be able to access the My profile page.

- Logout from TFS and refresh USSP page, USSP page should redirected to Login page.

- Log out manually from My Profile page, Associated TFS session should also log out (As per discussion with SE).  
  
 - Close the browser and reopen again and verify TFS and USSP goes to Login page

To achieve the expected behavior, it is highly desirable to be able to find out the login status of the End-User(RP) at the OpenID Provider(OP).  
  
In OIDC, there are some ways to check the login status of the end user.  
  
1) Send prompt=none as a request parameter in authentication request.  
2) By using hidden RP iframe polling the OP iframe.  
  
We have opt for 2nd option related to iframe polling. As it make use of OpenID connect Session Management which gives a better solution without causing any network traffic.

# TFS and USSP workflow with IAM **by using hidden RP iframe and OP iframe.**



**RP** : Relaying Party  
**OP** : OpenID Provider(OP)

**OPBS** : OP Browser State. All the RPs sharing the same browser session will have the same *opbs* cookie and the OP server will store that in a database for the particular browser session.

**Unchanged:** User authentication status at the OpenID provider has not changed. **Changed:** User authentication status has changed, due to logout, an expired session or some other event. Re-authenticate with prompt=none and id\_token\_hint=old ID token. **Error :** RP should not re-authenticate and handle that as a **logout** of the end user.

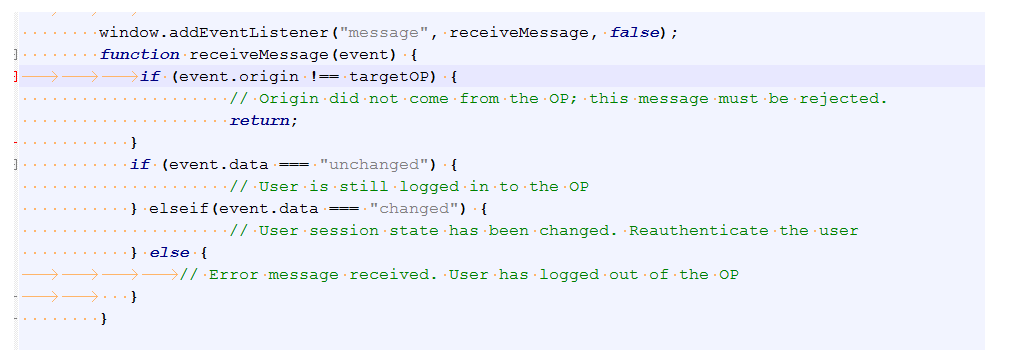
# Procedure **OIDC session management from RP Side**

1. Create an html page and load an invisible RP iframe from TFS.   
   Call the OIDC Session iframe endpoint and pass TFS logged in user client id as query parameter. A RP iframe will be like below :  
     
   Note : Need to set the logged in user client-id in the cookie.
2. To check the login status of a user, invoke the Window.postMessage() function of the OP iframe, passing the client ID and session state as the message parameter in the format Client ID + " " + Session State and the host name of the OP as the target origin parameter. Set a time interval to check end-user session.



Note : Need to persist session\_state and nonce value during authentication process. Need to decide time interval for polling. Or We can check session state on each api call so that we can get updated state immediately. This approach is implemented by USSP also.

1. Write a code to listen postMessages response from the OP that contain the value changed/unchanged/Error to reflect the respective login status of the user.



# Handle silent authentication

Once we get the “changed” session response from OP, we need to re-authenticate the user.  
  
To re-authenticate, RP will send authentication request to OP using authentication endpoint.  
  
In TFS currently there are three ways to re-authenticate the user.  
  
1) On session state changed, call the logout api explicitly. This way user is getting re-authenticated.  
But for every authentication request whole application gets reload. This can interrupt the user experience.  
  
2) Call the authentication endpoint using window.location.href.  
This way we can get the updated session-state in the response but in the url.  
As the response return in url, application reload and lands on error page.  
  
So there are limitation in the existing behavior.   
  
3) To re-authenticate the user without affecting user experience we need to implement OIDC silent-authentication.

**So silent-authentication can be implanted using “oidc-client-js”.  
 -** Download oidc-client-js from <https://cdnjs.com/libraries/oidc-client>  
 - Create a html page “silent-refresh.html” for silent-authentication  
 - Include oidc-client-js file silent-refresh.html. Configure the settings and send the re-authentication request using oidc-client-js functions.