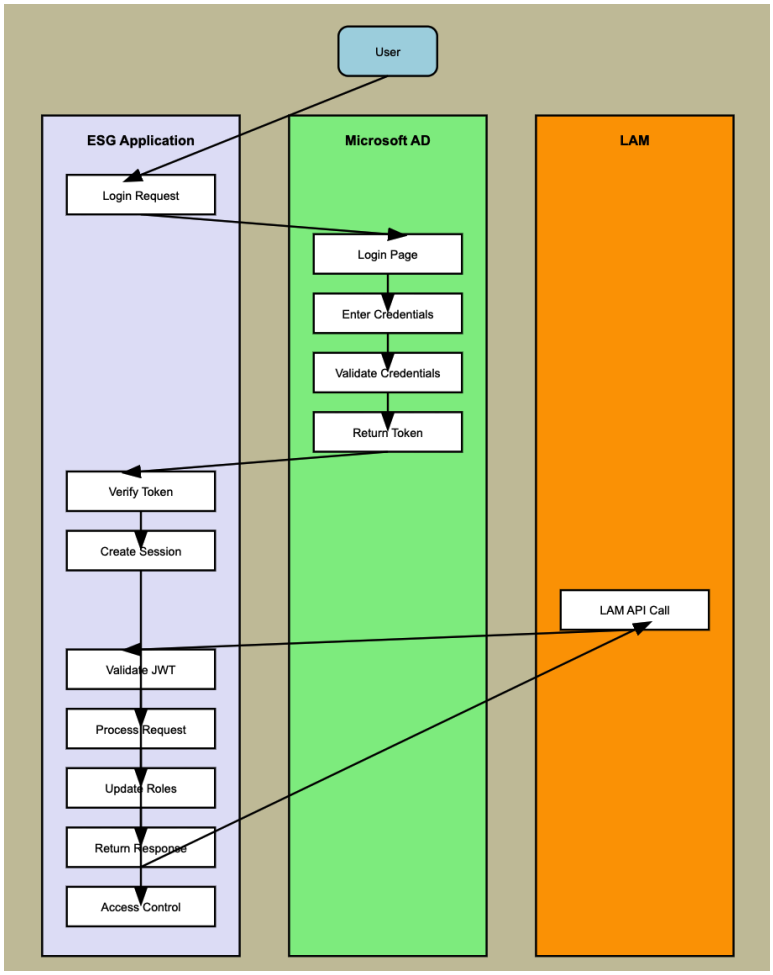
LAM and AD Integration Concept Document

This document outlines the proposed integration between the ESG application and the client's Logical Access Module (LAM) and Active Directory (AD) systems. The integration aims to provide seamless authentication for client AD users, authorization, and role management through LAM.

1. **Authentication Flow:**
    - Users will authenticate directly through Microsoft AD.
2. **Authorization and Role Management:**
    - Roles will be predefined in ESG application and shared with the client's IT team.
    - Post API for role-updates to be customized as per shared structure and documentation to be shared with the client's IT team.
    - LAM will use APIs to create users and assign roles.
    - Role updates will be processed via API calls.
3. **User Provisioning:**
    - All user provisioning will be done through LAM .
    - User deprovisioning and account deactivation will be handled through LAM-generated requests.
4. **API Security:**
    - JWT token-based authentication will be used for LAM to consume APIs.
    - Additional measures to be discussed.
5. **Audit and Compliance:**
    - Audit logging for user activities and role changes will be maintained in ESG application.
6. **Error Handling and Monitoring:**
    - Integration errors will be handled and reported as per the error codes provided in the document.
7. **Data Synchronization:**
    - Data synchronization will be trigger-based whenever APIs are called.
8. **Environment and Deployment:**
    - Support for dev, test, and prod environments will be provided.

High Level Integration Flow:



1. **User Authentication:**
   a. User attempts to log in to the ESG application.
   b. The application redirects the user to the Microsoft AD login page.
   c. User enters credentials and completes MFA if required.
   d. AD validates the credentials and returns an access token.
   e. ESG application verifies the token and creates a session for the user.

2. **Role Assignment and Management:**
   a. LAM sends a POST request to ESG Application via API with user and role information.
   b. ESG Application validates the JWT token from LAM.
   c. The Role Management service processes the request and updates the user's roles in the database.
   d. The API returns a success or error response to LAM.

3. **User Access Control:** a. When a user attempts to access a feature or module, the application checks their roles.
   b. The Role Management service retrieves the user's roles from the database.

c. The application grants or denies access based on the user's roles.

4. **User Deprovisioning:** a. LAM sends a delete or deactivate request to ESG Application via API.
   b. ESG Application processes the request and updates the user's status in the database.
   c. The application terminates any active sessions for the deprovisioned user.

Points to be Discussed :

1. Specific requirements for multi-factor authentication (MFA) implementation.
2. Any specific performance requirements for authentication and authorization processes.
3. Preferred fallback authentication method in case of AD unavailability.
4. Any additional security measures required for API communication between LAM and ESG application.

Required Active Directory Information

To successfully integrate esg application with Active Directory environment, we require the following information:

Basic AD Information

- AD Domain Name
- AD Forest Name (if applicable)

User Information

- Required user attributes (username, organization, email, department, role,etc.)

Network Connectivity

- Network access requirements
- Firewall rules (if applicable)

Azure AD Specifics

1. **Azure AD Tenant ID:**

   The unique identifier for Azure AD tenant.

2. **Application (Client) ID:**

   The ID assigned to application in Azure AD.

3. **Client Secret (if using OAuth 2.0/OpenID Connect):**

   The secret key generated for application in Azure AD.

4. **SAML Entry Point (Azure AD Endpoint):**

   The SAML endpoint URL provided by Azure AD.

   Example: `https://login.microsoftonline.com/{tenant_id}/saml2`

5. **SAML Callback URL:**

   The endpoint where Azure AD will send the authentication response.

   Example: `https://esg-app.com/auth/callback`

6. **SAML Issuer (Identifier):**

   The unique identifier assigned to the application by Azure AD.

   Example: `https://sts.windows.net/{tenant_id}/`

7. **Certificate Path:**

   The path to the X.509 certificate file used for signing SAML requests.

   Example: `./certs/azure_ad_cert.pem`

8. **Key Path:**

   The path to the private key file associated with the certificate.

   Example: `./certs/azure_ad_key.pem`

9. **Session Secret:**

   A secret key used to sign session cookies. This should be a random and secure string.

   Example: `random_session_secret`

Please note that some of these details may vary depending on specific AD setup and the chosen authentication method.