Analysis: Token Exchange vs Client Credentials for CSS-214

User Story: CSS-214

Topic: Applicability of SAML Token Exchange for backend job

# Background

AIB’s identity management supports different flows for obtaining tokens:  
  
1. SAML/JWT Token Exchange Flow  
 - Used in user-facing scenarios (e.g., Jira, BFF applications).  
 - Starts with a subject token (JWT/SAML assertion from Ping Access).  
 - Exchanged with Ping Federate for a staff token enriched with business attributes (staff ID, CIF key, region).  
 - Required when APIs mandate user identity context.  
  
2. Client Credentials Flow  
 - Designed for service-to-service (machine-to-machine) integrations.  
 - A backend system presents its client ID and secret to Ping Federate.  
 - Ping Federate issues an access token representing the system identity.  
 - Downstream APIs validate the token against the system client configuration.

# Our Use Case (CSS-214)

- The system in scope is a backend job with no interactive user login.  
- It must call Collate APIs, which are hosted on-prem.  
- Current setup: standard Client Credentials Flow.

# Observed Issues with Current Tokens

Tokens obtained via the client credentials grant show two limitations when validated by Collate:  
  
1. No 'sub' claim  
 - The access token does not include a 'sub' (subject) field.  
 - Collate APIs enforce this as part of their validation standard.  
  
2. Issuer ('iss') value lacks 'https://'  
 - Example: token shows iss=authz-security-test.aib.pri instead of iss=https://authz-security-test.aib.pri.  
 - Collate requires issuer URIs to be properly prefixed with 'https://' as per OAuth 2.0 / OIDC conventions.  
  
Because of these two issues, tokens from the pure client credentials flow are rejected by Collate APIs during standard validation.

# Analysis

- Token Exchange Flow is not inherently designed for backend jobs since no user identity is involved.  
- Client Credentials Flow is the correct flow for machine-to-machine scenarios.  
- However, the Collate product enforces stricter validation on tokens:  
 - 'sub' must be present.  
 - 'iss' must be in HTTPS URI format.  
  
Thus, while token exchange is not conceptually needed, the format of tokens from client credentials may not be compatible with Collate without configuration adjustments on the IdP (Ping Federate).

# Conclusion

- For CSS-214, the backend system is machine-to-machine.  
- Client Credentials Flow remains the correct approach.  
- SAML Token Exchange is not required.  
- But: The current tokens from client credentials do not meet Collate’s validation rules due to missing 'sub' and non-HTTPS issuer.  
  
Recommendation: Engage with the Ping Federate / Identity team to update the client configuration so that:  
1. Tokens include a valid 'sub' claim (service identity).  
2. Issuer value follows HTTPS URI standard.  
  
Once addressed, the backend job can call Collate APIs directly with client credential tokens.