Introduction

Purpose and scope

This runbook explains how a public client, such as a web or mobile application, can integrate with the Verifier acting as an Authorization Server (AS) using the Authorization Code Flow with PKCE. It provides developers with the end-to-end steps required to obtain and use tokens securely, from initiating the authorization request to exchanging the authorization code and calling protected APIs.

Main aspects covered include:

- Integration of public clients with the Verifier using Authorization Code Flow + PKCE.
- Secure use of PKCE (Proof Key for Code Exchange) to prevent authorization code interception.
- OAuth 2.1 authorization_code profile with code_verifier and code_challenge.
- Token acquisition and usage for accessing Verifier-protected resources.
- Security considerations, error handling, and observability.

Intended audience

- Frontend developers integrating web or mobile applications with the Verifier.
- Technical integrators responsible for configuring the public client.
- SRE and security engineers reviewing client security compliance.

High-level architecture

embedded-image-AuthCodePKCE-arch.png

1. The public client redirects the user to the Verifier Authorization Endpoint, including the code challenge (PKCE) and other OAuth parameters.
2. The user authenticates and grants consent.
3. The Verifier returns an authorization code to the client via redirection.
4. The client exchanges the authorization code for tokens at the Token Endpoint using the code_verifier.
5. The Verifier issues access and ID tokens with limited lifetime.
6. The client uses the access token to call protected APIs.

High-level flow

embedded-image-AuthCodePKCE-flow.png

1. The user initiates the login process from the public client, which sends an authorization request to the Verifier (AS).
2. After successful authentication and consent, the AS returns an authorization code.
3. The public client securely exchanges the authorization code and code_verifier for tokens.
4. The AS issues an access token and an ID token.
5. The client uses the access token to access protected resources on the Verifier.
6. The resource server validates the access token and returns the requested data.

Revision #1
Created 10 November 2025 15:24:05 by Roger Miret
Updated 10 November 2025 15:42:21 by Roger Miret