CGR SAP Connector

Security And OAuth 2.0

Authentication establishes who is calling. CGR whitelisting and SAP authorization determine what they may reach. Apply all three layers.

TECHNICAL DOCUMENTATION

Version 1.0 · September 2026

The Three Layers

Authentication establishes caller identity; CGR whitelisting and SAP authorization determine access permissions. Apply all three security layers for comprehensive protection.

- **Transport security.** Activate SSL and use HTTPS exclusively.
- **Identity isolation.** Deploy a separate SAP identity for each external integration, rather than sharing one.
- **CGR authorization.** Configure the permitted objects in `/CGR/API_AUTH`.
- **Backend authorization.** Preserve the standard SAP checks for tables, RFC functions and OAuth scopes.
- **Auditability.** Use `/CGR/MONI` and `/CGR/API_LOG` for request and response tracing.

BASELINE RECOMMENDATIONS

Implement HTTPS, use dedicated technical identities, apply the narrowest possible CGR object whitelists, and maintain narrowly scoped SAP roles. Store credentials securely outside source code, and rotate them to your own policy.

SICF Authentication

The SICF product assigns user credentials on the service's **Logon Data** tab, for non-interactive endpoints.

- Deploy restricted technical users rather than personal dialog accounts.
- Restrict object access strictly in `/CGR/API_AUTH`.
- Assign only the necessary SAP table, report and RFC authorizations.
- Mandate HTTPS, to prevent credential exposure over unencrypted channels.

WHAT STORING CREDENTIALS ON THE SERVICE MEANS

Every call then runs as that single service user. It is convenient, but it removes per-caller attribution — so use it only where the caller is itself a trusted, authenticated system.

OData: OAuth 2.0 Client Credentials

Prerequisites: a functional SAP NetWeaver Gateway, HTTPS/SSL configured in `STRUST`, and administrative access to the Gateway security transactions.

1. Create a System-type user in `SU01` (for example `OAUTH_CLIENT_01`), configure a secure secret, keep it active, and add it to `/CGR/API_AUTH`.
2. Enable OAuth on the OData service via `/IWFND/MAINT_SERVICE`, and confirm the OAuth indicator appears.
3. Register the client at the OAuth configuration application, reachable via `SOAUTH2`. Ensure the client ID matches the system user.
4. Add the service-generated scope and select the **Client Credentials** grant type.

```
https://<server>:<port>/sap/bc/webdynpro/sap/oauth2_config?sap-client=<client>&sap-language=EN#
```

Scope Authorization

1. Open `PFCG` and create a role such as `Z_OAUTH_ODATA_ACCESS`, or update an existing one.
2. Manually add authorization object `S_SCOPE`.
3. Set `OA2_CLIENT` to the registered client ID (for example `OAUTH_CLIENT_01`).
4. Set `OA2_SCOPE` to the generated service scope (for example `/CGR/CONN_SRV_0001`).
5. Generate the profile and assign the role to the identity used for OAuth-protected calls.

A VALID BEARER TOKEN IS NOT ENOUGH

The SAP identity still requires both its `/CGR/API_AUTH` configuration and the standard backend authorizations for the objects requested. Authentication and authorization remain separate questions.

TRANSACTIONS USED

Transaction	Purpose
`SU01`	Maintain the OAuth technical user
`/IWFND/MAINT_SERVICE`	Register the OData service; enable the OAuth scope
`SOAUTH2` / `oauth2_config`	Register clients; select grant types; assign scopes
`PFCG`	Create the role; maintain the `S_SCOPE` authorization object
`/CGR/API_AUTH`	Whitelist the accessible SAP tables, functions and reports

CGR SOFTWARE

Send this document to your security and Basis teams.

cgrsoft.com
sales@cgrsoft.com