

Consumer Consent Service (CCS) API Technical Specification

Contents

1	Introduction.....	4
2	CCS Directory Discovery	5
3	CCS Authorisation Server.....	8
4	CCS Shared Signals Framework.....	15
5	Account Holder Verification	17
6	Errors and Response Codes	18
7	Privacy Considerations	20

Technical Specification Document

Consumer Consent Service Definition

Version: 0.1

Effective Date: TBC

Change History

Version Number	Implementation Date	Reason for Change
0.1	N/A	Initial draft for CCS industry consultation

1 Introduction

- 1.1. The Consumer Consent Service (CCS) provides a central, consistent mechanism for capturing, managing and evidencing Permission provided by an Energy Data Subject for the sharing of Energy Data across the retail market. It enables Authorised Third Parties (ATPs) to request and obtain Permission in a standardised manner and ensures that Energy Data Subjects retain visibility and clear control over how their Energy Data is used.
- 1.2. This document describes the mechanisms enabling CCS Users to interact with the CCS via API. Interactions managed via the CCS User Portal are not included in this document. This is a Category 2 Product that forms part of the REC Technical Specification.
- 1.3. There are two separate API based routes for CCS Users to interact with the CCS:
 - (a) interactions with the CCS Authorisation Server, including the initial request for authorisation and ongoing management of Access Tokens and Refresh Tokens; and
 - (b) interactions with the CCS Directory, to maintain organisational data such as user endpoints, contact details and security credentials, and to allow other CCS Users to request this information to support validation activities.
- 1.4. The scope of this document is to describe:
 - (a) the purpose of each of the CCS APIs;
 - (b) the discovery and governance flows the APIs support, from ecosystem discovery through to cryptographic trust and certification;
 - (c) the API operational characteristics including request correlation, pagination and filtering, error handling and throttling;
 - (d) the mechanism for interacting with CCS endpoints and associated validation;
 - (e) the error codes to be used when API validation fails.
- 1.5. This API Technical Specification should be read in conjunction with:
 - (a) The CCS Service Definition, which defines the service being delivered by the CCS Provider;
 - (b) the CCS Arrangements Schedule, which sets out the rights and obligations of CCS Users;
 - (c) the Data Specification, which defines the Market Messages used to interact with the CCS and the associated Data Items including the Permission authorisation details which define the Permission payload referred to in Paragraph 3; and

- (d) the Consumer Experience Guidelines (CEGs) which set out the principles, standards and expectations for how Energy Data Subjects should be supported, informed and treated when interacting with ATPs.

2 CCS Directory Discovery

- 2.1. The CCS Directory records details of ATPs i.e., organisations wishing to access Energy Data and Energy Data Holders (EDHs) i.e., organisations wishing to share Energy Data and details of available Data Products linked to individual EDHs, allowing ATPs to view the available data and associated EDH.
- 2.2. The following information is held within the CCS Directory:
 - (a) organisations and their administrative users - the accredited legal entities and the people authorised to manage their presence in the ecosystem.
 - (b) servers - the servers an organisation operates within the ecosystem; for an EDH, this is where its data services are anchored.
 - (c) API resources and discovery endpoints - the Data Products an EDH publishes, and the addresses at which they can be reached.
 - (d) applications - the applications an organisation runs, and the roles and authorised capabilities each has been granted.
 - (e) certificates - the security certificates issued by the CCS registration authority, used to authenticate a CCS User and to sign CCS Market Messages.
 - (f) certifications - evidence demonstrating that a CCS User has passed the assurance checks required before it is Qualified.
- 2.3. Two specific APIs are available to support CCS Directory discovery:
 - (a) The Participants API, which provides an endpoint allowing a user to retrieve an up-to-date list of all participants registered within the ecosystem in a single call.
 - (b) the Connect API, which provides a deeper level of information allowing an ATP or EDH to discover information relating to specific ATPs and EDHs including API endpoints for requesting Energy Data, security certificates and keys for validation and participant roles to understand what authorised capabilities are in place. The Connect API is only available to Qualified CCS Users.
- 2.4. The CCS Directory supports separation of roles within ATP and EDH organisations, enabling controls to be applied at the correct level. The different roles are explained below, together with details of how the Participants API and Connect API are used to access information:
 - (a) The legal entity - the APIs support identification of the CCS User, exposing participant and organisation-level directory information. The Participants API provides an up-to-date list of participants registered within the ecosystem or

federation, while the Connect API organises detailed CCS Directory resources under organisations.

- (b) The registered CCS User - the Connect API uses the organisation as the primary parent resource for related CCS Directory records, including servers, applications, certificates and administrator users. This allows CCS to distinguish between the registered CCS User and the individual systems, software or the certificates which it operates.
- (c) The software client or application - the Connect API includes ATP endpoints that allow registered software clients to be listed and retrieved separately from the organisation itself. This enables governance to be applied to a specific application or client, rather than assuming that every application operated by an organisation has the same authorised capabilities or assurance status.
- (d) The server - the Connect API provides endpoints to list and retrieve servers associated with an organisation. This allows CCS to identify the specific server against which an organisation's API resources, discovery endpoints and certifications are registered, rather than treating these as generic organisation attributes.
- (e) The API resource - The Connect API allows API resources to be retrieved under an organisation's server. This helps identify which actual API capabilities are exposed through that server, such as Permission APIs, data-access APIs, revocation APIs or discovery APIs.
- (f) The roles and authorised capabilities - the Connect API includes authority claim endpoints which are attached to applications. This allows the CCS to express what a particular software client is authorised to do, for example whether it is permitted to act in a defined role or access a particular data type. As a result, authorisation can be linked to the software and its approved role, not just to the organisation.
- (g) The certificate or key - the Connect API provides certificate endpoints at both organisation level and application level. This makes it possible to distinguish between certificates or keys that identify the organisation from certificates that are associated with a specific software client, supporting secure communications, signing, authentication and key lifecycle management.
- (h) The certification evidence - the Connect API includes certification endpoints for servers and applications. This allows CCS to hold and retrieve technical conformance or testing evidence associated with a server or application. Such evidence may support the CCS Testing and Qualification processes but does not determine whether a CCS User is Qualified, which remains a decision for RECCo in accordance with the CCS Arrangements Schedule.

2.5. Neither the Participants API nor the Connect API define the Permission or Energy Data access transaction itself. Instead, they provide the CCS Directory metadata that can be used before or during data sharing requests to ensure that the CCS User,

application, server, certificates and authority claims are valid and appropriate.

- 2.6. The Connect API uses OAuth 2.0 and supports two authentication flows:
- (a) authorization codes for human web users; and
 - (b) client credentials for administrative and system-to-system access.
- 2.7. The Connect API defines four access scopes:
- (a) `directory:website`– for web-based user operations;
 - (b) `directory:admin`– for full administrative functions available to super and domain users;
 - (c) `directory:software`– for system-to-system operations;
 - (d) `directory:servicedesk` – for service desk application operations.
- 2.8. Both the Participants API and the Connect API use `x-fapi-interaction-id` as a correlation identifier to support operational traceability across API messages.
- 2.9. The Connect API defines reusable pagination and sorting parameters as set out below:
- (a) `page` – the page number to return of the result set;
 - (b) `size` – the size of the pages to return; and
 - (c) `sort` – the field name to sort.
- 2.10. The Connect API also includes status and free-text filters for several list operations e.g. filter by `AuthorisationServerId` that contain the given string, case insensitive.
- 2.11. The Connect API defines a common set of HTTP error responses, including bad request unauthorised, forbidden, not found, too many requests, internal server error, bad gateway and upstream timeout.
- 2.12. To support operational resilience and predictable behaviour during capacity constraints, the Connect API gateway response templates also include throttling and integration timeout messages:
- (a) **Throttled** - There are no available services to process your request and maximum number of configured processing instances has been reached. Your request is now throttled. Please try again.
 - (b) **Integration timeout** - There are no available services to process your request. This increase in demand is resulting in more resources being initialised to increase capacity. Please try again in 30 seconds.

- 3.1. The CCS Authorisation Server implements the FAPI 2.0 Security Profile and communicates with ATPs and EDHs through secure API endpoints to support the authorisation of Permission-based access to Energy Data. This Paragraph 3 describes how the FAPI 2.0 Security Profile is applied within the CCS and sets out the additional CCS-specific requirements and profile choices applicable to ATPs and EDHs.

Discovery

- 3.2. Communication with the CCS Authorisation Service is based on OpenID Connect Discovery 1.0, building on RFC 8414 (OAuth 2.0 Authorization Server Metadata). Rather than requiring CCS endpoint URLs and configuration to be hard coded:
- (a) ATPs retrieve a discovery document in JSON format, containing the CCS endpoint details, together with the supported Permission types, and signing algorithms; and
 - (b) EDHs retrieve the CCS public signing keys from the discovery document in order to validate Access Token JWTs.
- 3.3. The CCS publishes its service endpoints through the discovery document, including dedicated mutual TLS (mTLS) endpoints where required. As required by the FAPI 2.0 Security Profile, Access Tokens are bound to the ATP certificate used during token issuance, and the same certificate must be presented when those tokens are subsequently used. This supports secure client authentication and token validation.
- 3.4. The CCS discovery document defines the Energy Data Subject identity claims, scopes and related capabilities that may be requested by ATPs. ATPs obtain this information from the discovery document and may request claims either through defined scopes or by requesting specific claims directly. The claims available to an ATP are governed by the authorised capabilities registered for that ATP within the CCS Directory.

JSON Web Key Set

- 3.5. The CCS uses private signing keys represented as JSON Web Keys (JWKs) as defined in JWK RFC 7517, to sign tokens. To verify that a token is genuine, ATPs and EDHs obtain the CCS's public signing keys from a published JSON Web Key Set (JWKS) endpoint. When validating a token, the ATP or EDH selects the public key whose key ID matches the value in the JWT header. The CCS may publish multiple keys simultaneously to support key rotation. The verifier should always select the key whose key ID matches the one in the JWT header when verifying a token's signature.

Permission and Authorisation Mechanism

- 3.6. The CCS uses Rich Authorization Requests (RAR) (RFC 9396) to define Permission for access to Energy Data. Permission is expressed at the level of Permission type and purpose rather than at the level of individual Data Products. The schema for the `authorization_details` attribute of the RAR payload is defined in the Permission

authorisation details. Each entry in the `authorization_details` array represents a Permission request and carries three nested blocks:

- (a) type details block - the dataset-specific details of the Permission request. The structure of this block varies depending on the 'type' e.g. the payload for Metered Data access will differ from that required for Tariff Data access. Where applicable, the type details distinguish between the Permission period and the requested data period. The Permission period defines the period for which access is authorised. The requested data period defines the period to which the Energy Data requested by the ATP relates and may include historical Energy Data. The requested data period shall not extend beyond the end of the Permission period and, for Metered Data, shall not begin before the start of the occupancy period validated through CCS IDV;
 - (b) ATP supplied hints - optional values provided by the ATP to assist CCS IDV and CCS RMP Matching. These values are not authoritative; and
 - (c) CCS enrichment fields – values added by the CCS following CCS IDV and RMP Matching. These values supersede the ATP-supplied hints.
- 3.7. Information from the `authorization_details` payload is used by the CCS to present clear and specific information to the Energy Data Subject when Permission is requested, enabling them to understand precisely what access is being requested before granting Permission.
- 3.8. In addition to defining the structure of individual `authorization_details` entries, the Permission authorisation details defines the validation rules applied by the CCS. These include rules that apply across the `authorization_details` array as a whole, rather than solely to individual entries.
- 3.9. Separately from the Permission payload, an ATP may request claims relating to an Energy Data Subject's identity, such as name and verified address (via the `managed_addresses_claim`), using OpenID Connect claims request parameter ([OpenID Connect Core §5.5](#)). Contact details are not released. The CCS discovery document is the authoritative source for supported scopes, claims and claims parameter support.

ATP Authentication

- 3.10. The CCS uses `private_key_jwt` for ATP authentication: the ATP signs a single use client assertion JWT with its own private key and the CCS verifies the signature against the ATP's registered public key. The CCS retrieves the ATP's public key from the CCS Directory, using the relevant `client_id`. The CCS does not maintain a separate per-client keystore and does not retrieve keys directly from ATP-managed `jwt_keys_uri` endpoints. ATP public keys are managed centrally through the CCS Directory, which provides a consistent mechanism for key registration, rotation and revocation across the ecosystem. As required by the FAPI 2.0 Security Profile, a fresh client assertion must be generated for each authenticated request, including requests to the PAR and token endpoints, with a unique `jti` and appropriate `exp` value. The `aud_claim` within the

client authentication assertion shall contain the CCS issuer identifier as published in the CCS discovery document.

Pushed Authorization Request (PAR)

- 3.11. The CCS requires ATPs to submit Permission requests using OAuth 2.0 Pushed Authorization Requests (RFC 9126) with the request parameters encapsulated within a signed request object in accordance with RFC 9101 (JWT-Secured Authorization Request (JAR)). These parameters are sent directly to the CCS using an authenticated server-to-server POST request. In response, the CCS validates the request and returns a short-lived request uri reference.
- 3.12. As required by the FAPI 2.0 Security Profile, the Pushed Authorization Request includes PKCE code_challenge/code_challenge_method parameters in accordance with RFC 7636, using the S256 method. This enables the CCS to verify during the authorization code exchange that the ATP exchanging the authorization code is the same ATP that initiated the authorisation request.
- 3.13. The signed request object contains the authorisation request parameters associated with the Permission request, including:
 - (a) the aud_claim which shall reflect the CCS issuer identifier URL;
 - (b) the exp_claim which shall have a lifetime of no longer than 10 minutes after the nbf claim; and
 - (c) The nbf_claim that is no more than 10 minutes in the past.
- 3.14. The request object is distinct from the client assertion used for ATP authentication. The client assertion is used by the CCS to authenticate the ATP, whereas the request object contains the details of the authorisation request itself. Both are JSON Web Tokens (JWTs) signed by the ATP using its private key.
- 3.15. The request object may contain OIDC scopes requesting identity information relating to the Energy Data Subject. Where a scope requests a set of claims, e.g. Energy Data Subject information such as name and verified addresses (via the managed_addresses_claim), the CCS shall return the claims associated with that scope in accordance with the OIDC specification.
- 3.16. Energy Data Subject identity claims are requested with the claims request parameter ([OpenID Connect Core 1.0 §5.5](#)) within the signed request object. The claims request parameter enables the ATP to request specific identity claims to be returned in the ID Token.
- 3.17. Within the claims request parameter, a claim shall be marked as essential where the ATP requires the claim to be returned and may be marked as optional where the ATP would prefer, but does not require, the claim to be returned.
- 3.18. The CCS discovery document indicates whether the OIDC claims request parameter is supported. The use of the claims request parameter does not extend an ATP's

entitlement to Energy Data Subject identity information. Any claims released by the CCS shall remain subject to the scopes and authorised capabilities applicable to that ATP.

- 3.19. ATP-supplied Energy Data Subject information hints and Energy Data Subject identity claims serve different purposes. Energy Data Subject information hints are optional values supplied by the ATP to assist with CCS IDV and RMP matching. Such hints are not authoritative and may be superseded by information obtained or verified by the CCS. Energy Data Subject identity claims are identity attributes released by the CCS in accordance with the supported scopes and claims published in the CCS discovery document.
- 3.20. Where an ATP receives an Energy Data Subject identity claim from the CCS, the claim represents information released by the CCS following its verification and authentication processes. ATP-supplied Energy Data Subject information hints do not constitute verified Energy Data Subject identity information. The Permission authorisation details identifies those fields that may be supplied as Energy Data Subject information hints.
- 3.21. A PAR may contain one or more entries within the `authorization_details` parameter. Multiple entries may be used where the Permission request relates to more than one EDH, or the data is being requested for more than one defined purpose. Where a PAR contains multiple entries, the validation rules defined in the Permission authorisation details shall apply to the request as a whole in addition to any rules applying to individual entries.
- 3.22. The PAR and all other interactions with the CCS Authorisation Server shall include an `x-fapi-interaction-id` request header, with its value being a unique UUID for each request ([RFC4122](#)), to help correlate log entries between the client and server.

Authorization Endpoint

- 3.23. On receipt of a valid `request_uri` from the PAR endpoint, the ATP redirects the Energy Data Subject's browser to the CCS authorisation endpoint using only the `client_id` and `request_uri`, as required by the FAPI 2.0 Security Profile. No other parameters are exposed in the browser.
- 3.24. Upon receipt of the authorisation request, the CCS resolves and validates the `request_uri`, authenticates the Energy Data Subject in accordance with the CCS IDV and RMP Matching processes and presents a Permission screen describing the access being requested by the ATP, based on the `authorization_details` contained within the validated PAR.
- 3.25. Where the Energy Data Subject grants Permission, the CSS redirects the browser back to the ATP's registered `redirect_uri` with a short-lived, single-use authorization code and the original state value, in accordance with the FAPI 2.0 Security Profile. The authorization code does not itself grant access to Energy Data and may only be used by the ATP to obtain tokens from the CCS token endpoint. The `code_challenge` established during the PAR process is bound to the authorization code and verified during the token exchange.

- 3.26. An ATP's redirect uris are recorded against it in the CCS Directory. The CCS shall accept only redirect uris so recorded, matched exactly, and shall reject any authorisation request containing a redirect uri that is not recorded in the CCS Directory.
- 3.27. As required by the FAPI 2.0 Security Profile, the redirect response to the ATP shall include an iss parameter identifying the CCS as the issuer in accordance with RFC 9207. The ATP shall verify that the iss value matches the expected CCS issuer before accepting the response. This provides protection against mix-up attacks, where a malicious or misconfigured second authorisation server tricks the ATP into completing a flow with the wrong party.

Token Request

- 3.28. Following receipt of a valid authorization code, the ATP shall exchange the authorization code for tokens by submitting a Token Request, with ID Token issuance ([OpenID Connect Core 1.0 §3.1.3](#)) to the CCS token endpoint in accordance with RFC 6749. The request shall be submitted directly to the CCS using the mTLS endpoint identified in mtls_endpoint_aliases.
- 3.29. The ATP shall authenticate to the CCS using the client authentication mechanism and shall provide the original code_verifier associated with the authorisation request, which the CCS verifies as required by the FAPI 2.0 Security Profile.
- 3.30. In accordance with the FAPI 2.0 Security Profile, the CCS verifies that:
- (a) the authorization code is valid and has not previously been used;
 - (b) the redirect_uri matches the redirect_uri associated with the original authorisation request;
 - (c) the ATP has successfully authenticated; and
 - (d) the PKCE validation requirements referenced in Paragraph 3.12 have been satisfied.
- 3.31. Upon successful validation, the CCS shall invalidate the authorization code and issue an Access Token, A Refresh Token and, in some cases, an ID Token. The CCS shall only issue an ID Token where the original authorisation request included the openid scope.
- 3.32. Every token request shall contain exactly one resource parameter in accordance with RFC 8707. The resource parameter identifies the EDH for which the Access Token is requested. The CCS shall include within the Access Token only those authorised permissions for which the EDH identified by the resource parameter is responsible.
- 3.33. Where the resource parameter does not correspond to any authorised permission associated with the grant, the CCS shall reject the request with an invalid_target error.
- 3.34. The token response issued by the CCS is derived from the validated PAR and the resulting authorisation outcome. Therefore, the authorization_details associated with

the token response shall not be interpreted as a direct reproduction of the contents of the original `authorization_details` payload.

- 3.35. In producing the granted `authorization_details` in the token response, the CCS may:
- (a) add / enrich information established through CCS IDV and RMP Matching activities;
 - (b) validate information supplied by the ATP against Energy Data Subject identity, entitlement, address, Permission terms and renewal information; and
 - (c) remove or supersede ATP-supplied hints where authoritative information has been established by the CCS.
- 3.36. The enrichment fields are established by the CCS. An ATP shall not include an enrichment block in an authorisation request, and the CCS shall reject any authorisation request whose authorisation details include one.

Access Token Content

- 3.37. The Access Token shall be issued as a JWT in accordance with RFC 9068. The `aud_claim` within the Access Token shall identify the Energy Data Holder specified by the resource parameter.
- 3.38. The `authorization_details` returned in the token response and the `authorization_details` contained within the Access Token may differ. The token response represents the Permission granted to the ATP, whereas the Access Token contains only the information required by the EDH to fulfil the authorised request, including the authorised data period where applicable.
- 3.39. The Permission authorisation details shall specify, for each field within `authorization_details`, whether it appears in the token response; in the Access Token; or in both.
- 3.40. Any field-level restrictions associated with a Permission type shall be applied by the relevant EDH, i.e. where the data returned is limited to the fields belonging to that type.
- 3.41. An Access Token issued by the CCS represents authorisation granted by the CCS in respect of a Permission. Access Tokens shall be certificate-bound to, and usable only by, the relevant ATP. Before honouring an Access Token presented by an ATP, the EDH shall validate the Access Token in accordance with the applicable JWT and security requirements.
- 3.42. The EDH shall validate the Access Token in accordance with clause 5.3.4 of the FAPI 2.0 Security Profile. This includes as a minimum:
- (a) the Access Token signature using the CCS public signing key obtained from the CCS JSON Web Key Set (JWKS) in line with RFC 7519;
 - (b) that the Access token has not expired; and

- (c) that certificate confirmation (cnf.x5t#S256) claim aligns with the client certificate presented by the ATP on the mTLS connection in accordance with RFC 8705.
- 3.43. In addition, the EDH shall validate that the aud_claim is valid against the EDH resource identifier registered in the CCS Directory and shall apply field-level redaction so that only the data fields belonging to that Permission type are returned to the ATP.
- 3.44. Each Access Token shall contain a grant_id claim identifying the grant from which the Access Token was issued. The grant_id value shall remain consistent across all Access Tokens and Refresh Token exchanges associated with the same grant. The grant_id is an opaque identifier and shall not contain information that directly identifies an Energy Data Subject.
- 3.45. The token response specifies the token's lifetime via the expires_in value, in second from the time of the response in accordance with RFC 6749. The Access Token itself carries its absolute expiry in the exp claim. This prevents misalignment based on the ATP's clock.
- 3.46. The ID Token may include a RECCo specific managed_addresses field containing an array of verified addresses associated with properties occupied by the Energy Data Subject, each carrying the property address and postcode. Where the managed_addresses claim is provided, it is used in place of the standard OpenID Connect address claim.

Refresh Token Request

- 3.47. A Refresh Token is issued alongside the Access Token. No Refresh Token shall be issued with a validity extending beyond the end of the Permission to which it relates. As a result, allowing the ATP to call the token endpoint with its Refresh Token to obtain a new Access Token, provided the underlying Permission is still valid. If the underlying Permission has expired or been revoked, the request will be rejected (invalid_grant) and the ATP shall restart the process with a new PAR
- 3.48. The Refresh Token request and associated response is server-to-server and requires the same client authentication as the other token endpoint calls. The request includes only one resource parameter e.g. one EDH, with the Access Token content filtered accordingly.
- 3.49. A Refresh Token will be provided for one-time Permissions. The ATP shall not populate the end_date_time for one-time Permissions as the CCS will set the value to 25 hours after the Permission start date. A one-time Permission is not exercised until the requested Energy Data has been received in full, including any provided asynchronously by an EDH.

Token Revocation

- 3.50. Where an Energy Data Subject withdraws Permission within the ATP's own interface, the ATP shall call the revocation endpoint to ensure the CCS ends the grant immediately (RFC 7009). This Token Revocation Request is server-to-server and

requires the same client authentication as the other token endpoint calls.

- 3.51. A revocation_reason must be included, aligning to a reason code defined within the Data Specification. The CCS records the reason code against the grant and issues to the ATP via the Permission Revocation Notification. The revocation reason is not issued to the EDH.
- 3.52. Revoking a Refresh Token ends the grant recorded within the CCS, therefore the CCS will refuse any further Refresh Token Requests. For the avoidance of doubt, it does not invalidate an Access Token already issued in relation to that grant.

Grant Endpoint

- 3.53. The CCS issues a grant_id alongside the tokens associated with a grant. The grant endpoint enables an ATP to receive the current state of that grant as recorded by the CCS, including the Permissions granted, relevant dates and whether the grant remains active. Calls to the grant endpoint are server-to-server over mTLS, presenting the Access Token issued from that grant, which the CCS validated together with the certificate binding.
- 3.54. The authorization_details returned by the grant endpoint shall reflect the enriched and validated Permission information held by the CCS and shall be consistent with the authorization_details returned by token endpoint, rather than the ATPs original copy from the PAR.

Client Credentials Token Request

- 3.55. The Client Credentials Token Request enables an ATP to obtain an Access Token for activities performed in its own capacity and not on behalf of an Energy Data Subject e.g. retrieval of its own registration information or EDH metadata from the CCS Directory, see Paragraph 2. This Client Credentials Token Request is server-to-server and requires the same client authentication as the other token endpoint calls.
- 3.56. The Access Token will not include authorization_details and a Refresh Token will not be issued in response to a Client Credentials Token Request. Upon expiry of the Access Token, the ATP may obtain a new Access Token by submitting a further Client Credentials Token Request.

4 CCS Shared Signals Framework

- 4.1. The CCS Shared Signals Framework uses [OpenID Shared Signals Framework 1.0 \(Final\)](#) and the underlying IETF Security Event Token standards (RFC 8417, RFC 9493, RFC 8935). The framework is used to notify CCS Users that something about the grant has changed.
- 4.2. Under the CCS Shared Signals Framework, all notifications will be delivered by push-based SET Delivery (RFC 8935). A receiver registers in the CCS Directory, the endpoint at which it will take delivery and establishes a stream with the CCS pointing to that endpoint. The CCS calls that endpoint directly with the event and holds the

event until the receiver acknowledges it.

- 4.3. Events are delivered in the form of Security Event Tokens which are JWTs signed by the CCS.
- 4.4. The CCS and each TI Energy Supplier, when acting as a transmitter, shall support the provisions specified for a transmitter in the OpenID Shared Signals Framework 1.0, in particular its event delivery, transmitter configuration discovery and stream management clauses. In addition, it shall:
 - (a) populate notifications based on the receiver's role (conformance class) using its entity identifier;
 - (b) when creating a stream, include all Permissions associated with the authenticated Receiver. ATPs should receive Permissions as a 'Class A Receiver', and EDHs should receive Permissions as a 'Class B Receiver'. Newly created Permissions must be included immediately;
 - (c) authenticate all CCS User messages e.g. stream management, status and verification, by mTLS (RFC8705 pattern) based on the CCS User's identity and conformance class from the entity statement.
 - (d) advertise mutual TLS, in accordance with RFC 8705, as the stream authorisation scheme in its transmitter configuration;
 - (e) return an explicit error where a CCS User requests verification against a paused stream;
 - (f) support x-fapi-interaction-id on all CCS SSF management API requests and echo it on responses; and
 - (g) reject the creation of a stream, or a change to an existing stream where the delivery.endpoint.url does not exactly match an endpoint the receiver has registered in the CCS Directory.
- 4.5. Each CCS User, when acting as a receiver shall:
 - (a) support the provisions specified for a 'receiver' in the OpenID Signals Framework 1.0, in particular its Security Event Token profile and event delivery clauses;
 - (b) validate the Security Event Token signature against the transmitter jwks_uri, independently of transport authentication.
 - (c) reject any Security Event Token whose JOSE typ is not secevent+jwt;
 - (d) validate that iss equals both the stream configuration iss and the discovery issuer, and that aud is the Receiver's own Entity Identifier;
 - (e) process idempotently on jti;
 - (f) process, persist, then acknowledge, in that order;

- (g) register the endpoint at which it takes push delivery in the CCS Directory, against its server (EDH) or its application (ATP), before creating a stream; and
- (h) ensure its `delivery.endpoint_url` is served on a certificate registered within the CCS Directory.

5 Account Holder Verification

- 5.1. CCS Account Holder Verification enables the CCS to verify that an individual requesting access to Energy Data is a legitimate Account Holder for the relevant energy supply account. The verification process is implemented using OpenID Connect (OIDC) and follows a security model aligned with the existing CCS Permission architecture.
- 5.2. When a user initiates a service request that requires Account Holder Verification, the CCS acts as an OIDC client and the Energy Supplier acts as the identity provider (IdP) and authoritative source for Account Holder status. The CCS requests verification from the Energy Supplier and receives a signed identity token (ID Token) confirming the user's association with the Energy Supplier account
- 5.3. The process reuses the existing Pushed Authorization Request (PAR) pattern already established between the ATP and CCS, providing a consistent authentication and Permission framework.
- 5.4. Account Holder Verification is initiated by the CCS through the submission of a PAR to the Energy Supplier's authorisation server, only in response to an explicit Energy Data Subject request. Energy Supplier OpenID Connect Discovery (`.well-known`) shall be discoverable through the CCS Directory.
- 5.5. The Energy Supplier validates the request via mTLS (RFC8705 pattern) based on the CCS User's identity and conformance class from the entity statement. Energy Suppliers shall apply the same PAR (RFC 9126), signed request object (JAR) (RFC 9101) and `private_key_jwt` client authentication mechanisms as applied by the CCS when validating the ATP submitted PAR under Paragraph 3.
- 5.6. The CCS shall only request claims necessary for confirming Account Holder status and associated MPxNs and shall not request Energy Data Subject identity or contact information from the Energy Supplier.
- 5.7. Each Energy Supplier shall:
 - (a) support the authorization code flow with `response_type=code`, returning the authorization code on the redirect and the ID Token from the token endpoint;
 - (b) authenticate the Energy Data Subject and return the requested claims in the ID Token, in accordance with the claims request parameter (OpenID Connect Core 1.0 para 5.5);
 - (c) return the `account_holder_mpxns` claim in the ID Token whether or not the client marks it as essential. This will include the MPxNs for which the authenticated

Energy Data Subject holds an account-holder relationship; and

- (d) set the `sub_claim` to an identifier for the Energy Data Subject that is permanent and stable across authentications and shall not encode or derive it from personally identifiable information.

- 5.8. The CCS shall ensure that an Energy Supplier specific `sub_claim` may be associated with only one CCS account and shall verify that the identifier is not already linked to another CCS account before establishing the association.

6 Errors and Response Codes

- 6.1. For requests to the PAR, token, revocation and client credentials endpoints, the CCS shall return errors in a JSON response body, in accordance with RFC 6749. The response shall include an error parameter and may include an `error_description` parameter where additional information is required.
- 6.2. For requests to the Authorization Endpoint, errors shall be returned by redirecting the Energy Data Subject's browser to the ATP's registered `redirect_uri` and including the error parameter and original state value in the response.
- 6.3. Where the CCS is unable to validate the `client_id` or `redirect_uri`, the CCS shall not redirect the Energy Data Subject's browser to the supplied `redirect_uri`. Instead, the CCS shall display the error directly to the Energy Data Subject or otherwise terminate the request without performing a redirect.
- 6.4. ATPs shall be capable of receiving and processing error responses returned by the CCS in accordance with the relevant OAuth 2.0 endpoint specification.
- 6.5. The following errors will be used:

Error type	HTTP Response	Applicable Endpoints	Context
<code>invalid_request</code>	400	PAR, token, revocation	Malformed request, a missing or repeated required parameter, or a request object that fails to parse or verify,
<code>invalid_client</code>	401	PAR, token, revocation	Client authentication failed — unknown <code>client_id</code> , an invalid <code>client_assertion</code> signature, an expired or replayed <code>jti</code> , or a client certificate that does not correspond to the registered client.
<code>invalid_grant</code>	400	token	The authorization code is expired, already used, or was issued to a different client; the <code>code_verifier</code> does

			not match the code_challenge; or the refresh_token has been revoked or the Permission behind it has expired.
unauthorizedclient	400	token	The client is not permitted to use the requested grant_type.
unsupported_grant_type	400	token	A grant_type the CCS does not support.
invalid_scope	400	PAR, token	A requested scope is unknown or not permitted for this client.
invalid_target	400	token	The resource value is missing, malformed, repeated, or matches no authorization_details entry in the grant.
invalid_request_uri	400	authorisation	The request_uri is unknown, already consumed, or expired. PAR request_uri values are short-lived and single-use.
access_denied	redirect	authorisation	The Energy Data Subject declined Permission or abandoned the journey. Returned with error_description = "End-User aborted interaction".
access_denied	redirect	authorisation	CCS IDV could not be completed for the Energy Data Subject. Returned with error_description = "IDV Failure".
access_denied	redirect	authorisation	CCS RMP Matching could not be completed, or no RMP could be matched to the verified Energy Data Subject. Returned with error_description = "mpxn matching failure".
invalid_token	401	grant	The Access Token is expired or revoked, or its cnf thumbprint does not match the certificate presented on the connection.
-	403	grant	The grant_id is valid but the grant belongs to another ATP.
-	404	grant	No grant exists for the grant_id given.

- 6.6. Successful responses shall use HTTP 200 OK throughout, except PAR which returns 201 Created and CCS SSF push delivery where the EDH returns 202 Accepted.

7 Privacy Considerations

- 7.1. The CCS shall issues a unique grant_id for each grant established between an ATP and an Energy Data Subject.
- 7.2. A grant_id shall be an opaque identifier and shall not contain, reveal or enable the derivation of personally identifiable information relating to the Energy Data Subject.
- 7.3. The CCS shall ensure that the use of a grant_id does not enable the activities of an Energy Data Subject to be correlated across different ATPs.
- 7.4. The sub claim issued by the CCS within identity and authorisation responses shall be an opaque subject identifier. The sub claim shall not comprise an email address, customer identifier or any other value from which the identity of the Energy Data Subject may be directly determined. Where an Energy Data Subject interacts with more than one ATP, the sub claim shall be generated in a manner that does not enable the Energy Data Subject's activities to be correlated across those ATPs.
- 7.5. ATPs and EDHs shall treat grant_id and sub as opaque identifiers and shall not infer or derive Energy Data Subject identity information from those values alone.

Privacy Controls

- 7.6. The CCS, ATPs and EDHs shall implement appropriate technical and organisational measures to protect the confidentiality, integrity and security of Permissions, grants, tokens and Energy Data Subject data throughout the authorisation journey, in accordance with the applicable CCS Arrangements and information security and data protection requirements.
- 7.7. Permission requests shall be communicated and processed using the security controls and protocols required by the FAPI 2.0 Security Profile and specified in this document, including mTLS, PAR, JWT-Secured Authorization Requests (JAR), PKCE, issuer validation and certificate-bound Access Tokens.

Data Minimisation and Purpose Limitation

- 7.8. Permissions shall be expressed using structured authorisation details that clearly identify the purpose and scope of access being requested.
- 7.9. Energy Data disclosed by an EDH shall be limited to the information authorised under the relevant Permission and applicable Permission type.
- 7.10. Information disclosed by the CCS to an ATP or EDH, including identity, address, premises, RMP and Permission information, shall be limited to the minimum information required for that recipient to undertake the relevant authorised activity.

- 7.11. ATP-supplied hints are provided solely to support CCS IDV and RMP Matching. Such hints are not authoritative Energy Data Subject identity information and shall not be propagated to EDHs.
- 7.12. Where the CCS establishes verified Energy Data Subject information through its identity verification, matching or validation processes, such information shall take precedence over ATP-supplied hints.

Revocation and Grant Management

- 7.13. The CCS shall provide mechanisms to support the timely propagation of grant revocation and status changes to relevant ecosystem participants.
- 7.14. EDHs performing activities associated with a grant shall be capable of recognising when a grant has been revoked, expired or otherwise ceased to be valid.

Energy Data Subject Transparency

- 7.15. The CCS shall provide a consistent Permission journey through which an Energy Data Subject is informed of the access being requested, the purposes for which data will be used and the parties with whom data will be shared.
- 7.16. The CCS shall maintain an authoritative record of Permissions and grants established through the CCS, enabling a consistent representation of consumer data-sharing arrangements across the ecosystem.