ETSI TS 119 472-3 V1.1.1 (2026-03)

TECHNICAL SPECIFICATION

Electronic Signatures and Trust Infrastructures (ESI); Profiles for Electronic Attestation of Attributes; Part 3: Profiles for issuance of EAA or PID

Reference
DTS/ESI-0019472-3

Keywords
attribute attestation, digital identity, EUDI Wallet, trust services

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - APE 7112B
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° w061004871

Contents

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The declarations pertaining to these essential IPRs, if any, are publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI IPR online database.

Pursuant to the ETSI Directives including the ETSI IPR Policy, no investigation regarding the essentiality of IPRs, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

DECT™, **PLUGTESTS™**, **UMTS™** and the ETSI logo are trademarks of ETSI registered for the benefit of its Members. **3GPP™**, **LTE™** and **5G™** logo are trademarks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners. **oneM2M™** logo is a trademark of ETSI registered for the benefit of its Members and of the oneM2M Partners. **GSM**® and the GSM logo are trademarks registered and owned by the GSM Association.

Foreword

This Technical Specification (TS) has been produced by ETSI Technical Committee Electronic Signatures and Trust Infrastructures (ESI).

The present document is part 3 of a multi-part deliverable covering profiles for Electronic Attestation of Attributes, as identified below:

Part 1: "General requirements";

Part 2: "Profiles for EAA/PID Presentations to Relying Party";

Part 3: "Profiles for issuance of EAA or PID".

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

1 Scope

The present document specifies a protocol for allowing an EUDI Wallet to request a Personal Identification Data (PID hereinafter) Provider, the issuance of a PID, and to request an Electronic Attestation of Attributes (EAA hereinafter) Provider, the issuance of EAA.

The protocol specified builds on OpenID4VC-HAIP [1], which defines a profile of OpenID4VCI [2].

In addition to that, the protocol extends OpenID4VC-HAIP [1], for meeting requirements imposed by the EUDI Wallet ecosystem as specified in EUDI Wallet ARF [i.1], by EUDI Wallet TS03 [i.2], and by EAA formats, categories, and profiles specified in ETSI TS 119 472-1 [3].

More specifically, the present document defines requirements for:

1) Inclusion of the access certificates issued to PID/EAA Provider in the Issuer Metadata.

2) Inclusion of the registration certificates issued to PID/EAA Provider in the Issuer Metadata.

3) Inclusion of Embedded Disclosure Policy details in the Issuer Metadata.

4) Inclusion of details on the PID/EAA Provider preferences on the method to limit the number of times that a PID/EAA may be presented, in the Issuer Metadata.

5) Sending the Wallet Instance Attestation (WIA hereinafter) to the Pushed Authorisation Endpoint, and the Token Endpoint.

6) Sending the Wallet Unit Attestation (WUA hereinafter) to the Credential Endpoint.

7) Proofing possession of the private key associated to the public key in the WUA.

8) Proofing possession of the private key(s) associated to the public key(s) to be bound to the PID/EAA(s) to be issued.

9) Proofing that the same WSCA/WSCD possesses the private key associated to the public key in the WUA and the private key(s) associated to the public key(s) to be bound to the PID/EAA(s) to be issued.

10) Sending the details of the format(s) of the PID/EAA requested to the Authorisation Endpoint, and the Credential Endpoint.

11) Issuance of EAA in formats that are not covered by OpenID4VCI [2], as EAA built on X.509 Attribute certificates, for instance (X509-AC EAA specified in clause 8 of ETSI TS 119 472-1[3]).

12) Claims path for X509-AC EAA format, not covered by OpenID4VCI [2].

NOTE: Some of the specifications used by the present document are not yet final. Should breaking changes occur in some of the mentioned drafts, they should not be taken into account unless a new version of the present document is generated referencing the updated drafts or final versions.

2 References

2.1 Normative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

Referenced documents which are not found to be publicly available in the expected location might be found in the ETSI docbox.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long-term validity.

The following referenced documents are necessary for the application of the present document.

[1] OpenID4VC-HAIP (24 December 2025): "OpenID4VC High Assurance Interoperability Profile 1.0".

[2] OpenID4VCI (16 September 2025): "OpenID for Verifiable Credential Issuance 1.0".

[3] ETSI TS 119 472-1: "Electronic Signatures and Trust Infrastructures (ESI); Profiles for Electronic Attestation of Attributes; Part 1: General requirements".

[4] ETSI TS 119 475: "Electronic Signatures and Trust Infrastructures (ESI); Relying party attributes supporting EUDI Wallet user's authorization decisions".

[5] IETF draft-ietf-oauth-attestation-based-client-auth-07 (September 2025): "OAuth 2.0 Attestation-Based Client Authentication".

[6] IETF RFC 4514 (June 2006): "Lightweight Directory Access Protocol (LDAP): String Representation of Distinguished Names".

[7] IETF RFC 4648 (October 2006): "The Base16, Base32, and Base64 Data Encodings".

[8] IETF RFC 7515 (May 2015): "JSON Web Signature (JWS)".

[9] OpenID4 VP (9 July 2025): "OpenID for Verifiable Presentations 1.0".

[10] NIST SP 800.38D (November 2007): "Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC".

[11] W3C® Recommendation 15 May 2025: "Securing Verifiable Credentials using JOSE and COSE".

2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long-term validity.

The following referenced documents may be useful in implementing an ETSI deliverable or add to the reader's understanding, but are not required for conformance to the present document.

[i.1] EUDI Wallet ARF: "Architecture and Reference Framework version 2.4.0".

[i.2] EUDI Wallet TS03: "Specification of Wallet Unit Attestations (WUA) used in issuance of PID and Attestations".

[i.3] Commission Implementing Regulation (EU) 2024/2979 of 28 November 2024 laying down rules for the application of Regulation (EU) No 910/2014 of the European Parliament and of the Council as regards the integrity and core functionalities of European Digital Identity Wallets.

[i.4] EUDI Wallet TS05: "Specification of common formats and API for Relying Party Registration information".

[i.5] ETSI TS 119 471: "Electronic Signatures and Trust Infrastructures (ESI); Policy and Security requirements for Providers of Electronic Attestation of Attributes Services".

[i.6] IETF RFC 7519 (May 2015): "JSON Web Token (JWT)".

[i.7] ETSI TS 119 472-2: "Electronic Signatures and Trust Infrastructures (ESI); Profiles for Electronic Attestation of Attributes; Part 2: Profiles for EAA/PID Presentations to Relying Party".

3 Definition of terms, symbols and abbreviations

3.1 Terms

For the purposes of the present document, the terms given in ETSI TS 119 471 [i.5], ETSI TS 119 472-1 [3] and the following apply:

PID/EAA Provider's Registrar: entity in charge of registering the PID/EAA Provider's information necessary to allow for their electronic identification and authentication towards European Digital Identity Wallets

Wallet Instance Attestation (WIA): JWT signed by the Wallet Provider that attests the integrity of the app

NOTE 1: Based on clause 2 of EUDI Wallet TS03 [i.2].

NOTE 2: The WIA allows PID/EAA Providers to protect their endpoints by only communicating with Wallet Applications which integrity are ensured by the Wallet Providers.

NOTE 3: The contents of the WIA are as specified in EUDI Wallet TS03 [i.2].

NOTE 4: JWT is specified in IETF RFC 7519 [i.6].

Wallet Unit Attestation (WUA): JWT signed by the Wallet Provider that describes the components of the Wallet Unit or allows authentication and validation of those components

NOTE 1: The WUA allows PID/EAA Providers to ensure that they only issue PID/EAA that are cryptographically bound to keys that are properly protected (i.e. in a WSCD with sufficiently high attack resistance), as well as to revoke their credentials in case a Wallet Provider revokes a Wallet Unit (EUDI Wallet TS03 [i.2]).

NOTE 2: The contents of the WUA are as specified in EUDI Wallet TS03 [i.2].

3.2 Symbols

Void.

3.3 Abbreviations

For the purposes of the present document, the following abbreviations apply:

ARF	Architecture and Reference Framework
EAA	Electronic Attestation of Attributes
EDP	Embedded Disclosure Policy
EUDI	European Digital Identity
HAIP	High Assurance Interoperability Profile
JWS	JSON Web Signature
JWT	JSON Web Token

PID	Personal Identification Data
PID/EAA Provider	PID or EAA Provider
X509-AC	X.509 Attribute Certificate
WIA	Wallet Instance Attestation
WSCA	Wallet Secure Cryptographic Application
WSCD	Wallet Secure Cryptographic Device
WU	Wallet Unit
WUA	Wallet Unit Attestation

3.4 Notation

The present document assigns one identifier for each requirement.

These identifiers result from the concatenation of the following three components:

1) A topic identifier, for signalling the topic targeted by the requirement.
2) The number of the clause where the requirement is defined.
3) A number of 2 digits. In each clause the number will start in 01 and it will increase in one unity for each requirement.

The following topic identifiers are used:

1) GEN-REQ for general requirements on the issuance flows.
2) ISS-MDATA for general requirements on Issuer Metadata.
3) ISS-MDATA-ACC_CERT for requirements on provision of access certificates in Issuer Metadata.
4) ISS-MDATA-REG_CERT for requirements on provision of registration certificates in Issuer Metadata.
5) ISS-MDATA-EAA-REUSE-POL for requirements for PID/EAA reuse policy.
6) ISS-MDATA-EBD for requirements on provision of Embedded Disclosure Policies.
7) ISS-CRED-OFFER for requirements on the Credential Offer.
8) AUTH-REQ for requirements on the Pushed Authorisation Request.
9) AUTH-REQ-PROC for requirements on processing of the Pushed Authorisation Request.
10) TOKEN-REQ for requirements on the Token Request.
11) TOKEN-REQ-PROC for requirements on processing of the Token Request.
12) CRED-REQ for requirements on the Credential Request.
13) CRED-REQ-PROC for requirements on processing of the Credential Request.
14) CRED-RESP for requirements on Credential Response.
15) NOT-REQ for requirements on Notification Request.
16) CRYPTO for requirements on crypto suites.
17) SEC for security considerations.
18) ISS-MDATA-X509-AC EAA for requirements on paths for identifying attributes in X509-AC EAAs in Issuer Metadata.

4 EAA issuance flows

4.1 General requirements

GEN-REQ-4.1-01: The EUDI Wallet shall implement the profile of the protocol defined in OpenID4VCI [2] specified by the requirements defined for the Wallet in OpenID4VC-HAIP [1], clause 4 and its subclauses, unless specified otherwise in the present document.

GEN-REQ-4.1-02: When PID Providers and the EAA Providers are issuing PIDs and EAAs into the EUDI Wallet, they shall implement the protocol defined in OpenID4VCI [2], specified by the requirements defined for the Issuer in OpenID4VC-HAIP [1], clause 4 and its subclauses, unless specified otherwise in the present document.

GEN-REQ-4.1-03: The EUDI Wallet shall support the following flows: the Authorisation Code Flow as specified in clause 3.4 of OpenID4VCI [2], and the Pre-Authorised Code Flow as specified in clause 3.5 of OpenID4VCI [2].

GEN-REQ-4.1-04: PID/EAA Providers implementing the protocol defined in clause 4 and its subclauses of the present document, shall implement at least one of the following flows: the Authorisation Code Flow as specified in clause 3.4 of OpenID4VCI [2], or the Pre-Authorised Code Flow as specified in clause 3.5 of OpenID4VCI [2].

GEN-REQ-4.1-05: PID Providers implementing the Pre-Authorised Code Flow shall perform user authorisation with the physical presence.

NOTE: As described in OpenID4VCI [2], Pre-Authorised Code Flow does not provide sufficient security against hijacking attacks for remote scenarios and therefore online issuance requiring user authentication with Level of Assurance HIGH is discarded for the Pre-Authorised Code Flow.

GEN-REQ-4.1-06: The custom URL scheme `"eu-eaa-offer://"` shall be used to invoke the EUDI Wallet.

4.2 Requirements for Issuer Metadata (PID/EAA Provider Metadata)

4.2.1 General requirements

ISS-MDATA-4.2.1-01: The Issuer Metadata shall be a signed metadata according to clause 12.2.3 of OpenID4VCI [2] with a JSON Web Signature (JWS hereinafter) specified in IETF RFC 7515 [8].

ISS-MDATA-4.2.1-02: The signing certificate of the signature on the Issuer Metadata shall be the access certificate of the PID/EAA Provider.

ISS-MDATA-4.2.1-03: The set of possible values for the format parameter of `credential_configurations_supported` parameter of the Issuer Metadata shall include one or more of the values defined for the Credential Format Identifier in Annex A of OpenID4VCI [2].

4.2.2 Provision of access certificates of PID/EAA Provider to EUDI Wallet

ISS-MDATA-ACC_CERT-4.2.2-01: The JWS signing the Issuer Metadata shall contain, in its Protected Header, the `x5c` header parameter as specified in IETF RFC 7515 [8].

ISS-MDATA-ACC_CERT-4.2.2-02: The `x5c` header parameter shall contain the base64 (IETF RFC 4648 [7]) encoding of the DER-encoded access certificate of the PID/EAA Provider, which shall also be the signing certificate.

ISS-MDATA-ACC_CERT-4.2.2-03: The `x5c` header parameter should contain the base64 (IETF RFC 4648 [7]) encoding of the DER-encoded certificates in the signing certificate's path until the trust anchor, but excluding it.

4.2.3 Provision of registration certificates of PID/EAA Provider to EUDI Wallet

ISS-MDATA-REG_CERT-4.2.3-01: The Issuer Metadata shall contain the `issuer_info` metadata parameter.

ISS-MDATA-REG_CERT-4.2.3-02: The `issuer_info` parameter shall be placed at the top level of the Signed JWT payload used for retrieving the Issuer Metadata.

ISS-MDATA-REG_CERT-4.2.3-03: The `issuer_info` parameter shall have the same structure as the `verifier_info` parameter specified in clause 5.1 of OpenID4 VP [9].

ISS-MDATA-REG_CERT-4.2.3-04: One of the elements in the `issuer_info` array parameter may contain the PID/EAA Provider's registration certificate.

ISS-MDATA-REG_CERT-4.2.3-05: The value of the `format` member of the element in the `issuer_info` array parameter that contains the PID/EAA Provider's registration certificate shall be `"registration_cert"`.

NOTE 1: This value of the `format` member of the element identifies this element as the container of the PID/EAA Provider's registration certificate.

ISS-MDATA-REG_CERT-4.2.3-06: The value of the `data` member of the element in the `issuer_info` array parameter that contains the PID/EAA Provider's registration certificate shall be the registration certificate of the PID/EAA Provider.

ISS-MDATA-REG_CERT-4.2.3-07: One of the elements of `issuer_info` shall contain the PID/EAA Provider's registration information provided by the PID/EAA Provider's Registrar.

ISS-MDATA-REG_CERT-4.2.3-08: The value of the `format` member of the element in the `issuer_info` array parameter that contains the PID/EAA Provider's registration information shall be `"registrar_dataset"`.

NOTE 2: This value of the `format` member of the element identifies this element as the container of the PID/EAA Provider's registration information provided by the PID/EAA Provider's Registrar.

ISS-MDATA-REG_CERT-4.2.3-09: The value of the `data` member of the element in the `issuer_info` array parameter that contains the PID/EAA Provider's registration information shall be a non-empty JSON Object.

The `data` member of the element in the `issuer_info` array parameter that contains the PID/EAA Provider's registration information:

- ISS-MDATA-REG_CERT-4.2.3-10: Shall contain the `identifier` member specified in clause B.2.2 of ETSI TS 119 475 [4], whose value shall be the PID/EAA Provider's identifier.
- ISS-MDATA-REG_CERT-4.2.3-11: Shall contain the `srvDescription` member specified in clause B.2.2 of ETSI TS 119 475 [4], which shall be an array of JSON MultiLangString objects (defined in clause B.2.6 of ETSI TS 119 475 [4]), whose contents shall be registered user-friendly descriptions of the PID/EAA Provider's service.
- ISS-MDATA-REG_CERT-4.2.3-12: Shall contain the `registryURI` member specified in clause B.2.1 of ETSI TS 119 475 [4], whose value shall be the URI of the PID/EAA Provider's Registrar.
- ISS-MDATA-REG_CERT-4.2.3-13: Shall contain the `providesAttestations` member, which contains information of the attestation types that the PID/EAA Provider intends to issue for Wallet Units.

NOTE 3: For avoidance of confusion on issuance intentions, this element is typically expected to contain same attestation or attestations, or a superset of the attestation types that are communicated to the Wallet Unit in the related Credential Offer message of OpenID4VCI and its `credential_configuration_ids` parameter, if/as a credential offer is used as part of the issuance flow (issuance can also be executed without an offer phase, if desired attestation type is known e.g. out-of-band). The `providesAttestations` tells the Wallet Unit what credentials the issuer is registered for in its national Registrar and can thus always be used for verification of the issuer's registered attestation types, whereas the `credential_configuration_ids` in Credential Issuer Metadata tells the Wallet Unit what credentials are under active offer by the issuer for a given use case.

NOTE 4: See clause 2.1 of EUDI Wallet TS05 [i.4] for the specifications of `providesAttestations` member.

NOTE 5: This information is required by the Wallet Unit to be able to verify the PID/EAA Provider's registration information upon issuance and present this information to the User. Note that in case the PID/EAA Provider uses the services of an intermediary, the `identifier` of requirement ISS-MDATA-REG_CERT-4.2.3-10 will be different from the identifier of the intermediary as included in the intermediary's access certificate.

4.2.4 Provision of PID/EAA reuse policy

4.2.4.1 General requirements

The present clause specifies requirements for a PID/EAA reuse policy, for controlling the number of times that the Wallet Unit may present the PID/EAA to Relying Parties.

ISS-MDATA-EAA-REUSE-POL-4.2.4.1-01: The `credential_metadata` child parameter of `credential_configurations_supported` parameter in the Issuer Metadata may contain the parameter `credential_reuse_policy`, with the details on the preferences of the PID/EAA Provider on the policy that it shall use to limit the number of times that a Wallet Unit may present the PID/EAA.

ISS-MDATA-EAA-REUSE-POL-4.2.4.1-02: Absence of the `credential_reuse_policy` parameter shall be understood as that the PID/EAA Provider does not limit the number of times that a Wallet Unit may present the PID/EAA.

ISS-MDATA-EAA-REUSE-POL-4.2.4.1-03: The `credential_reuse_policy` parameter shall be a JSON Object.

ISS-MDATA-EAA-REUSE-POL-4.2.4.1-04: The `credential_reuse_policy` parameter shall have the `id` member, identifying the PID/EAA reuse policy.

ISS-MDATA-EAA-REUSE-POL-4.2.4.1-05: The `id` member shall be a JSON string.

ISS-MDATA-EAA-REUSE-POL-4.2.4.1-06: The `credential_reuse_policy` parameter may have the `options` member for providing specific details of the reuse policy identified by the `id` member.

ISS-MDATA-EAA-REUSE-POL-4.2.4.1-07: The `options` member shall be a JSON Array of objects, whose contents shall depend on the specific reuse policy.

ISS-MDATA-EAA-REUSE-POL-4.2.4.1-08: If the `id` member has the value `"arf_annex_ii"` then the PID/EAA reuse policy shall be as specified in clause 4.2.4.2 of the present document.

4.2.4.2 ARF pre-defined PID/EAA reuse policy

NOTE 1: The present clause profiles the parameter `credential_reuse_policy` for the case where the PID/EAA reuse policy is as specified in Annex 2 of EUDI Wallet ARF [i.1] (requirement ISSU_37 and related).

ISS-MDATA-EAA-REUSE-POL-4.2.4.2-01: If the `id` member has the value `"arf_annex_ii"` the `options` array member shall be present.

ISS-MDATA-EAA-REUSE-POL-4.2.4.2-02: If the `id` member has the value `"arf_annex_ii"` the element within the `options` member shall contain the member of label `"details"` mapped to an array of strings.

ISS-MDATA-EAA-REUSE-POL-4.2.4.2-03: If the `id` member has the value `"arf_annex_ii"` the array mapped to the label `"details"` shall contain either `"once_only"` or `"limited_time"`.

ISS-MDATA-EAA-REUSE-POL-4.2.4.2-04: If the `id` member has the value `"arf_annex_ii"` the array mapped to the label `"details"` may contain either `"rotating-batch"` or `"per-relying-party"`, or both of them.

NOTE 2: The `"once_only"` value is used to indicate that the Wallet Unit meets the requirements ISSU_43 to ISSU_47 of Annex 2 of EUDI Wallet ARF [i.1].

NOTE 3: The "limited-time" value is used to indicate that the Wallet Unit meets the requirements ISSU_48 to ISSU_50 of Annex 2 of EUDI Wallet ARF [i.1].

NOTE 4: The "rotating-batch" value is used to indicate that the Wallet Unit meets the requirements ISSU_51 to ISSU_54 of Annex 2 of EUDI Wallet ARF [i.1].

NOTE 5: The "per-relying-party" value is used to indicate that the Wallet Unit meets the requirements ISSU_55 to ISSU_57 of Annex 2 of EUDI Wallet ARF [i.1].

ISS-MDATA-EAA-REUSE-POL-4.2.4.2-05: If the id member has the value "arf_annex_ii" and the array mapped to the label "details" contains only one element, the Wallet Unit shall behave as specified by the string contained in this element.

NOTE 6: In this case the value of the element is as specified by requirement ISS-MDATA-EAA-REUSE-POL-4.2.4.2-05 in the present clause.

ISS-MDATA-EAA-REUSE-POL-4.2.4.2-06: If the id member has the value "arf_annex_ii" and the array mapped to the label "details" contains more than one string, the Wallet Unit should follow the method whose string first appears in the mentioned array and the Wallet Unit supports.

EXAMPLE 1: If the array mapped to the label "details" contains ["per-relying-party", "rotating-batch", "once_only"], and the Wallet Unit supports "rotating-batch" and "once_only" then the Wallet Unit would use method "rotating-batch".

ISS-MDATA-EAA-REUSE-POL-4.2.4.2-07: If the id member has the value "arf_annex_ii" and the array mapped to the label "details" contains any of the following values: "once_only", "rotating-batch", or "per-relying-party", then the JSON Object that contains this array mapped to the label "details" shall also contain a JSON number mapped to the label "batch_size".

ISS-MDATA-EAA-REUSE-POL-4.2.4.2-08: The JSON number mapped to the label "batch_size" shall specify the maximum array size for the proofs parameter in a Credential Request and shall be 2 or greater.

ISS-MDATA-EAA-REUSE-POL-4.2.4.2-09: If the id member has the value "arf_annex_ii" and the array mapped to the label "details" contains the value "once_only", then the JSON Object that contains this array mapped to the label "details" shall also contain a JSON number mapped to the label "reissue_trigger_unused".

ISS-MDATA-EAA-REUSE-POL-4.2.4.2-10: The JSON number mapped to the label "reissue_trigger_unused" shall specify the lower limit for the number of unused PIDs/EAAs held by the Wallet Unit.

ISS-MDATA-EAA-REUSE-POL-4.2.4.2-11: The JSON number mapped to the label "reissue_trigger_unused" shall be lower than the value of the JSON Number mapped to the label "batch_size".

NOTE 7: According to Annex 2 of EUDI Wallet ARF [i.1], once the Wallet Unit reaches this lower limit, it requests the issuance of a new batch of PIDs/EAAs.

ISS-MDATA-EAA-REUSE-POL-4.2.4.2-12: If the id member has the value "arf_annex_ii" and the array mapped to the label "details" contains any of the following values: "limited-time", "rotating-batch", or "per-relying-party", then the JSON Object that contains this array mapped to the label "details" shall also contain a JSON number mapped to the label "reissue_trigger_lifetime_left".

ISS-MDATA-EAA-REUSE-POL-4.2.4.2-13: The JSON number mapped to the label "reissue_trigger_lifetime_left" shall specify the number of seconds before the expiration of the PID/EAA.

NOTE 8: According to Annex 2 of EUDI Wallet ARF [i.1], once the instant indicated by the mentioned number of seconds before the expiration of the PID/EAA arrives, the Wallet Unit requests a new issuance of a new PID/EAA.

EXAMPLE 2: Below follows a non-normative example for the type "once_only".

```
{
  "credential_reuse_policy" : {
    "id": "arf_annex_ii",
    "options": [
      {
        "details": ["once_only"],
        "batch_size": 10,
        "reissue_trigger_unused": 2
      }
    ]
  }
}
```

EXAMPLE 3: Below follows a non-normative example for the type "limited-time".

```
{
  "credential_reuse_policy" : {
    "id": "arf_annex_ii",
    "options": [
      {
        "details": ["limited-time"],
        "reissue_trigger_lifetime_left": 86400
      }
    ]
  }
}
```

EXAMPLE 4: Below follows a non-normative example for the type "rotating-batch".

```
{
  "credential_reuse_policy" : {
    "id": "arf_annex_ii",
    "options": [
      {
        "details": [rotating-batch"],
        "batch_size": 50,
        "reissue_trigger_lifetime_left": 86400
      }
    ]
  }
}
```

EXAMPLE 5: Below follows a non-normative example for the type "per-relying-party".

```
{
  "credential_reuse_policy" : {
    "id": "arf_annex_ii",
    "options": [
      {
        "details": ["per-relying-party"],
        "batch_size": 10,
        "reissue_trigger_lifetime_left": 86400
      }
    ]
  }
}
```

ISS-MDATA-EAA-REUSE-POL-4.2.4.2-14: If both the `batch_credential_issuance` (specified in OpenID4VCI [2]) and the `credential_reuse_policy` are present, then the `batch_credential_issuance` shall be ignored.

4.2.5 Provision of Embedded Disclosure Policy

4.2.5.1 Introduction (informative)

As specified in [i.3] 'embedded disclosure policy' means a set of rules, embedded in an electronic attestation of attributes by its provider, that indicates the conditions that a wallet-relying party has to meet to access the electronic attestation of attributes.

In line with [i.3] it is required that:

1) Wallet providers ensure that electronic attestations of attributes with common embedded disclosure policies set out in Annex III of [i.3] can be processed by the wallet units that they provide.

2) Wallet instances are able to process and present such embedded disclosure policies referred to in paragraph 1 (above) in conjunction with data received from the requesting wallet-relying party.

3) Wallet instances verify whether the wallet-relying party complies with the requirements of the embedded disclosure policy and inform the wallet user of the result.

An embedded disclosure policy is meta data associated with an EAA when issued to a wallet holder either by including the content of the policy itself or by retrieving the policy from a pre-loaded policy definition.

The embedded disclosure policy is applied by wallet to control the disclosure of an EAA including selective disclosure of attributes within an EAA.

This policy may be loaded into a wallet using the EAA issuer protocol as specified in the present document.

The embedded disclosure policy is not revealed to the relying party through the relying party presentation protocol as specified in ETSI TS 119 472-2 [i.7].

NOTE: The EAA rule book informs the relying part of any access restrictions in place. When accessing a particular EAA it is expected that the relying party takes this into account.

Any further access control policy management controls are outside the scope of the present document.

4.2.5.2 Requirements for the data model of Embedded Disclosure Policy

ISS-MDATA-EBD-4.2.5.2-01: The Embedded Disclosure Policy shall be identified by a unique URI.

ISS-MDATA-EBD-4.2.5.2-02: The Embedded Disclosure Policy may be accessible via this URI.

ISS-MDATA-EBD-4.2.5.2-03: The Embedded Disclosure Policy associated with EAA shall be identified by including its unique URI. The EAA provider shall either include the Embedded Disclosure Policy identifier with the data set for the Embedded Disclosure Policy or just provide the identifier if the policy data set has already been pre-loaded into the Wallet Unit.

ISS-MDATA-EBD-4.2.5.2-04: The Embedded Disclosure Policy may include a description of the applicability of an embedded disclosure to a particular community and/or class of application with common security requirements.

ISS-MDATA-EBD-4.2.5.2-05: The Embedded Disclosure Policy may include an identifier of the authority responsible for the policy.

ISS-MDATA-EBD-4.2.5.2-06: The Embedded Disclosure Policy may indicate that no policy restrictions apply for the associated EAA.

ISS-MDATA-EBD-4.2.5.2-07: The Embedded Disclosure Policy may contain a list of authorised relying parties for the associated EAA:

a) as identified by their subject distinguished name as held in the wallet-relying party access certificate in the form of LDAP string as specified in IETF RFC 4514 [6]; and/or

NOTE 1: The LDAP representation of the LDAP string represents those elements of the identification attributes as listed in ETSI TS 119 475 [4], clauses 5.1.2 and 5.1.4 that are in the X.509 subject distinguished name as required to distinguish a specific relying party. For legal person this is `commonName`, `organizationName`, `organizationIdentifier`, `countryName`. For natural person this is `commonName`, `givenName`, `surname`, `serialNumber`, `countryName`.

NOTE 2: `organizationIdentifier` attribute type is represented by the LDAP string "ORGID". `serialNumber` attribute type is represented by the LDAP string "SN".

b) as indicated by the URI encoded entitlements required of relying parties as specified in ETSI TS 119 475 [4], and held in the wallet relying party registration certificate.

NOTE 3: Other equivalent forms of URI encoded authorisations can be provided in the wallet-relying party access certificate but this may be ignored by the Wallet Unit.

ISS-MDATA-EBD-4.2.5.2-08: The Embedded Disclosure Policy may define a specific list of roots of trust, to indicate that EUDI Wallet users should only disclose specific EAAs to Relying Parties in possession of access certificates derived from one of these roots or intermediate certificates.

ISS-MDATA-EBD-4.2.5.2-09: Each element of the list of roots of trust, as specified in ISS-MDATA-EBD-4.2.5.2-08 above, shall include the issuer's distinguished name in the form of LDAP string as specified in IETF RFC 4514 [6] and the issuer's certificate serial number.

ISS-MDATA-EBD-4.2.5.2-10: Other information may be included in an Embedded Disclosure Policy Extension which may be ignored by the wallet.

ISS-MDATA-EBD-4.2.5.2-11: Wallet Units should still be able to cope with extensions being present even if they are ignored.

ISS-MDATA-EBD-4.2.5.2-12: An Embedded Disclosure Policy Extension may be defined which contains alternative policy rules (no policy as in ISS-MDATA-EBD-4.2.5.2-05, authorised relying party only as in ISS-MDATA-EBD-4.2.5.2-06 or specific roots of trust as in ISS-MDATA-EBD-4.2.5.2-07) to be applied to specified attributes within the EAA which are allowed to be disclosed using selective disclosure.

ISS-MDATA-EBD-4.2.5.2-13: The Embedded Disclosure Policy should contain a link to a website of the Attestation Provider explaining the disclosure policy in layman's terms for display by the Wallet Unit to the User and to allow the User to navigate to the website.

4.3 Requirements for the Credential Offer

ISS-CRED-OFFER-4.3-01: PID/EAA Providers shall support at least one of the following Grant Types: `"authorisation_code"` or `"urn:ietf:params:oauth:grant-type:pre-authorised_code"`, as defined in clause 4.1.1 of OpenID4VCI [2].

4.4 Requirements for the Pushed Authorisation Request

4.4.1 Scope of application

AUTH-REQ-4.4.1-01:The requirements defined in clause 4.4 of the present document, and its subclauses, shall only apply to the PID/EAA Providers that implement the Authorisation Code Flow as specified in clause 3.5 of OpenID4VCI [2].

4.4.2 Requirements for contents of the Pushed Authorisation Request

AUTH-REQ-4.4.2-01: The Pushed Authorisation Request shall include the Wallet Instance Attestation (WIA) as an `OAuth-Client-Attestation` parameter, as specified in Appendix E of OpenID4VCI [2], and IETF draft-ietf-oauth-attestation-based-client-auth-07 [5].

NOTE 1: The contents of the WIA are specified in section 2.3.3 of EUDI Wallet TS03 [i.2].

AUTH-REQ-4.4.2-02: The Pushed Authorisation Request shall include a Proof-of-Possession of the public key included within the `cnf` claim, as specified in Appendix E of OpenID4VCI [2], and IETF draft-ietf-oauth-attestation-based-client-auth-07 [5].

NOTE 2: According to EUDI Wallet TS03 [i.2], WIAs are expected to have a time-to-live of less than 24 hours; i.e. the difference between expiration time and the time of issuance is expected to be less than 24 hours.

4.4.3 Requirements for processing the Pushed Authorisation Request

AUTH-REQ-PROC-4.4.3-01: When a PID/EAA Provider receives a WIA, then it shall check that the signature of the JWT verifies under the Wallet Provider's public key as found on the Trusted List of Wallet Providers.

AUTH-REQ-PROC-4.4.3-02: When a PID/EAA Provider receives a WIA, then it shall check that it has not expired.

AUTH-REQ-PROC-4.4.3-03: When a PID/EAA Provider receives a WIA with a Proof-of-Possession, then it shall check that the signature of the Proof-of-Possession verifies under the public key present in the `cnf` claim of the WIA.

4.5 Requirements for the Token Request

4.5.1 Requirements for contents of the Token Request

TOKEN-REQ-4.5.1-01: The Token Request shall include the Wallet Instance Attestation (WIA) as an `OAuth-Client-Attestation` parameter, as specified in Appendix E of OpenID4VCI [2], and IETF draft-ietf-oauth-attestation-based-client-auth-07 [5].

NOTE: The contents of the WIA are specified in section 2.3.3 of EUDI Wallet TS03 [i.2].

TOKEN-REQ-4.5.1-02: The Token Request shall include a Proof-of-Possession of the public key included within the `cnf` claim, as specified in Appendix E of OpenID4VCI [2], and IETF draft-ietf-oauth-attestation-based-client-auth-07 [5].

4.5.2 Requirements for processing the Token Request

TOKEN-REQ-PROC-4.5.2-01: When a PID/EAA Provider receives a WIA, then it shall check that the signature of the JWT verifies under the Wallet Provider's public key as found on the Trusted List of Wallet Providers.

TOKEN-REQ-PROC-4.5.2-02: When a PID/EAA Provider receives a WIA, then it shall check that it has not expired.

TOKEN-REQ-PROC-4.5.2-03: When a PID/EAA Provider receives a WIA with a Proof-of-Possession, then it shall check that the signature of the Proof-of-Possession verifies under the public key present in the `cnf` claim of the WIA.

TOKEN-REQ-PROC-4.5.2-04: Refresh tokens for credential refresh may be supported.

NOTE: When PID/EAA Providers choose to issue refresh tokens that can be used to generate new credentials, they need to carefully assess the associated security implications. This assessment is expected to consider factors such as the type of credentials being issued and the sensitivity of the attributes within. Providers also need to take into account that they cannot make assumptions about how or where the EUDI Wallet stores the refresh tokens.

4.6 Requirements for the Credential Request

4.6.1 Requirements for contents of the Credential Request

4.6.1.1 Common requirements

CRED-REQ-4.6.1.1-01: If the PID/EAA is cryptographically bound to the device, the Credential Request shall include the `proofs` parameter.

CRED-REQ-4.6.1.1-02: The `proofs` parameter shall contain either the `jwt` child member or the `attestation` child member.

NOTE 1: Using `proofs->jwt`, the WU sends proof of possession of private key(s) corresponding to public key(s) that the PID/EAA will be bound to in the Credential Request.

NOTE 2: Using `proofs->attestation`, the WU sends attestation(s) of public key(s) that the PID/EAA will be bound to in the Credential Request, without proof of possession of the corresponding private key(s).

4.6.1.2 Using proofs->jwt mechanism

CRED-REQ-4.6.1.2-01: The `jwt` JSON array member shall have 1 element.

CRED-REQ-4.6.1.2-02: The element in the `jwt` member shall contain the `nonce` claim in their corresponding JWT Bodies, obtained from the Nonce Endpoint of the PID/EAA Provider.

CRED-REQ-4.6.1.2-03: The element in the `jwt` array member shall contain the `key_attestation` parameter in the Protected Header.

CRED-REQ-4.6.1.2-04: The `key_attestation` parameter of the element in the `jwt` array member shall be the WUA, signed by the Wallet Provider.

CRED-REQ-4.6.1.2-05: The WUA shall be a key attestation in JWT format as specified in clause D.1 of OpenID4VCI [2].

NOTE 1: The contents of the WUA are specified in section 2.3.4 of EUDI Wallet TS03 [i.2].

CRED-REQ-4.6.1.2-06: The `attested_keys` claim in the `key_attestation` parameter of the element in the `jwt` array, shall contain one or more public keys owned by the WU.

CRED-REQ-4.6.1.2-07: The element in the `jwt` array member shall be signed by the Wallet Unit using the private key corresponding to the public key present in the first element of the `attested_keys` claim in the `key_attestation` parameter (the WUA).

NOTE 2: Taken from the method of transport for the WUA in EUDI Wallet TS03 [i.2].

NOTE 3: Meeting the former requirements, the WU proves possession of the private key corresponding to the first public key in the `attested_keys` array claim in the WUA, attested by the Walled Provider.

4.6.1.3 Using proofs->attestation mechanism

CRED-REQ-4.6.1.3-01: The `attestation` array member shall contain only one element.

CRED-REQ-4.6.1.3-02: The element in the `attestation` array member shall contain the `nonce` claim in its JWT Body, obtained from the Nonce Endpoint of the PID/EAA Provider.

CRED-REQ-4.6.1.3-03: The element in the `attestation` array member shall be the WUA.

NOTE: The element in the `attestation` array member does not prove possession of the private keys corresponding to the attested public keys.

4.6.2 Requirements for processing the Credential Request

4.6.2.1 Requirements for processing Credential Request with proofs->jwt

CRED-REQ-PROC-4.6.2.1-01: The PID/EAA Provider shall verify that the signature of the `key_attestation` parameter in the element of the `jwt` array member (the WUA), verifies under the Wallet Provider's public key as found on the Trusted List for Wallet Providers.

CRED-REQ-PROC-4.6.2.1-02: The PID/EAA Provider shall verify that the signature of the element of the `jwt` array member verifies under the public key contained in the first position of the `attested_keys` array within the `key_attestation` parameter.

CRED-REQ-PROC-4.6.2.1-03: The PID/EAA Provider shall verify that the element of the `jwt` array member includes the `nonce` claim in its JWT Body with a valid nonce value from its Nonce endpoint.

CRED-REQ-PROC-4.6.2.1-04: The PID/EAA Provider shall generate as many PIDs/EAAs as elements present in the `attested_keys` array within the `key_attestation` parameter.

CRED-REQ-PROC-4.6.2.1-05: Each PID/EAA shall be bound to one of the public keys present in the `attested_keys` array within the `key_attestation` parameter.

CRED-REQ-PROC-4.6.2.1-06: There shall not be two PIDs/EAAs bound to the same public key.

4.6.2.2 Requirements for processing Credential Request with proofs->attestation

CRED-REQ-PROC-4.6.2.2-01: The PID/EAA Provider shall verify that the signature of the element of the `attestation` array member (the WUA) verifies under the Wallet Provider's public key as found on the Trusted List for Wallet Providers.

CRED-REQ-PROC-4.6.2.2-02: The PID/EAA Provider shall generate as many PIDs/EAAs as elements present in the `attested_keys` array within the element of the `attestation` array member.

CRED-REQ-PROC-4.6.2.2-03: Each PID/EAA shall be bound to one of the public keys present in the `attested_keys` array within the element of the `attestation` array member.

CRED-REQ-PROC-4.6.2.2-04: There shall not be two PIDs/EAAs bound to the same public key.

4.7 Requirements for the Notification Request

NOT-REQ-4.7-01: If the EUDI Wallet supports notifications, then it shall send one or more Notification Requests per `notification_id` value received from the PID/EAA Provider.

5 Crypto suites

CRYPTO-5-01: In addition to the crypto suites specified in OpenID4VC-HAIP [1], the EUDI Wallet and the PID/EAA Providers shall support the A128GCM algorithm and the A256GCM algorithm specified in NIST SP 800.38D [10].

6 Security considerations

SEC-6-01: The security considerations in clause 13 of OpenID4VCI [2] shall apply.

SEC-6-02: The privacy considerations in clause 15 of OpenID4VCI [2] shall apply.

Annex A (normative): Requirements for issuance of X509-AC EAAs

A.1 Introduction

The present annex defines specific requirements for issuing X509-AC EAAs, which have been specified in clause 8 of ETSI TS 119 472-1 [3].

A.2 Requirements for Issuer Metadata

ISS-MDATA-A.2-01: If the PID/EAA Provider is able to issue X509-AC EAAs, then the set of possible values for the format parameter of `credential_configurations_supported` parameter of the Issuer Metadata shall include the value `"x509_attr"`, which shall be used for identifying a X509-AC EAA as specified in ETSI TS 119 472-1 [3].

A.3 Rules for path in Issuer Metadata for X509-AC EAAs

The present annex specifies rules for building paths for identifying attributes in X509-AC EAAs in Issuer Metadata.

ISS-MDATA-X509-AC EAA-A-01: For X509-AC EAAs, the attributes in the EAA shall be identified by the `path` member within the `claims` member of the `credential_metadata` member.

ISS-MDATA-X509-AC EAA-A-02: The `path` member shall be an array of 2 or 3 strings, where:

1) The first one shall identify the type of EAA.
2) The second one shall contain the OID of the attribute, represented as a sequence of integer numbers separated by the '.' character.
3) The third one, if present, shall contain the name of the attribute.

Annex B (normative): Requirements for issuance of JSON-LD W3C-VC EAAs secured with enveloping proofs

B.1 Introduction

The present annex defines specific requirements for issuing JSON-LD W3C-VC EAA secured with enveloping proofs, which have been specified in clauses 3.1.1 and 3.2.1 of W3C VC_JOSE_COSE [11], and further profiled in clause 7 of ETSI TS 119 472-1 [3].

B.2 Requirements for Issuer Metadata

ISS-MDATA-B.2-01: If the PID/EAA Provider is able to issue JSON-LD W3C VC JOSE EAAs, defined in clause 3.1.1 of W3C VC_JOSE_COSE [11] and further profiled in ETSI TS 119 472-1 [3], then the set of possible values for the format parameter of `credential_configurations_supported` parameter of the Issuer Metadata shall include the value `"vc+jwt"`.

NOTE 1: For more details on the specific profiles, see clause 7.6.4.2 of ETSI TS 119 472-1 [3].

ISS-MDATA-B.2-02: If the PID/EAA Provider is able to issue JSON-LD W3C VC SD-JWT EAAs, defined in clause 3.2.1 of W3C VC_JOSE_COSE [11] and further profiled in ETSI TS 119 472-1 [3], then the set of possible values for the format parameter of `credential_configurations_supported` parameter of the Issuer Metadata shall include the value `"vc+sd-jwt"`.

NOTE 2: For more details on the specific profiles, see clause 7.6.4.3 of ETSI TS 119 472-1 [3].

Annex C (informative): Change history

Date	Version	Information about changes
8/9/2025	V0.0.1	First complete version
6/10/2025	V0.0.2	Submission of WIA and WUA to their respective Endpoints redefined.
17/10/2025	V0.0.3	Dispositions to comments raised to v0.0.2 implemented. Among others: - Registration certificate in `issuer_info` (same structure as `verifier_info` in OpenID4 VP) - Rework of transport of WIA for better alignment to TS-03 - Rework of transport of WUA - Allowing of pre-Authorised code flow - Requirement for refresh tokens: may instead should of HAIP
30/10/2025	V0.0.4	Last agreements reached implemented. Among others: - Two proofs methods: `jwt` and `attestation`. - `proofs->jwt`: first element WUA. Other elements signed with private keys corresponding to public keys in WUA. - New structure for supporting PID/EAA reuse. Profile of this structure for policy specified in Annex II of ARF. - Ephemeral WIA dropped from Authorisation End point and Token End point. - Dropped `credential_format_id`. - Dropped clause on requirements on EDP in Credential Response
3/11/2025	V0.0.5	Implemented editorial changes as a result of comments by Technical Officer to v0.0.4
25/11/2025	V0.0.6	Implemented dispositions to comments arrived from BDR, IDNow, CHTL, and EC.
10/12/2025	V0.0.7	Implemented some missing agreements with EC (comments arrived in two rounds and the agreements for the comments in the second round were not included in v0-0.6): - Do not restrict user authorisation in pre-authorised code flow to physical presence. The comment said: "Note that OpenId4VCI allows the use of OTP (one-time-password) in case of Pre-Authorised Code flow, delivered via a separate channel, to further protect such scenarios. It was agreed to also allow authentication with a Level of Assurance HIGH. - Drop the mandatory requirement that the PoP of WIA has the challenge parameter: this is not a mandatory requirement in TS3. According to TS3 "it's up to the Credential Issuer to require challenge by providing a challenge end point or not. Implemented agreements reached in adhoc call of 9/12/2025 on comments by Idakto. 1) Drop former GEN-REQ-4.1-06 (duplication of OIDF4VCI). 2) Reworded ISS-MDATA-4.2.1-03 as detailed in the dispositions of comments. 3) Reworded ISS-MDATA-REG_CERT-4.2.3-02 for clarifying where the `issuer_info` element is placed when the Issuer Metadata is transferred using a signed JWT. 4) ISS-MDATA-REG_CERT-4.2.3-08: reworded, and a new ISS-MDATA-REG_CERT-4.2.3-09 is added so that the `issuer_info` either contains the PID/EAA Provider's registration certificate or an element containing the PID PID/EAA Provider's registration information. 5) Added definition of PID/EAA Provider's Registrar in clause 3.1. 6) Replace the old ISS-MDATA-REG_CERT-4.2.3-15 (ISS-MDATA-REG_CERT-4.2.3-16 in v0.0.7 because the insertion of a new requirement as detailed in bullet 4 above) by a requirement that requires the presence of `providesAttestations` member as specified in EUDIW TS05. 7) Dropped `allow` element in `credential_reuse_policy` element.

Date	Version	Information about changes
15/12/2025	V0.0.8	a) Implemented agreement that there is only ONE element in jwt array, which contains the WUA in the credential request. This agreement was not implemented in v0.0.7 by mistake. This has implied deletion of the clause specifying elements of jwt array different than the first one, and also deletion of the clause specifying the processing of such elements. b) Implemented a fix for a last-moment gap identified by EC team in a message received on 11/12/2025. The gap affected PID/EAA reuse policy as specified in Annex 2 of ARF (clause 4.2.4.2 of the present document), and was as follows: 1) When the policy is "once_only" an indication is needed of, a lower limit for the number of unused PIDs/EAAs held by the WU, because once reached this limit, the WU requests the issuance of a new batch of PIDs/EAAs. This indication is a JSON Number mapped to the label "reissue_trigger_unused". 2) When the policy is "limited-time", "rotating-batch", or "per-relying-party", then an indication is needed of the number of seconds before the expiration of the PID/EAA because, once arrived this moment, the WU requests the issuance of a new PID/EAA. c) Reverted an agreement reached at the ad hoc call, because of a review of ARF annex 2. The agreement was that the issuer_info in the Issuer Metadata would contain EITHER the Registration Certificate OR the registration information from the Registrar. EC team claims that after a new reading of the HLRs of ARF Annex 2, the Registration information has to be always present regardless the presence of a Registration Certificate. d) Fixing two editorial issues: wrong numbering of notes in clause 4.2.3, and fixing the wording of Note 2 in clause 4.2.5.2.
16/12/2025	V0.0.9	Added requirement that "the `data` member of the element in the `issuer_info` array parameter that contains the PID/EAA Provider's registration information shall contain the `entitlement` member, which contains the entitlements of the RP" in clause 4.2.3
17/12/2025	V0.0.10	Implemented the following changes due to the agreements reached at ESI#88 plenary: 1) Remove the option of perform user authorisation on-line with authentication with LoA HIGH in the Pre-Authorised Code flow (Requirement GEN-REQ-4.1-05), and add a note on this. 2) Remove the `entitlement` member in `issuer_info` element's `data` member. 3) Remove the `trade_name` member in `issuer_info` element's `data` member
17/12/2025	V0.0.11	NOTE on Pre-Authorised Code Flow reworded to highlight hijacking risk of online authorisations. All the references to Annex 2 of EUDI Wallet ARF [i.1] have been moved to NOTES (as it is an informative reference).
15/1/2026	V0.0.12	Implemented dispositions for all the comments reached to v0.0.11 during the RC

History

Version	Date	Status
V1.1.1	March 2026	Publication