ETSI TS 119 478 V1.1.1 (2026-01)

TECHNICAL SPECIFICATION

Electronic Signatures and Trust Infrastructures (ESI);
Specification of interfaces related to Authentic Sources

Reference
DTS/ESI-0019478

Keywords
API, authentication, authorization, EUDI Wallet,
smart interface

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).

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.

Executive summary

Article 45e of the amended eIDAS-Regulation (EU) No 910/2014 [i.33] obliges the EU Member States to ensure from 24 December 2026 onwards, that qualified trust service providers are able to verify at least the attributes listed Annex VI of the regulation, wherever those attributes rely on authentic sources within the public sector, at the request of the user. The present document specifies the interface for this purpose.

Furthermore, the present document also specifies a second interface, which allows to retrieve additional attributes of the user upon request, if supported by the involved authentic sources.

The present document specifies the necessary technical framework for the required access of the involved authentic sources, which includes architectures, components, interfaces, protocols and data structures.

1 Scope

Article 45e of the amended eIDAS-Regulation (EU) No 910/2014 [i.33] obliges the EU Member States to ensure until 24 December 2026, that qualified trust service providers are able to verify at least the attributes listed Annex VI of the Regulation, wherever those attributes rely on authentic sources within the public sector, at the request of the user.

Article 9 of CIR (EU) 2025/1569 [i.11] provides more information with respect to the verification mechanism, which can involve single points of verification for the attributes listed in Annex VI. Such a verification access point receives the attributes and the identification data of the subject of the attribute as input (see Article 9 (3) of [i.11]) and returns a verification result, which exclusively states whether the attribute under consideration has been verified or not, and the public sector body responsible for the authentic source or, where applicable, the public sector body designated to act on behalf of the authentic source against which the attribute has been verified (see Article 9 (4) of [i.11]).

According to Article 10 (1) of CIR (EU) 2025/1569 [i.11], the "Member States may refer to and re-use the common services of the technical system set out in Article 14 of regulation (EU) 2018/1724 [i.35], as well as the national components connected to them". In a similar manner, according to Article 10 (2) of CIR (EU) 2025/1569 [i.11], "the European Commission shall refer to and re-use, where appropriate, the common services of the technical system pursuant to regulation (EU) 2018/1724 [i.35]". To be able to profit from the functionality of these services, the present document introduces in clause 5 the **Discover Interface**, which allows to access the semantic repository and the data services directory within the common services of the OOTS. The **I1 (Discover)** interface described in clause 5 is an HTTP-based interface according to ISO 15000-3 [24], as profiled by the European Commission in [i.16]. Annex A of the present document contains an OpenAPI specification [27] for this interface.

The present document specifies in clause 6 the **Authentic Source Interface**. The **I2 (Verify)** interface allows to verify attributes against the content stored in an authentic source as requested in Article 45e of the amended eIDAS-Regulation (EU) No 910/2014 [i.33]. In addition, an optional **I3 (Retrieve)** interface is specified, which allows to retrieve attributes from an authentic source, or an intermediary acting on behalf the authentic source.

The Authentic Source Interface specified in the present document contains in clause 6.1 an **HTTP-based** Application Programming Interface (API), which corresponding OpenAPI specification [27] is provided in Annex B of the present document. Furthermore, the present document also specifies in clause 6.2 a second **ISO 15000-based** interface, which utilizes the electronic delivery mechanisms according to ISO 15000-2 [23].

Moreover, according to Article 9 (5) of [i.11], the "Member States may impose access controls or other verification mechanisms that provide integrity, authenticity, and confidentiality to determine that the requester is a qualified trust service provider and is acting at the request of a legitimate user". For the HTTP-based API, clause 6.1.3 specifies such an authorization mechanism based on the OAuth 2.0 authorization framework according to IETF RFC 6749 [6] and related standards. For the ISO 15000-based interface, the qualified trust service status is verified and the trust service provider is registered as client in advance, under procedures out of scope of the present specification. In this case the evidence that the client is acting at the request of a legitimate user is also gathered before the request under procedures out of scope of the present specification.

The present document specifies the necessary technical framework for the required access of the involved authentic sources, which includes architectures, components, interfaces, protocols and data structures.

Note, that policy and security requirements for the involved resource service endpoints are beyond the scope of the present document.

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] IETF RFC 1738: "Uniform Resource Locators (URL)".

[2] IETF RFC 3986: "Uniform Resource Identifier (URI): Generic Syntax".

[3] IETF RFC 4648: "The Base16, Base32, and Base64 Data Encodings".

[4] IETF RFC 5141: "A Uniform Resource Name (URN) Namespace for the International Organization for Standardization (ISO)".

[5] IETF RFC 5280: "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile".

[6] IETF RFC 6749: "The OAuth 2.0 Authorization Framework".

[7] IETF RFC 6819: "OAuth 2.0 Threat Model and Security Considerations".

[8] IETF RFC 6838: "Media Type Specifications and Registration Procedures".

[9] IETF RFC 7519: "JSON Web Token (JWT)".

[10] IETF RFC 7591: "OAuth 2.0 Dynamic Client Registration Protocol".

[11] IETF RFC 7636: "Proof Key for Code Exchange by OAuth Public Clients".

[12] IETF RFC 8259: "The JavaScript Object Notation (JSON) Data Interchange Format".

[13] IETF RFC 8705: "OAuth 2.0 Mutual-TLS Client Authentication and Certificate-Bound Access Tokens".

[14] IETF RFC 9068: "JSON Web Token (JWT) Profile for OAuth 2.0 Access Tokens".

[15] IETF RFC 9110: "HTTP Semantics".

[16] IETF RFC 9449: "OAuth 2.0 Demonstrating Proof of Possession (DPoP)".

[17] IETF RFC 9535: "JSONPath: Query Expressions for JSON".

[18] IETF RFC 9562: "Universally Unique IDentifiers (UUIDs)".

[19] IETF RFC 9700: "Best Current Practice for OAuth 2.0 Security".

[20] ISO 639:2023: "Code for individual languages and language groups".

[21] ISO 3166-1:2020: "Codes for the representation of names of countries and their subdivisions — Part 1: Country code".

[22] ISO 8601-1:2019: "Date and time — Representations for information interchange — Part 1: Basic rules".

[23] ISO 15000-2:2021: "Electronic business eXtensible Markup Language (ebXML) — Part 2: Applicability Statement (AS) profile of ebXML messaging service".

[24] ISO 15000-3:2023: "Electronic business eXtensible Markup Language (ebXML) — Part 3: Registry and repository".

[25] ISO/IEC 27001:2022: "Information security, cybersecurity and privacy protection — Information security management systems — Requirements".

[26] OASIS: "ebXML Messaging Protocol Binding for RegRep Version 1.0", Committee Specification 01, 09 March 2021.

[27] OpenAPI Initiative: "OpenAPI Specification".

[28] OpenID Foundation: "FAPI 2.0 Security Profile".

[29] OpenID Foundation: "OpenID Connect Core 1.0 incorporating errata set 2".

[30] W3C® Recommendation 21 March 2017: "W3C XML Path Language (XPath) 3.1".

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] Commission Implementing Regulation (EU) No 1352/2013 of 4 December 2013 establishing the forms provided for in Regulation (EU) No 608/2013 of the European Parliament and of the Council concerning customs enforcement of intellectual property rights.

[i.2] Commission Implementing Regulation (EU) 2015/1501 of 8 September 2015 on the interoperability framework pursuant to Article 12 (8) of Regulation (EU) No 910/2014 of the European Parliament and of the Council on electronic identification and trust services for electronic transactions in the internal market.

[i.3] Commission Implementing Regulation (EU) 2020/2244 of 17 December 2020 laying down rules for the application of Directive (EU) 2017/1132 of the European Parliament and of the Council as regards technical specifications and procedures for the system of interconnection of registers and repealing Commission Implementing Regulation (EU) 2015/884.

[i.4] Commission Implementing Regulation (EU) 2021/1042 of 18 June 2021 laying down rules for the application of Directive (EU) 2017/1132 of the European Parliament and of the Council as regards technical specifications and procedures for the system of interconnection of registers and repealing Commission Implementing Regulation (EU) 2020/2244.

[i.5] Commission Implementing Regulation (EU) 2022/1463 of 5 August 2022 setting out technical and operational specifications of the technical system for the cross-border automated exchange of evidence and application of the "once-only" principle in accordance with Regulation (EU) 2018/1724 of the European Parliament and of the Council.

NOTE: CIR (EU) 2022/1463 defines the "Once-Only Technical System" (OOTS) for the cross-border automated exchange of evidence referred to in Article 14 (1) of the "Single Digital Gateway" (SDG) Regulation (EU) 2018/1724 [i.35], whereas the technical details are specified within [i.19].

[i.6] Commission Implementing Regulation (EU) 2022/1860 of 10 June 2022 laying down implementing technical standards for the application of Regulation (EU) No 648/2012 of the European Parliament and of the Council with regard to the standards, formats, frequency and methods and arrangements for reporting.

[i.7] Commission Implementing Regulation (EU) 2024/2977 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 person identification data and electronic attestations of attributes issued to European Digital Identity Wallets.

[i.8] Commission Implementation 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.9] Commission Implementing Regulation (EU) 2025/846 of 6 May 2025 laying down rules for the application of Regulation (EU) No 910/2014 of the European Parliament and of the Council as regards cross-border identity matching of natural persons.

[i.10] Commission Implementing Regulation (EU) 2025/848 of 6 May 2025 laying down rules for the application of Regulation (EU) No 910/2014 of the European Parliament and of the Council as regards the registration of wallet-relying parties.

[i.11] Commission Implementing Regulation (EU) 2025/1569 of 29 July 2025 laying down rules for the application of Regulation (EU) No 910/2014 of the European Parliament and of the Council as regards qualified electronic attestations of attributes and electronic attestations of attributes provided by or on behalf of a public sector body responsible for an authentic source.

[i.12] Council Directive 2006/112/EC of 28 November 2006 on the common system of value added tax.

[i.13] Council Regulation (EU) No 389/2012 of 2 May 2012 on administrative cooperation in the field of excise duties and repealing Regulation (EC) No 2073/2004.

[i.14] ETSI TS 119 411-8: "Electronic Signatures and Trust Infrastructures (ESI); Policy and security requirements for Trust Service Providers issuing certificates; Part 8: Access Certificate Policy for EUDI Wallet Relying Parties".

[i.15] ETSI TS 119 461: "Electronic Signatures and Trust Infrastructures (ESI); Policy and security requirements for trust service components providing identity proofing of trust service subjects".

[i.16] European Commission: "Common Services API Specification".

[i.17] European Commission: "DCAT-AP 3.0".

[i.18] European Commission: "eDelivery ebCore Party Id 2.0".

[i.19] European Commission: "OOTS Technical Design Documents".

[i.20] European Commission: "Specification of systems enabling the notification and subsequent publication of Provider information".

[i.21] European Commission: "Specification of common formats and API for Relying Party Registration information".

[i.22] European Data Protection Board: "Guidelines 01/2022 on data subject rights - Right of access".

[i.23] IETF RFC 7521: "Assertion Framework for OAuth 2.0 Client Authentication and Authorization Grants".

[i.24] IETF RFC 7523: "JSON Web Token (JWT) Profile for OAuth 2.0 Client Authentication and Authorization Grants".

[i.25] IETF RFC 7662: "OAuth 2.0 Token Introspection".

[i.26] IETF RFC 8610: "Concise Data Definition Language (CDDL): A Notational Convention to Express Concise Binary Object Representation (CBOR) and JSON Data Structures".

[i.27] ISO 17442-1:2020: "Financial services — Legal entity identifier (LEI) - Part 1: Assignment".

[i.28] ISO/IEC 18013-5: "Personal identification - ISO - compliant driving licence - Part 5: Mobile driving licence (mDL) application".

[i.29] ISO/IEC 23220-2: "Cards and security devices for personal identification - Building blocks for identity management via mobile devices - Part 2: Data objects and encoding rules for generic eID-System".

[i.30] JSON Schema Community: "JSON Schema Specification".

[i.31] OpenID Foundation: "OpenID for Verifiable Credential Issuance 1.0".

[i.32] OpenID Foundation: "OpenID for Verifiable Credential Presentation 1.0".

[i.33] Regulation (EU) No 910/2014 of the European Parliament and of the Council of 23 July 2014 on electronic identification and trust services for electronic transactions in the internal market and repealing Directive 1999/93/EC.

NOTE: The Regulation (EU) No 910/2014 [i.33] on electronic identification and trust services enacted in 2014, which is informally referred to as "eIDAS-Regulation", has been amended in 2024 by the Regulation (EU) 2024/1183 [i.36] on the "European Digital Identity Framework" to form what is called in the present document the "amended eIDAS-Regulation".

[i.34] Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation).

NOTE: The Regulation (EU) 2016/679 is referred to in the present text as "GDPR".

[i.35] Regulation (EU) 2018/1724 of the European Parliament and of the Council of 2 October 2018 establishing a single digital gateway to provide access to information, to procedures and to assistance and problem-solving services and amending Regulation (EU) No 1024/2012.

NOTE: The Regulation (EU) 2018/1724 is in the present document referred to as "SDG-Regulation".

[i.36] Regulation (EU) 2024/1183 of the European Parliament and of the Council amending Regulation (EU) No 910/2014 as regards establishing the European Digital Identity Framework.

[i.37] W3C Recommendation 22 August 2024: "Data Catalog Vocabulary (DCAT) - Version 3".

[i.38] W3C Recommendation 16 July 2020: "JSON-LD 1.1".

[i.39] W3C Recommendation 5 April 2012: "W3C XML Schema Definition Language (XSD) 1.1 Part 1: Structures".

[i.40] W3C Recommendation 5 April 2012: "W3C XML Schema Definition Language (XSD) 1.1 Part 2: Datatypes".

[i.41] W3C Recommendation 03 March 2022: "Verifiable Credentials Data Model v1.1".

3 Definition of terms, symbols, abbreviations and notation

3.1 Terms

For the purposes of the present document, the terms given in the amended eIDAS-Regulation [i.33], the SDG-Regulation [i.35] including the related implementing acts CIR (EU) 2015/1501 [i.2], CIR (EU) 2022/1463 [i.5], CIR (EU) 2025/848 [i.10] as well as the OAuth 2.0 standard from IETF RFC 6749 [6] and the following apply:

NOTE: The most important terms are included here for the convenience of the reader.

attribute: characteristic, quality, right or permission of a natural or legal person or of an object

NOTE: See Article 3 (43) of the eIDAS-Regulation [i.33].

authentic source: repository or system, held under the responsibility of a public sector body or private entity, that contains and provides attributes about a natural or legal person or object and that is considered to be a primary source of that information or recognized as authentic in accordance with Union or national law, including administrative practice

NOTE: See Article 3 (47) of the eIDAS-Regulation [i.33] and clause 4 of the present document.

authentic source interface: interface. which allows to verify or retrieve attributes, which are stored within an authentic source

NOTE: See clause 6 of the present document.

authentic source interface provider: provider of an authentic source interface

NOTE: The authentic source interface provider can be an authentic source or an intermediary acting on its behalf.

authorization server: server issuing access tokens to the client after successfully authenticating the resource owner and obtaining authorization

NOTE: See clause 1.1 of IETF RFC 6749 [6] and clause 6.1.3 of the present document.

authorization server provider: provider of an authorization server

catalogue of attributes: digital repository of attributes that is maintained and published online by the Commission

NOTE: See Article 2 (3) of CIR (EU) 2025/1569 [i.11].

client: application making protected resource requests on behalf of the resource owner and with its authorization

NOTE 1: The term "client" does not imply any particular implementation characteristics (e.g. whether the application executes on a server, a desktop, or other devices).

NOTE 2: See clause 1.1 of IETF RFC 6749 [6] and clause 6.1.3 of the present document.

common services: collection of services provided by the European Commission related to the catalogue of attributes and the once-only technical system, which contain the semantic repository and the data service directory and allow to be accessed through the discover interface

NOTE: See also Article 1 (10) and Article 4 (1) of CIR (EU) 2022/1463 [i.5].

data service directory: registry which can be accessed through the discover interface and which contains the list of data services for the verification or retrieval of attributes

NOTE: See also Article 1 (7) and Article 5 of CIR (EU) 2022/1463 [i.5].

discover interface: interface provided by a discover interface provider, which allows to search for the unique identifier of a specific attribute and retrieve related semantic specifications, data models and information related to data services for the verification or retrieval of attributes

NOTE: See clause 5 of the present document and also Article 1 (7) and Article 5 of CIR (EU) 2022/1463 [i.5].

discover interface provider: provider of a discover interface

electronic attestation of attributes: attestation in electronic form that allows attributes to be authenticated

NOTE: See Article 3 (44) of the eIDAS-Regulation [i.33].

electronic attestation of attributes issued by or on behalf of a public sector body responsible for an authentic source: electronic attestation of attributes issued by a public sector body that is responsible for an authentic source or by a public sector body that is designated by the Member State to issue such attestations of attributes on behalf of the public sector bodies responsible for authentic sources in accordance with Article 45f and with Annex VII

NOTE: See Article 3 (46) of the eIDAS-Regulation [i.33].

European Digital Identity Wallet: electronic identification means which allows the user to securely store, manage and validate person identification data and electronic attestations of attributes for the purpose of providing them to relying parties and other users of European Digital Identity Wallets, and to sign by means of qualified electronic signatures or to seal by means of qualified electronic seals

NOTE: See Article 3 (42) of the eIDAS-Regulation [i.33].

Once-Only Technical System ('OOTS'): technical system for the cross-border automated exchange of evidence

NOTE: See Article 14 (1) of Regulation (EU) 2018/1724 [i.35] and Article 1 (1) of CIR (EU) 2022/1463 [i.5].

person identification data: set of data that is issued in accordance with Union or national law and that enables the establishment of the identity of a natural or legal person, or of a natural person representing another natural person or a legal person

NOTE: See Article 3 (3) of the eIDAS-Regulation [i.33].

public sector electronic attestation of attributes provider: trust service provider, which issues electronic attestations of attributes issued as, or on behalf of, a public sector body responsible for an authentic source

qualified electronic attestation of attributes: electronic attestation of attributes which is issued by a qualified trust service provider and meets the requirements laid down in Annex V of Regulation (EU) No 910/2014

NOTE: See Article 3 (45) of the eIDAS-Regulation [i.33].

qualified trust service provider: trust service provider who provides one or more qualified trust services and is granted the qualified status by the supervisory body

NOTE: See Article 3 (20) of the eIDAS-Regulation [i.33].

resource owner: entity capable of granting access to a protected resource

NOTE 1: When the resource owner is a natural person, it is referred to as an end-user.

NOTE 2: See clause 1.1 of IETF RFC 6749 [6] and clause 6.1.3 of the present document.

resource server: server hosting the protected resources, capable of accepting and responding to protected resource requests using access tokens

NOTE: See clause 1.1 of IETF RFC 6749 [6] and clause 6.1.3 of the present document.

semantic data specifications: specifications, with human- and/or machine-processable representations, that define how data is recommended to be organized, described and interpreted to facilitate accurate understanding and exchange between systems, applications, and people

NOTE: Artefacts comprising a semantic data specification can include visual diagrams (e.g. UML class diagrams), human-readable documentation, schemas (e.g. RDF schema, XML schema and JSON schema), alongside pertinent code lists and constraints, in relevant formats.

semantic repository: collection of descriptions of semantic assets, including semantic data specifications and schemas related to attributes which can be accessed through the discover interface and which allow to search for the unique identifier of a specific attribute and retrieve related semantic specifications and pertinent schema distributions

NOTE: See Article 1 (10) and Article 7 of CIR (EU) 2022/1463 [i.5].

trust service: electronic service normally provided for remuneration which consists of any of the following:

a) the issuance of certificates for electronic signatures, certificates for electronic seals, certificates for website authentication or certificates for the provision of other trust services;

b) the validation of certificates for electronic signatures, certificates for electronic seals, certificates for website authentication or certificates for the provision of other trust services;

c) the creation of electronic signatures or electronic seals;

d) the validation of electronic signatures or electronic seals;

e) the preservation of electronic signatures, electronic seals, certificates for electronic signatures or certificates for electronic seals;

f) the management of remote electronic signature creation devices or remote electronic seal creation devices;

g) the issuance of electronic attestations of attributes;

h) the validation of electronic attestation of attributes;

i) the creation of electronic timestamps;

j) the validation of electronic timestamps;

k) the provision of electronic registered delivery services;

l) the validation of data transmitted through electronic registered delivery services and related evidence;

m) the electronic archiving of electronic data and electronic documents;

n) the recording of electronic data in an electronic ledger.

NOTE 1: See Article 3 (16) of the eIDAS-Regulation [i.33].

NOTE 2: Within the scope of the present document the trust services for the issuing of certificates according to Article 3 (16) (a) of the eIDAS-Regulation [i.33] and for the issuing of electronic attestations of attributes according to Article 3 (16) (g) of the eIDAS-Regulation [i.33] are especially relevant, as the interfaces to the authentic sources specified in the present document aim at facilitating the implementation of such trust services.

trust service provider: natural or a legal person who provides one or more trust services either as a qualified or as a non-qualified trust service provider

NOTE: See Article 3 (19) of the eIDAS-Regulation [i.33] and clause 4 of the present document.

user: natural or legal person, or a natural person representing another natural person or a legal person, that is subject of the attribute stored in the authentic source and which is capable of authorizing the access to the authentic source interface

wallet-relying party: relying party that intends to rely upon wallet units for the provision of public or private services by means of digital interaction

NOTE: See Article 2 (1) of CIR (EU) 2025/848 [i.10].

wallet-relying party access certificate: certificate for electronic seals or signatures authenticating and validating the wallet-relying party issued by a provider of wallet-relying party access certificates

NOTE: See Article 2 (12) of CIR (EU) 2025/848 [i.10] and ETSI TS 119 411-8 [i.14].

wallet-relying party registration certificate: data object that describes the intended use of the relying party and indicates the attributes the relying party has registered to intend to request from users

NOTE: See Article 2 (15) of CIR (EU) 2025/848 [i.10].

wallet unit: unique configuration of a wallet solution that includes wallet instances, wallet secure cryptographic applications and wallet secure cryptographic devices provided by a wallet provider to an individual wallet user

NOTE: See Article 2 (2) of CIR (EU) 2025/848 [i.10] and clause 4 of the present document.

wallet user: means a user who is in control of the wallet unit

NOTE: See Article 2 (9) of CIR (EU) 2025/848 [i.10] and clause 4 of the present document.

3.2 Symbols

Void.

3.3 Abbreviations

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

API	Application Programming Interface
ASIP	Authentic Source Interface Provider
AuthSrc	Authentic Source
AZSP	Authorization Server Provider
CBOR	Concise Binary Object Representation
CDDL	Concise Data Definition Language
CIR	Commission Implementing Regulation
DIP	Discover Interface Provider
DPoP	Demonstrating Proof of Possession
DSD	Data Service Directory
eID	Electronic IDentification
eIDAS	Electronic Identification and Trust Services
EUDIW	European Digital Identity Wallet
GDPR	General Data Protection Regulation
JSON	JavaScript Object Notation
JSON-LD	JSON-based Serialization for Linked Data
JWT	JSON Web Token
MTLS	Mutual Transport Layer Security
OOTS	Once-Only Technical System
OpenID4VCI	OpenID for Verifiable Credential Issuance
OpenID4VP	OpenID for Verifiable Credential Presentation
PID	Person Identification Data
PKCE	Proof Key for Code Exchange
PubEAA	Public sector Electronic Attestation of Attributes provider
QTSP	Qualified Trust Service Provider
RDF	Resource Description Framework
RIM	Registry Information Model
SDG	Single Digital Gateway
SR	Semantic Repository
TSP	Trust Service Provider
URI	Unique Resource Identifier
URL	Uniform Resource Locator
URN	Uniform Resource Name
UUID	Universally Unique IDentifier
WRPAC	Wallet-Relying Party Access Certificate
XML	eXtensible Markup Language

3.4 Notation

Each requirement is identified as follows:

- REQ-ASIP-< the clause number>-<2-digit number - incremental>-<optional sub-identifier> identifies requirements specified for the Authentic Source Interface Provider (ASIP).
- REQ-AZSP-< the clause number>-<2-digit number - incremental>-<optional sub-identifier> identifies requirements specified for the Authorization Server Provider (AZSP).
- REQ-DIP-< the clause number>-<2-digit number - incremental>-<optional sub-identifier> identifies requirements specified for the Discover Interface Provider (DIP).
- REQ-TSP-< the clause number>-<2-digit number - incremental>-<optional sub-identifier> identifies requirements specified for the (Qualified) Trust Service Provider (TSP).

4 System Architecture

4.1 Overview

The high-level overview of the system architecture is depicted in Figure 1 below and described in the following clauses.

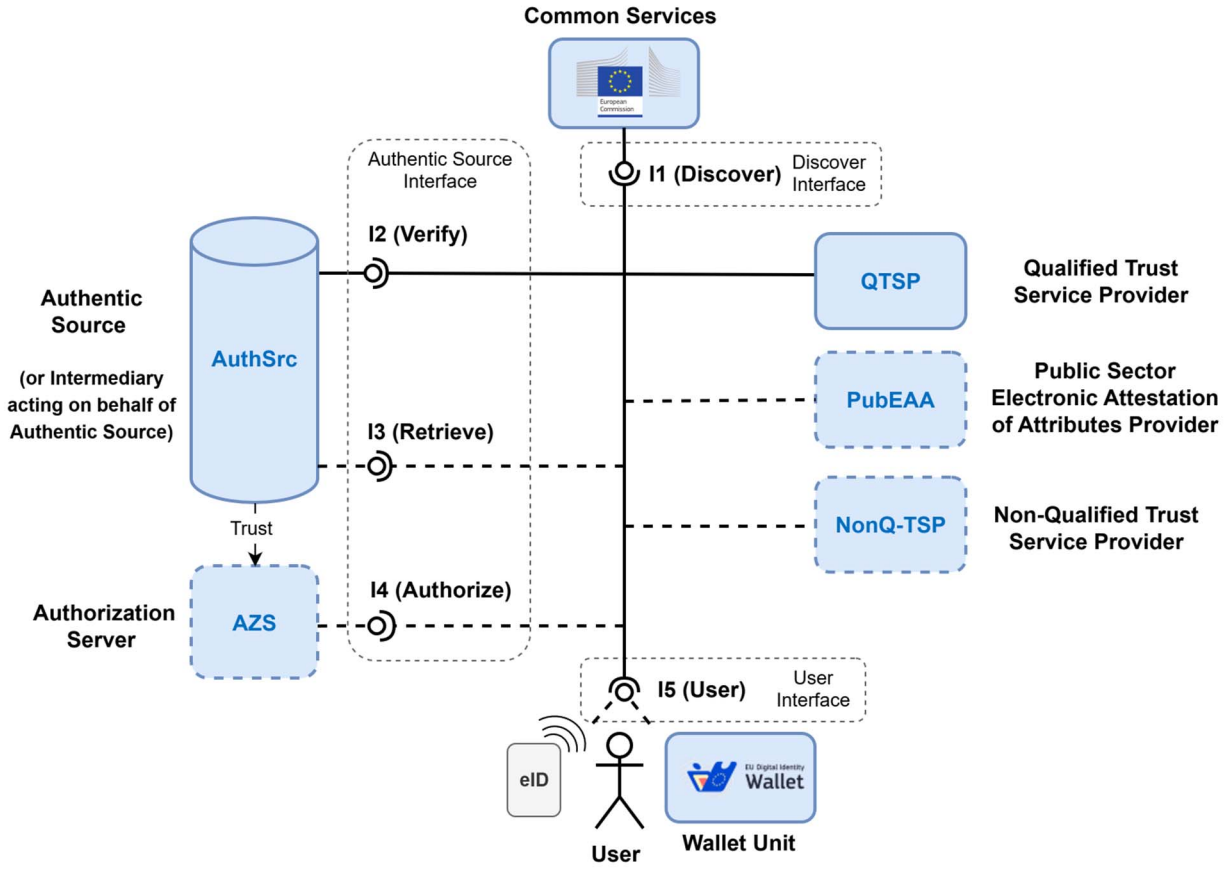

Figure 1: High Level Overview of the System Architecture

4.2 Common Services

The **Common Services** are operated by the European Commission for the once-only technical system according to CIR (EU) 2022/1463 [i.5]. Their functionality can be reused for the catalogue of attributes according to Article 2 (3) of CIR (EU) 2025/1569 [i.11]. The **I1 (Discover)** interface, as specified in clause 5, is designed to be used in this context. It is envisioned to be provided by a Discover Interface Provider (DIP) and allows to search for the unique identifier of a specific attribute and retrieve related semantic specifications, data models and information related to data services for the verification or retrieval of attributes. The data services within the authentic source interface are provided by an Authentic Source Interface Provider (ASIP), which can be an authentic source, or an intermediary acting on its behalf. The ASIP provide end points for the verification of attributes using the **I2 (Verify)** interface, and optionally for the retrieval of attributes using the **I3 (Retrieve)** interface, if this is supported.

4.3 Authentic Source

The **Authentic Source** is defined in Art. 3 (47) of the amended eIDAS-Regulation (EU) No 910/2014 [i.33] to be "a repository or system, held under the responsibility of a public sector body or private entity, that contains and provides attributes about a natural or legal person or object and that is considered to be a primary source of that information or recognized as authentic in accordance with Union or national law, including administrative practice". The authentic source interface is provided by an ASIP, which is an authentic source or an intermediary acting on its behalf and contains at least the **I2 (Verify)** interface. It can also contain the **I3 (Retrieve)** interface. If the authentic source interface uses HTTP / OAuth as specified in clause 6.1, the **I4 (Authorize)** interface is used for the authorization of the request by the user, as specified in clause 6.1.3.

REQ-ASIP-4.3-01: The ASIP shall provide the **I2 (Verify)** interface.

This interface can be used by qualified trust service providers according to Article 45e of the amended eIDAS-Regulation [i.33], to verify at least the attributes listed Annex VI of the Regulation, wherever those attributes rely on authentic sources within the public sector, at the request of the user.

REQ-ASIP-4.3-02: The ASIP may provide the **I3 (Retrieve)** interface.

This is an additional interface, which allows to retrieve attributes values from the authentic source, or an intermediary acting on its behalf, upon the request of the user.

NOTE: A legal basis for supporting the retrieval of attributes upon the request of the user, which is the data subject according to Article 4 (1) of GDPR [i.34], can be Article 15 (3) of GDPR [i.34], if such a request is not restricted by union or national law as explained in clause 6 of [i.22]. If there is no such restriction, Article 15 (3) of GDPR [i.34] requires the controller in the sense of Article 4 (7) GDPR [i.34] to provide a copy of the stored data to the data subject "in a commonly used electronic form" upon its electronic request.

For the logical interface **I2 (Verify)** the Member States can, according to Article 9 (5) of CIR (EU) 2025/1569 [i.11], "impose access controls or other verification mechanisms that provide integrity, authenticity, and confidentiality to determine that the requester is a trust service provider and is acting at the request of a legitimate user". Such access control and authorization mechanisms can also be applied to the **I3 (Retrieve)** interface.

REQ-ASIP-4.3-03: The ASIP should impose access controls or other verification mechanisms that provide integrity, authenticity, and confidentiality to determine that the requester is a trust service provider and is acting at the request of a legitimate user.

The present document specifies two different technical variants for the realization of the authentic source interface.

REQ-ASIP-4.3-04: The ASIP shall choose to support either:

- the interface based on **HTTP / OAuth** using HTTP according to IETF RFC 9110 [15], with authorization based on OAuth according to IETF RFC 6749 [6], as specified in clause 6.1; or
- the interface based on **ISO 15000** according to the different parts of ISO 15000 [23] and [24], as specified in clause 6.2;

or both.

4.4 Authorization Server

The **Authorization Server** with the interface **I4 (Authorize)**, based on the OAuth 2.0 authorization framework specified in IETF RFC 6749 [6] and related standards, is used in case the authentic source interface is based on HTTP, as specified in clause 6.1.

REQ-AZP-4.4-01 [CONDITIONAL]: If the authentic source interface is based on HTTP, as specified in clause 6.1, there shall be an Authorization Server Provider (AZSP), which operates an authorization server according to the standards specified in clause 6.1.3.

In case of the authentic source interface based on HTTP, the **I4 (Authorize)** is used for the authorization of the requests for verification or retrieval of attributes sent by the (qualified) trust service provider, which acts as client, to the authentic source, or the intermediary acting on its behalf, which acts as resource server.

REQ-AZP-4.4-02 [CONDITIONAL]: If the authentic source interface is based on ISO 15000, as specified in clause 6.2, the authorization of requests by the (qualified) trust service providers shall be handled directly within the endpoints of the authentic source interface.

NOTE 1: Article 45e of the amended eIDAS-Regulation (EU) No 910/2014 [i.33] requires that the verification requests are "at the request of the user" and Article 9 (3) of CIR (EU) 2025/1569 [i.11] requires that the "verification request shall set out the attributes and the identification data of the subject of the attribute for which the qualified trust service provider requests the verification."

NOTE 2: In case of the HTTP- and OAuth 2.0-based interface, the authorization of the request by the user and the gathering of the necessary identification data of the subject of the attribute is performed *within* the user authentication, identification and consent step (9) in Figure 3, which corresponds to step (B) in clause 4.1 of IETF RFC 6749 [6] as part of the request from the (qualified) trust service provider to the authentic source interface provider. That the identification data have been gathered to authorize a specific request is ensured by *technical means* within the OAuth 2.0 based authorization code flow according to clause 4.1 of IETF RFC 6749 [6], as specified in clause 6.1.3 in more detail.

NOTE 3: In case of the ISO 15000-based interface, any authorization of the later request by the user and the gathering of the necessary identification data of the subject of the attribute is performed *before* the request from the (qualified) trust service provider to the authentic source interface under the responsibility of the (qualified) trust service provider. That the identification data have been gathered to authorize a specific request needs to be ensured by *technical or organizational means* within the system of the (qualified) trust service provider, which is audited in a suitable manner.

4.5 User

The **User** with the logical interface **I5 (User)** is part of the environment of the user and allows the trust service provider, which acts as OAuth 2.0 client, in case of the HTTP-based interface, or requester, in case of the ISO 15000-based interface, to obtain the personal identification data of the subject of the attribute as mentioned in Article 9 (3) of CIR (EU) 2025/1569 [i.11]. The authentication, identification and authorization of the request by the user always takes place before the verification or retrieval request is sent to the authentic source, or an intermediary acting on its behalf.

NOTE: Article 45e of Regulation (EU) No 910/2014 [i.33] only requires that the verification request is performed "at the request of the user, in accordance with Union or national law". Furthermore, Article 9 (3) of CIR (EU) 2025/1569 [i.11] only requires that "the verification request shall set out the attributes and the identification data of the subject of the attribute for which the qualified trust service provider requests the verification". This means that using the EUDIW to obtain the identification data of the subject of the attribute within the authentication and authorization of the request for the verification or retrieval of attributes is only one possible option. The Authorization Server, or an analogous component for user authentication, identification and authorization in case of the ISO 15000-based interface, is also allowed to utilize other methods for identity proofing, like notified Electronic Identification (eID) means or methods in line with ETSI TS 119 461 [i.15] for example.

4.6 Trust Service Providers

There are **Trust Service Providers**, where one can distinguish between:

- **Qualified Trust Service Providers**, which can access the **I2 (Verify)** interface according to Article 45e of the eIDAS-Regulation (EU) No 910/2014 [i.33] mentioned above and which can be granted access to the **I3 (Retrieve)** interface, if supported by the ASIP;
- **Public Sector Electronic Attestation of Attributes Provider** according to Article 45f of the eIDAS-Regulation (EU) No 910/2014 [i.33], which can, if they are different from the authentic source itself, be allowed to access the **I2 (Verify)** and/or **I3 (Retrieve)** interface, if this is granted by the ASIP; and
- **Non-Qualified Trust Service Provider**, which can be allowed to access the **I2 (Verify)** and/or **I3 (Retrieve)** interface, if this access is granted by the ASIP.

5 Discover Interface

5.1 I1 (Discover)

REQ-DIP-5.1-01: The DIP shall provide the **(I1) Discover** interface, which allows to search for attributes and their unique identifier, as well as retrieve detailed information on specific data service endpoints.

The **(I1) Discover** interface can be mapped to the interfaces provided in the context of the OOTS common services according to CIR (EU) 2022/1463 [i.5] and can be used in the context of catalogue of attributes according to Article 2 (3) of CIR (EU) 2025/1569 [i.11]. It is realized as a lightweight HTTP-based API that enables the discovery and retrieval of metadata about attributes and related data services, which are offered by ASIPs.

The discover interface is designed to be interoperable with the semantic repository and the data service directory of the common services of the OOTS according to CIR (EU) 2022/1463 [i.5], as Article 10 (2) of CIR (EU) 2025/1569 [i.11] requires that for the catalogue of attributes, "the European Commission shall refer to and re-use, where appropriate, the common services of the technical system pursuant to Regulation (EU) 2018/1724 [i.35]". This allows that authentic sources, attributes, data services and other OOTS related data can be registered in a single shared set of registries:

- **Semantic Repository (SR)** - is a catalogue of semantic assets (e.g. attributes, attestation schemes, code lists) enabling the discovery of the unique identifier for a specific attribute and related semantic information and data models.
- **Data Service Directory (DSD)** - is a registry of authentic sources and their data services, which allow to verify or retrieve attributes via interoperable endpoints.

Together, SR and DSD provide the functionalities of the discover interface, which enables (qualified) trust service providers or other requesters to search for attribute specific metadata and data service endpoints for the verification and/or retrieval services in some Member State.

The **(I1) Discover** interface provides two key operations:

- `search` - is used to find attributes and their unique identifier using the semantic repository. This operation is expected to be mostly used at design-time, with optional run-time use.
- `retrieve` - is used for obtaining information about the provided endpoints for the verification or retrieval of information using the data service directory. This operation is expected to be used at run-time.

5.2 Find Attributes (`search`)

5.2.1 Overview and example for `search`

REQ-DIP-5.2.1-01: The DIP shall provide the `search` query, which allows to search for catalogued attribute-related metadata in the semantic repository, optionally filtered by specific metadata.

REQ-DIP-5.2.1-02: The `search` request shall be implemented as HTTP GET request according to clause 9.3.1 of IETF RFC 9110 [15].

EXAMPLE: The `search` request is illustrated by the following example:

```
GET /search?assetType=attribute&country=DE&text=Birth
```

5.2.2 Input parameters for `search`

REQ-DIP-5.2.2-01: The `search` query shall contain the parameter `assetType`, which represents the **type** of the catalogued asset to search for.

REQ-DIP-5.2.2-02: Within the scope of the present document, the value of `assetType` shall be fixed to the value "`attribute`".

REQ-DIP-5.2.2-03: The `search` query may contain the parameter `creator`.

- **REQ-DIP-5.2.2-03-01 [CONDITIONAL]**: If it is present, the parameter `creator` shall be of type string and represent the **name of the creator** of the catalogue asset (e.g. entity submitting the attribute to the catalogue).

REQ-DIP-5.2.2-04: The `search` query may contain the parameter `country`.

- **REQ-DIP-5.2.2-04-01 [CONDITIONAL]**: If it is present, the parameter `country` shall be encoded as alpha-2 country code according to ISO 3166-1 [21] and shall allow to filter for the **country code** of the Member State of the creator of the catalogue asset.

NOTE 1: Article 7 (3) of CIR (EU) 2025/1569 [i.11] stipulates the following: "Member States shall request the inclusion of attributes listed in Annex VI to Regulation (EU) No 910/2014 [i.33] to the catalogue of attributes wherever those attributes rely on authentic sources for the purpose of the verification by qualified trust service providers."

REQ-DIP-5.2.2-05: The `search` query may contain the parameter `text`.

- **REQ-DIP-5.2.2-05-01 [CONDITIONAL]**: If it is present, the parameter `text` shall be of type string and allows a **free-text** search on the names and descriptions of the catalogued attribute.

REQ-DIP-5.2.2-06: The `search` query may contain the parameter `semanticDataSpecification`.

- **REQ-DIP-5.2.2-06-01 [CONDITIONAL]**: If it is present, the parameter `semanticDataSpecification` shall be a Unique Resource Identifier (URI) according to IETF RFC 3986 [2], and shall allow to filter for a specific **semantic data specification** that prescribes structured and standardized formats for the organization, description, and interpretation of data to ensure attribute conformity, semantic consistency, and interoperable exchange among systems, applications, and users, independent of the media type.

REQ-DIP-5.2.2-07: The `search` query may contain the parameter `schemaMediaType`.

- **REQ-DIP-5.2.2-07-01 [CONDITIONAL]**: If it is present, the parameter `schemaMediaType` shall be a media type according to IETF RFC 6838 [8], and shall filter for the distribution of the data model schema the attribute conforms to.

There are at least the following media types related to data model schema types:

EXAMPLE 1: `application/json-schema` - indicates, that a link to the latest **JSON-schema** according to [i.30] is to be returned.

NOTE 2: This type of data model schema is used in the HTTP-based authentic source interface, as specified in clause 6.1.

EXAMPLE 2: `application/ld+json` - indicates, that a link to the **JSON-LD-schema** according to the W3C JSON-LD 1.1 specification [i.38] is to be returned, if it is available in the catalogue of attributes.

EXAMPLE 3: `application/xml` - indicates, that a link to the **XML-schema** according to the W3C XML-Schema specifications [i.39] and [i.40] is to be returned, if it is available in the catalogue of attributes.

NOTE 3: This type of data model schema is used in the ISO 15000-based authentic source interface, as specified in clause 6.2.

EXAMPLE 4: `application/cddl` - indicates, that a link to the CDDL-schema according to IETF RFC 8610 [i.26] shall be returned, if it is available in the catalogue of attributes.

NOTE 4: This type of data model schema is used for the specification, generation and validation of electronic attestations of attributes according to ISO/IEC 18013-5 [i.28] or ISO/IEC 23220-2 [i.29].

NOTE 5: The catalogue of attributes according to Article 2 (3) of CIR (EU) 2025/1569 [i.11] may be expected to contain at least the JSON-schema according to [i.30] and the CDDL-schema according to [i.26], because Annex II (1) of CIR (EU) 2025/1569 [i.11] stipulates that PubEAA need to issue their attestations in a format according to Annex II of CIR (EU) 2024/2979 [i.8], which in turn refers to ISO/IEC 18013-5 [i.28] and W3C's Verifiable Credentials Data Model v1.1 [i.41].

NOTE 6: The optional elements above serve as filter. If such an optional element is missing, no filtering is applied with respect to the specific parameter. If no element for filtering is provided, the full set of metadata related to a specific attribute is returned.

5.2.3 Output parameters for `search`

REQ-DIP-5.2.3-01: The **output** of the `search` query shall be an `attributes` object.

REQ-DIP-5.2.3-02: An `attributes` object shall contain an array of `attribute` objects, which contain attribute-related metadata.

REQ-DIP-5.2.3-03: Each `attribute` object shall contain a specific `attributeIdentifier` element.

- **REQ-DIP-5.2.3-03-01:** The `attributeIdentifier` element shall be a Unique Resource Identifier (URI) according to IETF RFC 3986 [2], which shall serve as the **unique identifier** of the attribute.
- **REQ-DIP-5.2.3-03-02:** The `attributeIdentifier` element shall contain the namespace, the local identifier and the version of the attribute as required in Article 7 (5) of CIR (EU) 2025/1569 [i.11].

REQ-DIP-5.2.3-04 [CONDITIONAL]: In case the catalogue contains a localized friendly name of the attribute, the `attribute` object shall contain one or more localized `title` elements.

- **REQ-DIP-5.2.3-04-01 [CONDITIONAL]**: If it is present, a `title` element shall contain a `value` element with the free form text of the name of the attribute and a `language` element in form of a two-letter language code according to ISO 639 [20] to provide a **localized friendly name** of the attribute.

REQ-DIP-5.2.3-05 [CONDITIONAL]: In case the catalogue contains a localize description of the attribute, the `attribute` object shall contain one or more localized `description` elements.

- **REQ-DIP-5.2.3-05-01 [CONDITIONAL]**: If it is present, a `description` element shall contain a `value` element with the free form text of the description of the attribute and a `language` element in form of a two-letter language code according to ISO 639 [20] to provide a human-friendly **localized description** of the attribute.

REQ-DIP-5.2.3-06 [CONDITIONAL]: If the catalogue contains the name of the creator of the catalogue, the `attribute` object shall contain a `creator` element.

- **REQ-DIP-5.2.3-06-01 [CONDITIONAL]**: If it is present, the `creator` element shall be of type *string* and shall contain the **name of the creator** of the catalogue asset (e.g. entity submitting the attribute to the catalogue).

REQ-DIP-5.2.3-07 [CONDITIONAL]: If the country of the creator of the catalogue entry is present in the catalogue, the `attribute` object shall contain a `country` element.

- **REQ-DIP-5.2.3-07-01 [CONDITIONAL]**: If it is present, the `country` element shall be the alpha-2 **country code** according to ISO 3166-1 [21] of the Member State of the creator of the catalogue asset.

REQ-DIP-5.2.3-08 [CONDITIONAL]: If the semantic data specification of the attribute is present in the catalogue, the `attribute` object shall contain a `semanticDataSpecification` element.

- **REQ-DIP-5.2.3-08-01 [CONDITIONAL]**: If it is present, the `semanticDataSpecification` element shall be the **semantic data specification** of the attribute in form of a URI according to IETF RFC 3986 [2] and shall reference a structured, standardized, and media type-agnostic definition describing the attribute's data elements, semantics, and relationships to ensure consistent interpretation, interoperability, and reliable information exchange across systems and applications.

REQ-DIP-5.2.3-09: An `attribute` object shall contain one or more `schemaDistribution` elements, which shall specify the **schema distributions** specified in the semantic data specification of the attribute.

REQ-DIP-5.2.3-10: Each `schemaDistribution` element shall contain the following two child elements: `accessURL` and `mediaType`.

- **REQ-DIP-5.2.3-10-01**: The `accessURL` element shall be the Unique Resource Locator (URL) according to IETF RFC 1738 [1], which gives access to the schema distribution.

NOTE 1: See also clause 6.8.9 of W3C DCAT [i.37].

- **REQ-DIP-5.2.3-10-02**: The `mediaType` element shall be the media type according to IETF RFC 6838 [8] of the schema distribution.

NOTE 2: See clause 5.2.2 for examples for media types for data model schemas.

NOTE 3: The catalogued attributes and their corresponding specifications can be considered to be catalogued "Datasets" in the sense of clause 6.6 of W3C DCAT [i.37], or clause 7.7 of DCAT-AP [i.17], which can have multiple "Distributions" in different formats, as specified in clause 6.8 of W3C DCAT [i.37], or clause 7.9 of DCAT-AP [i.17]. As specified in clause 7.9 of DCAT-AP [i.17], such a "Distribution" can have, among others, an access URL (linked schema) and a media type as properties.

REQ-DIP-5.2.3-11: The output of the `search` query may contain additional elements.

5.3 Find Data Services for Attributes (`retrieve`)

5.3.1 Overview and example for `retrieve`

REQ-DIP-5.3.1-01: The DIP shall provide the `retrieve` query, which allows to lookup data services in the DSD for the verification or retrieval of attributes within a given Member State.

REQ-DIP-5.3.1-02: The `retrieve` request shall be implemented as HTTP GET request according to clause 9.3.1 of IETF RFC 9110 [15].

EXAMPLE: The `retrieve` request for the attribute with `attributeIdentifier` equal to "`https://example.com/attribute`", which needs to be URL-encoded as specified in clause 2 of IETF RFC 3986 [2] to "`https%3A%2F%2Fexample.com%2Fattribute`", is illustrated by the following example:

```
GET
retrieve?queryType=dataServices&attributeIdentifier=https%3A%2F%2F
example.com%2Fattribute&country=DE
```

5.3.2 Input parameters for `retrieve`

REQ-DIP-5.3.2-01: The `retrieve` query shall have as **input** parameters the following two parameters: `queryType` and `attributeIdentifier`.

- **REQ-DIP-5.3.2-01-01**: The `queryType` element shall specify the **type of the query** and shall in the scope of the present document be equal to the fixed value "`dataServices`".

- **REQ-DIP-5.3.2-01-02**: The `attributeIdentifier` element shall be the **unique identifier** of the attribute in form of a Unique Resource Identifier (URI) according to IETF RFC 3986 [2]. This identifier shall contain the namespace, the local identifier and the version of the attribute as required in Article 7 (5) of CIR (EU) 2025/1569 [i.11].

REQ-DIP-5.3.2-02: The `retrieve` query may contain the `country` element.

- **REQ-DIP-5.3.2-02-01 [CONDITIONAL]**: If it is present, the `country` element. shall be the alpha-2 **country code** according to ISO 3166-1 [21] of the Member State providing the data service for the verification or retrieval of the attribute.

- **REQ-DIP-5.3.2-02-02 [CONDITIONAL]**: If the `country` element is omitted, the data services related to the specified attribute in all EU Member States shall be returned.

REQ-DIP-5.3.2-03: The `retrieve` query may contain a `conformsTo` element.

- **REQ-DIP-5.3.2-03-01 [CONDITIONAL]**: If it is present, a `conformsTo` element shall be a URI according to IETF RFC 3986 [2], and shall be used to filter for endpoints with a specified interface technology.

The present specification defines the following two URNs for this purpose:

- `urn:ietf:rfc:9110` - to indicate that the HTTP-based interface specified in clause 6.1 is used; and
- `urn:iso:std:iso:15000` - to indicate that the ISO 15000-based interface specified in clause 6.2 is used.

5.3.3 Output parameters for `retrieve`

REQ-DIP-5.3.3-01: The **output** of the `retrieve` query shall be a `dataServices` object.

REQ-DIP-5.3.3-02: The `dataServices` object shall contain an array of `dataService` elements, which shall contain data service-related metadata as specified in the following.

REQ-DIP-5.3.3-03: The `dataService` object shall contain the following child elements:

1) `attributeIdentifier`;
2) `endpointDescription`;
3) `endpointURI`; and
4) `provider`.

- **REQ-DIP-5.3.3-03-01**: `attributeIdentifier` shall be the **unique identifier** of the attribute in form of a Unique Resource Identifier (URI) according to IETF RFC 3986 [2] and shall contain the namespace, the local identifier and the version of the attribute as required in Article 7 (5) of CIR (EU) 2025/1569 [i.4].
- **REQ-DIP-5.3.3-03-02**: `endpointDescription` shall be a URI according to IETF RFC 3986 [2] and shall point to or indicate the **interface specification** of the verification and/or retrieval service within the authentic source interface under consideration.
  - **REQ-DIP-5.3.3-03-02-01 [CONDITIONAL]**: In case of the HTTP-based interface specified in clause 6.1, this element shall refer to an OpenAPI-specification according to [27], which shall be based on the OpenAPI-based specification template provided in Annex A of the present document and which shall specify the corresponding metadata of the OAuth 2.0 based authorization server according to IETF RFC 6749 [6].
  - **REQ-DIP-5.3.3-03-02-02 [CONDITIONAL]**: In case of the ISO 15000-based interface specified in clause 6.2, this element shall be a Uniform Resource Name (URN) according to IETF RFC 5141 [4], and shall be fixed to the value "`urn:iso:std:iso:15000`".

NOTE 1: The term `endpointDescription` is defined in clause 6.9.2 of W3C's DCAT3 specification [i.37].

- **REQ-DIP-5.3.3-03-03**: `endpointURI` shall be a URI according to IETF RFC 3986 [2] and shall point to the **endpoint** of the verification and/or retrieval service within the authentic source interface under consideration.
  - **REQ-DIP-5.3.3-03-03-01 [CONDITIONAL]**: In the case of the HTTP-based interface specified in clause 6.1, this element shall be the Uniform Resource Locator (URL) of the endpoint according to IETF RFC 1738 [1].
  - **REQ-DIP-5.3.3-03-03-02 [CONDITIONAL]**: In the case of the ISO 15000-based interface specified in clause 6.2 this element shall be the party identifier and party identifier scheme of the access service of the data service according to [i.18].

NOTE 2: The `endpointURI` element is a slight generalization of the `endpointURL` property defined in clause 6.9.1 of W3C's DCAT3 specification [i.37]. This generalization is necessary to support the ISO 15000-based interface, in which the URI is not a URL, but the party identifier and party identifier scheme of the access service of the data service according to [i.18].

- **REQ-DIP-5.3.3-03-04**: `provider` shall contain the **identification of the ASIP**, which is responsible for the verification or retrieval of the attribute via the data service identified using the person attributes defined in CIR (EU) 2024/2977 [i.7] as specified in the following:
  - **REQ-DIP-5.3.3-03-04-01**: The `provider` element shall contain a child element `legalName`, which contains the **current legal name** of the ASIP.
  - **REQ-DIP-5.3.3-03-04-02**: The `provider` element may contain an `identifiers` element.
    - **REQ-DIP-5.3.3-03-04-02-01 [CONDITIONAL]**: If present, the `identifiers` element shall contain one or more `identifier` elements.
    - **REQ-DIP-5.3.3-03-04-02-02**: Each `identifier` element shall **identify the ASIP using normalized identifier(s)**.
    - **REQ-DIP-5.3.3-03-04-02-03**: Each `identifier` element shall contain a `type` element in form of a Unique Resource Identifier (URI) according to IETF RFC 3986 [2], which indicates the type of the identifier.
    - **REQ-DIP-5.3.3-03-04-02-04**: Each `identifier` element shall contain a child `identifier` element, which structure is defined by the `type` element.

EXAMPLE: For the different types of identifiers listed in CIR (EU) 2024/2977 [i.7], the technical specification [i.20] of the European Commission has defined the following URIs which can be used in the `type` element:

- http://data.europa.eu/eudi/id/EORI-No - Economic Operator Registration and Identification Number (EORI-No) according to (EU) No 1352/2013 [i.1].
- `http://data.europa.eu/eudi/id/LEI` - Legal Entity Identifier (LEI) according to (EU) No 2022/1860 [i.6] and ISO 17442-1 [i.27].
- `http://data.europa.eu/eudi/id/EUID` - European Unique Identifier (EUID) according to (EU) 2020/2244 [i.3] and (EU) 2021/1042 [i.4].
- `http://data.europa.eu/eudi/id/VATIN` - Value Added Tax Identification Number (VATIN) according to the Council Directive 2006/112/EC [i.12].
- `http://data.europa.eu/eudi/id/TIN` - Taxpayer Identification Number (TIN).
- `http://data.europa.eu/eudi/id/Excise` - Excise Number according to Art. 2 (12) of the Council Regulation (EC) No 389/2012 [i.13].

- **REQ-DIP-5.3.3-03-04-03 [CONDITIONAL]**: In case the ASIP is a public sector body, the `provider` element may contain an `establishedByLaw` element, which allows to **identify the ASIP by its legal basis** upon which it has been established.
  - **REQ-DIP-5.3.3-03-04-03-01 [CONDITIONAL]**: If present, the `establishedByLaw` element shall be an object, which shall contain a `legislativeIdentifier` in form of a URI according to IETF RFC 3986 [2].
  - **REQ-DIP-5.3.3-03-04-03-02 [CONDITIONAL]**: If present, the `establishedByLaw` element may contain a `legalBasis` element to specify the legal basis more precisely.
- **REQ-DIP-5.3.3-03-04-04**: The `provider` element should either contain an `identifiers` element or an `establishedByLaw` element.

- **REQ-DIP-5.3.3-03-04-05**: The `provider` element may contain a `currentAddress` element.
  - **REQ-DIP-5.3.3-03-04-05-01 [CONDITONAL]**: If present, the `currentAddress` element shall contain the **current address** of the ASIP.
  - **REQ-DIP-5.3.3-03-04-05-02 [CONDITONAL]**: If present, the different parts of the address within the `currentAddress` may be separated by commas.

REQ-DIP-5.3.3-04: The `dataService` object may contain a `country` element.

- **REQ-DIP-5.3.3-04-01 [CONDITIONAL]**: If present, the `country` element shall specify the Member State associated with the authentic source responsible for the verification or retrieval of the attribute, expressed as a country code according to ISO 3166-1 [21].

REQ-DIP-5.3.3-05: The output of the `retrieve` query may contain additional elements.

5.4 Use of HTTP and response format selection

REQ-DIP-5.4-01: The DIP shall support query requests using the HTTP GET method as defined in clause 9.3.1 of IETF RFC 9110 [15].

REQ-DIP-5.4-02: The DIP shall support the `Accept` header according to clause 12.5.1 of IETF RFC 9110 [15] to allow that clients specify their preferences regarding the response media type.

REQ-DIP-5.4-03: The DIP shall support at least one of the following values for the `Accept` header:

- **REQ-DIP-5.4-03-01 [CONDITIONAL]**: If the `Accept` header value is set to `application/x-ebrs+xml`, the DIP shall provide a response based on the `QueryManager` interface specified in ISO 15000-3 [24]:2023 [24] and the response messages shall conform to the RegRep XML schemas for the Registry Information Model (ebRIM) and Registry Services (ebRegS).
- **REQ-DIP-5.4-03-02 [CONDITIONAL]**: If the `Accept` header value is set to `application/json`, the DIP shall provide a response according to the OpenAPI specification and the referenced JSON schemas provided in Annex A.

NOTE 1: These schemas correspond to the response data specified in clauses 5.2.3 and 5.3.3 above.

REQ-TSP-5.4-01: (Q)TSPs which act as HTTP clients in a request to the discover interface, should include the `Accept` HTTP header according to clause 12.5.1 of IETF RFC 9110 [15] in order to indicate the appreciated response media type.

REQ-TSP-5.4-02: (Q)TSPs which act as HTTP clients in a request to the discover interface, shall examine the content of the HTTP response body in addition to the HTTP status code to determine the presence of successful or error responses.

NOTE 2: There are situations where the web infrastructure can change the HTTP status code.

6 Authentic Source Interface

6.1 Authentic Source Interface based on HTTP / OAuth

6.1.1 I2 (Verify)

6.1.1.1 Verify Request

REQ-ASIP-6.1.1.1-01: The **(I2) Verify** interface based on HTTP / OAuth shall use the HTTP POST method as defined in clause 9.3.3 of IETF RFC 9110 [15], as described in the following and specified in the OpenAPI specification in Annex B.

REQ-ASIP-6.1.1.1-02: The `POST /verify` shall contain a `verifyRequest` object, which shall specify which attributes and/or attribute fragments are requested to be verified.

REQ-ASIP-6.1.1.1-03: The `verifyRequest` object may contain an `attributes` property.

- **REQ-ASIP-6.1.1.1-03-01 [CONDITONAL]:** If present, the `attributes` property shall contain one or more `attribute` objects where each `attribute` object shall represent an attribute which is requested to be verified in its entirety.

REQ-ASIP-6.1.1.1-04: The `verifyRequest` object may contain an `attributeFragments` property.

- **REQ-ASIP-6.1.1.1-04-01 [CONDITONAL]**: If present, the `attributeFragments` property shall contain one or more `attributeFragment` objects where each `attributeFragment` object shall represent a fragment of an attribute where the fragment is identified using the JSONPath language according to IETF RFC 9535 [17].

REQ-ASIP-6.1.1.1-05: The `verifyRequest` object shall contain `attributes` or `attributeFragments` or both.

REQ-ASIP-6.1.1.1-06: An `attribute` object shall contain one `attributeIdentifier` and one `attributeValue` property:

- **REQ-ASIP-6.1.1.1-06-01**: The `attributeIdentifier` property shall be the **unique identifier** of the attribute in form of a Unique Resource Identifier (URI) according to IETF RFC 3986 [2] and shall contain the namespace, the local identifier and the version of the attribute as required in Article 7 (5) of CIR (EU) 2025/1569 [i.11].

- **REQ-ASIP-6.1.1.1-06-02**: The `attributeValue` property shall contain the **value** of the attribute, which is requested to be verified in its entirety.

  - **REQ-ASIP-6.1.1.1-06-02-01:** The `attributeValue` property shall contain the attribute encoded in a JSON-object according to IETF RFC 8259 [12], which conforms to the JSON-schema [i.30], which has been registered for the specific attribute type in the catalogue of attributes according to Article 2 (3) of CIR (EU) 2025/1569 [i.11].

REQ-ASIP-6.1.1.1-07: An `attributeFragment` object shall contain one `attributeIdentifier` property, one `location` property and one `value` property.

- **REQ-ASIP-6.1.1.1-07-01**: The `attributeIdentifier` property shall be the **unique identifier** of the attribute in form of a Unique Resource Identifier (URI) according to IETF RFC 3986 [2].

- **REQ-ASIP-6.1.1.1-07-02**: The `location` property shall contain a JSONPath expression according to IETF RFC 9535 [17], which shall specify the **location** of the attribute fragment within the complete attribute.

- **REQ-ASIP-6.1.1.1-07-03**: The `value` property shall contain the **value** of the JSON-based attribute fragment according to IETF RFC 8259 [12].

REQ-ASIP-6.1.1.1-08: The ASIP shall support the verification of full attributes using the `attributes` property.

- **REQ-ASIP-6.1.1.1-08-01**: The ASIP may support the verification of fragments using `attributeFragments`.

REQ-ASIP-6.1.1.1-09: If the ASIP receives a request for the verification of attribute fragments and does not support this optional functionality, it shall respond with an HTTP code 501 (Not Implemented).

REQ-ASIP-6.1.1.1-10: The ASIP may support fuzzy matching, meaning it considers attribute values or fragments included in the attribute as matching that differ in transliteration, blank spaces, hyphenation, concatenation, and similar orthographic variations from the corresponding attribute value held by the authentic source as matching.

- **REQ-ASIP-6.1.1.1-10-01 [CONDITONAL]:** In case the ASIP supports fuzzy matching, the ASIP shall return the attribute value or attribute fragment value stored in the authentic source in the `verifyResponse` element specified below. The ASIP shall return an `attributeValue` element within the `attributeVerificationResult` element, if a complete attribute was verified, and it shall return a `fragmentValue` element within a `fragmentVerificationResult`, if the verification of an attribute fragment was requested, and this option is supported by the ASIP.

NOTE 1: The decision whether the requesting provider uses the attribute value provided in the verification query or the matching attribute value held by the authentic source for issuing electronic attestations of attributes, is out of scope of the present specification.

NOTE 2: Article 2 of CIR (EU) 2025/846 [i.9] specifies general requirements for the unequivocal identity matching for natural persons in online cross-border services offered by or on behalf of a public sector body. Article 2 (6) of CIR (EU) 2025/846 [i.9] stipulates that the unequivocal identity matching "shall, to the extent possible, not be affected by differences in transliteration, blank spaces, hyphenation, concatenation, and similar orthographic variations that are required under Union law or national law of the Member State".

REQ-ASIP-6.1.1.1-11 [CONDITONAL]: In case the verification request contains attributes or attribute fragments of another data subject different from the user, the `verifyRequest` object may contain a `mandate` property, which clarifies that the user is allowed to do so.

REQ-ASIP-6.1.1.1-12: The ASIP may support the handling verification requests with `mandate` properties.

REQ-ASIP-6.1.1.1-13 [CONDITIONAL]: If the ASIP receives a verification request with a `mandate` property, which is not supported, it shall return an HTTP code 501 (Not Implemented).

NOTE 3: The specification of details for the `mandate` property and its handling by the ASIP is beyond the scope of the present specification.

6.1.1.2 Verify Response

REQ-ASIP-6.1.1.2-01: The ASIP shall return the result of a `/verify` call in a `verifyResponse` object, which structure shall be as specified in this clause.

REQ-ASIP-6.1.1.2-02: The `verifyResponse` object may contain an `attributeVerificationResults` element.

- **REQ-ASIP-6.1.1.2-02-01 [CONDITIONAL]**: If present, the `attributeVerificationResults` element shall contain an `attributeVerificationResult` element for each of the `attribute` elements provided in the `attributes` child element of the `verifyRequest` specified in clause 6.1.1.1.

REQ-ASIP-6.1.1.2-03: The `attributeVerificationResult` shall contain the `attributeIdentifier` element, which shall be the **unique identifier** of the attribute in form of a Unique Resource Identifier (URI) according to IETF RFC 3986 [2], as provided in the request.

REQ-ASIP-6.1.1.2-04: The `attributeVerificationResult` shall contain an `attributeVerificationResult` child element, which shall indicate the **result of the verification of the attribute** and which shall contain a Unique Resource Identifier (URI) according to IETF RFC 3986 [2] with exactly one of the following four potential values:

1) `http://uri.etsi.org/19478/VerificationResult/Match` - indicates that the value held for this user by the authentic source exactly matches the value claimed by the (Q)TSP.

2) `http://uri.etsi.org/19478/VerificationResult/NoMatch` - indicates that the value held for this user by the authentic source does not match the value claimed by the (Q)TSP.

3) `http://uri.etsi.org/19478/VerificationResult/MatchWithVariation` - indicates that the value held for this user by the authentic source matches the value claimed by the (Q)TSP with an admissible orthographic variation in the sense of REQ-ASIP-6.1.1.1-10 above.

4) `http://uri.etsi.org/19478/VerificationResult/Unknown` - indicates that no authentic source data was available to determine a match for the attribute for the user.

- **REQ-ASIP-6.1.1.2-04-01 [CONDITIONAL]**: In case the `attributeVerificationResult` is either `http://uri.etsi.org/19478/VerificationResult/Match` or `http://uri.etsi.org/19478/VerificationResult/MatchWithVariation` the `attributeVerificationResult` shall contain an `attributeValue` element.

- **REQ-ASIP-6.1.1.2-04-02 [CONDITIONAL]**: In case the returned `attributeVerificationResult` is `http://uri.etsi.org/19478/VerificationResult/Match`, the `attributeValue` element shall contain the value as in the request.

- **REQ-ASIP-6.1.1.2-04-03 [CONDITIONAL]**: In case the returned `attributeVerificationResult` is `http://uri.etsi.org/19478/VerificationResult/MatchWithVariation`, the `attributeValue` element shall contain the value of the attribute as stored in the authentic source.

REQ-ASIP-6.1.1.2-05 [CONDITIONAL]: If the ASIP supports the verification of attribute fragments and if the `verifyResponse` object is present, it shall contain a `fragmentVerificationResults` element, which shall contain a `fragmentVerificationResult` element for each of the `attributeFragment` elements provided in the `attributeFragments` child element of the `verifyRequest` specified in clause 6.1.1.1.

NOTE 1: See REQ-ASIP-6.1.1.1-09 for the case that the functionality is not supported.

REQ-ASIP-6.1.1.2-06 [CONDITIONAL]: If present, each `fragmentVerificationResult` element shall, contain the two child properties `attributeValue` and `fragmentVerificationResult` and may contain the `fragmentValue` property.

- **REQ-ASIP-6.1.1.2-06-01**: The `attributeIdentifier` element shall be the **unique identifier** of the attribute in form of a Unique Resource Identifier (URI) according to IETF RFC 3986 [2], as provided in the request.

- **REQ-ASIP-6.1.1.2-06-02**: The `fragmentVerificationResult` child element shall indicate the **result of the verification of the attribute fragment** with exactly one of the four potential values as specified in REQ-ASIP-6.1.1.2-04 above.

- **REQ-ASIP-6.1.1.2-06-03**: The `fragmentValue` child element shall have the structure as defined in REQ-ASIP-6.1.1.1-07 above.

- **REQ-ASIP-6.1.1.2-06-04**: The child elements `attributeIdentifier` and `location` of the `attributeFragment` element shall be identical to the corresponding elements in the request.

  - **REQ-ASIP-6.1.1.2-06-04-01 [CONDITIONAL]**: In case the `fragmentVerificationResult` is either `http://uri.etsi.org/19478/VerificationResult/Match` or `http://uri.etsi.org/19478/VerificationResult/MatchWithVariation` the `attributeVerificationResult` shall contain a `fragmentValue` element.

  - **REQ-ASIP-6.1.1.2-06-04-02 [CONDITIONAL]**: In case the returned `fragmentVerificationResult` is `http://uri.etsi.org/19478/VerificationResult/Match`, the `fragmentValue` element shall contain the value as in the request.

  - **REQ-ASIP-6.1.1.2-06-04-03 [CONDITIONAL]**: In case the returned `fragmentVerificationResult` is `http://uri.etsi.org/19478/VerificationResult/MatchWithVariation`, the `fragmentValue` element shall contain the value of the fragment of the attribute as stored in the authentic source.

REQ-ASIP-6.1.1.2-07: The `verifyResponse` object, shall contain a `provider` element to identify the ASIP. This element shall have the structure as specified in REQ-DIP-5.3.3-03-04 (see clause 5.3.3).

REQ-ASIP-6.1.1.2-08 [CONDITIONAL]: If and only if the ASIP is different from the authentic source the `verifyResponse` object shall contain an `authenticSource` element, which identifies the authentic source, which is responsible for the verification procedure.

REQ-ASIP-6.1.1.2-09 [CONDITIONAL]: If present, the `authenticSource` element shall have the same structure as the `provider` element specified in REQ-DIP-5.3.3-03-04 (see clause 5.3.3).

REQ-ASIP-6.1.1.2-10: The `verifyResponse` object may contain a `mandateResult` property.

REQ-ASIP-6.1.1.2-11 [CONDITIONAL]: In case the `verifyRequest` contained a `mandate` property and this functionality is supported by the ASIP (see REQ-ASIP-6.1.1.1-11), the `verifyResponse` object shall contain a `mandateResult`.

NOTE 2: The specification of details for the `mandate` property is beyond the scope of the present specification.

REQ-ASIP-6.1.1.2-12: The ASIP shall indicate the overall result of this operation by the resulting HTTP status code according to clause 15 of IETF RFC 9110 [15], which shall be used as follows:

- 200 (OK) - shall indicate that the submitted attribute verification request was successfully performed by the verification service. The detailed verification results shall be conveyed in the returned `verifyResponse` element.
- 400 (Bad Request) - shall indicate that the submitted content was malformed.
- 401 (Unauthorized) - shall indicate that the request was not successful due to an authentication or authorization error.
- 404 (Not Found) - shall indicate that the verification of the attribute was not successful.
- 501 (Not Implemented) - shall indicate that the requested optional feature is not implemented.

6.1.2 I3 (Retrieve)

6.1.2.1 Retrieve Request

REQ-ASIP-6.1.2.1-01: The **(I3) Retrieve** interface based on HTTP / OAuth shall use the HTTP POST method as defined in clause 9.3.3 of IETF RFC 9110 [15], as described in the following and specified in the OpenAPI specification in Annex B.

REQ-ASIP-6.1.2.1-02: The `POST /retrieve` shall contain a `retrieveRequest` object, which shall contain an `attributeIdentifiers` property.

REQ-ASIP-6.1.2.1-03: The `attributeIdentifiers` property shall contain one or more `attributeIdentifier` properties.

- **REQ-ASIP-6.1.2.1-03-01:** Each `attributeIdentifier` property shall specify the **unique identifier** of an attribute, which is requested to be retrieved from the authentic source.

NOTE 1: The `search` query within the **(I1) Discover** interface specified in clause 5.2 allows to find the unique identifier of a specific attribute within the catalogue of attributes and the `retrieve` query specified in clause 5.3 allows to retrieve the endpoints of the corresponding data service for the retrieval of an attribute from an authentic source.

REQ-ASIP-6.1.2.1-04: The `retrieveRequest` object may contain a `mandate` property, which can be used to request the retrieval of attributes on behalf of another data subject different from the user.

NOTE 2: The specification of details for this property and its handling by the ASIP is beyond the scope of the present specification.

6.1.2.2 Retrieve Response

REQ-ASIP-6.1.2.2-01: The ASIP shall return the result of a `/retrieve` call in a `retrieveResponse` object, which shall be structured as specified in this clause.

REQ-ASIP-6.1.2.2-02 [CONDITONAL]: If the retrieve request was successful, the `retrieveResponse` object shall contain an `attributes` property.

- **REQ-ASIP-6.1.2.2-02-01 [CONDITONAL]**: If present, the `attributes` property shall contain one or more `attribute` objects, which shall be structured as specified in REQ-ASIP-6.1.1.1-06 (see clause 6.1.1.1).

REQ-ASIP-6.1.2.2-03: The `retrieveResponse` object, shall contain a `provider` element, which shall identify the ASIP and which shall be structured as specified in REQ-DIP-5.3.3-03-04 (see clause 5.3.3).

REQ-ASIP-6.1.2.2-04 [CONDITIONAL]: If and only if the ASIP is different from the authentic source, the `retrieveResponse` object, shall contain an `authenticSource` element, which shall identify the authentic source, which holds the attribute value and which shall have the same structure as the `provider` element specified in REQ-DIP-5.3.3-03-04 (see clause 5.3.3).

REQ-ASIP-6.1.2.2-05: The `retrieveResponse` object, may contain a `mandateResult` property.

- **REQ-ASIP-6.1.2.2-05-01 [CONDITIONAL]**: If the `retrieveRequest` contained a `mandate` property (see REQ-ASIP-6.1.2.1-04), the `retrieveResponse` object shall contain a `mandateResult`.

NOTE: The specification of details for the `mandateResult` property is beyond the scope of the present specification.

REQ-ASIP-6.1.2.2-06: The ASIP shall indicate the overall result of this operation by the resulting HTTP status code according to clause 15 of IETF RFC 9110 [15], which shall be used as follows:

- 200 (OK) - shall indicate that the request for retrieval of the requested attributes was successfully.
- 400 (Bad Request) - shall indicate that the submitted content was malformed.
- 401 (Unauthorized) - shall indicate that the request was not successful due to an authentication or authorization error.
- 404 (Not Found) - shall indicate that at least one of the requested attributes, identified by a URI within the request, was not found.
- 501 (Not Implemented) - shall indicate that the requested optional feature is not implemented.

6.1.3 I4 (Authorize)

6.1.3.1 Overview and general requirements

REQ-AZSP-6.1.3.1-01: The authorization of requests for verification and retrieval in the HTTP based authentic source interface according to IETF RFC 9110 [15], as specified in clauses 6.1.1 and 6.1.2 above, shall be realized with the OAuth 2.0 based authorization framework according to IETF RFC 6749 [6] using the authorization code flow, as specified in clause 4.1 of IETF RFC 6749 [6], and related standards as specified below.

NOTE 1: The (qualified) trust service provider is acting as *client*, the ASIP is acting as *resource server* and the user is acting as *resource owner*, who grants access to the protected resource offered by the ASIP. See also note 1 in clause 4.4.

REQ-AZSP-6.1.3.1-02: The authorization server shall identify the user.

NOTE 2: The identification of the user is required by Article 45e of the eIDAS-Regulation [i.33] and Article 9 (3) of CIR (EU) 2025/1569 [i.11].

REQ-AZSP-6.1.3.1-03: The authorization server may perform the necessary authentication and identification with the EUDI-Wallet using the OpenID4VP protocol [i.32], or an alternative electronic identification procedure according to ETSI TS 119 461 [i.15].

REQ-AZSP-6.1.3.1-04: The authorization server shall support the Proof Key for Code Exchange (PKCE) mechanism according to IETF RFC 7636 [11].

REQ-TSP-6.1.3.1-01: The (qualified) trust service provider shall use the PKCE mechanism according to IETF RFC 7636 [11].

NOTE 3: See also clauses 2.1.1 and 4.5.3.1 of IETF RFC 9700 [19] and clause 5.3.2.2 of the FAPI 2.0 Security Profile [28].

REQ-AZSP-6.1.3.1-05: The authorization server shall either mandate Mutual Transport Layer Security (MTLS) according to clause 2 of IETF RFC 8705 [13], or the `private_key_jwt` mechanism according to clause 9 of the OpenID Connect specification [29], or both for authenticating the clients.

NOTE 4: The `private_key_jwt` mechanism according to clause 9 of the OpenID Connect specification [29] is based on the OAuth 2.0 Assertion Framework specified in IETF RFC 7521 [i.23] and the JSON Web Token (JWT) profile for OAuth 2.0 Client Authentication according to IETF RFC 7523 [i.24].

NOTE 5: See also clause 5.3.3.1 of the FAPI 2.0 Security Profile [28].

REQ-TSP-6.1.3.1-02: The (qualified) trust service provider shall use the client authentication mechanism mandated by the authorization server according to REQ-AZSP-6.1.3.1-05.

REQ-AZSP-6.1.3.1-06: The authorization server shall support sender-constrained access tokens using mutually authenticated TLS according to IETF RFC 8705 [13], or the Demonstrating Proof of Possession (DPoP) mechanism according to IETF RFC 9449 [16].

REQ-TSP-6.1.3.1-03: The (qualified) trust service provider shall use the access token protection mechanism supported by the authorization server according to REQ-AZSP-6.1.3.1-06.

REQ-AZSP-6.1.3.1-07: The authorization server shall support dynamic client registration according to IETF RFC 7591 [10], as specified in clause 6.1.3.2 below in more detail.

REQ-AZSP-6.1.3.1-08: The authorization server shall use JWT-based access tokens according to IETF RFC 9068 [14], which shall contain the required identification data of the user in form of a JWT according to IETF RFC 7519 [9].

NOTE 6: While using OAuth 2.0 token introspection according to IETF RFC 7662 [i.25] would be a technically conceivable alternative to the use of JWT-based access tokens, this approach would not fulfil the legal requirement of Article 9 (3) of CIR (EU) 2025/1569 [i.11], which stipulates, that the "verification request shall set out the attributes and the identification data of the subject of the attribute for which the qualified trust service provider requests the verification".

6.1.3.2 Dynamic Client Registration (IETF RFC 7591)

REQ-AZSP-6.1.3.2-01: The authorization server shall support dynamic client registration with a sealed software statement according to clause 3.1.1 of IETF RFC 7591 [10].

REQ-TSP-6.1.3.2-01: The (qualified) trust service provider shall use an X.509 based certificate according to IETF RFC 5280 [5], which supports the creation of electronic signatures or seals, and hence can be used for creating JWTs for registration and client authentication, for the creation of a signed or sealed software statement, which is submitted within the dynamic client registration procedure according to clause 3.1.1 of IETF RFC 7591 [10].

REQ-TSP-6.1.3.2-02: The X.509 based certificate according to IETF RFC 5280 [5] shall contain information, which allows to identify the (qualified) trust service provider.

REQ-TSP-6.1.3.2-03: The X.509 based certificate according to IETF RFC 5280 [5] should contain information, which enables the authorization server to perform an automated check, that the client is indeed a (qualified) trust service provider.

NOTE 1: While there would be different options for the verification of the identity and the entitlement of the (qualified) trust service provider, the present document recommends to use Wallet-Relying Party Access Certificates (WRPAC) according to ETSI TS 119 411-8 [i.14] for this purpose, if possible.

REQ-AZSP-6.1.3.2-02: The authorization server should accept Wallet-Relying Party Access Certificates (WRPAC) according to ETSI TS 119 411-8 [i.14] for the dynamic client registration procedure and should support the lookup of entitlement information in the applicable national register of wallet-relying parties.

NOTE 2: WRPAC contain the necessary identification data of the (qualified) trust service provider and the entitlement can be easily verified in an automated manner using the national register of wallet-relying parties according to Article 2 (10) and Article 3 of CIR (EU) 2025/848 [i.10] using the API according to the corresponding technical specification of the Commission [i.21].

NOTE 3: Using the WRPAC of an intermediary is not further considered here.

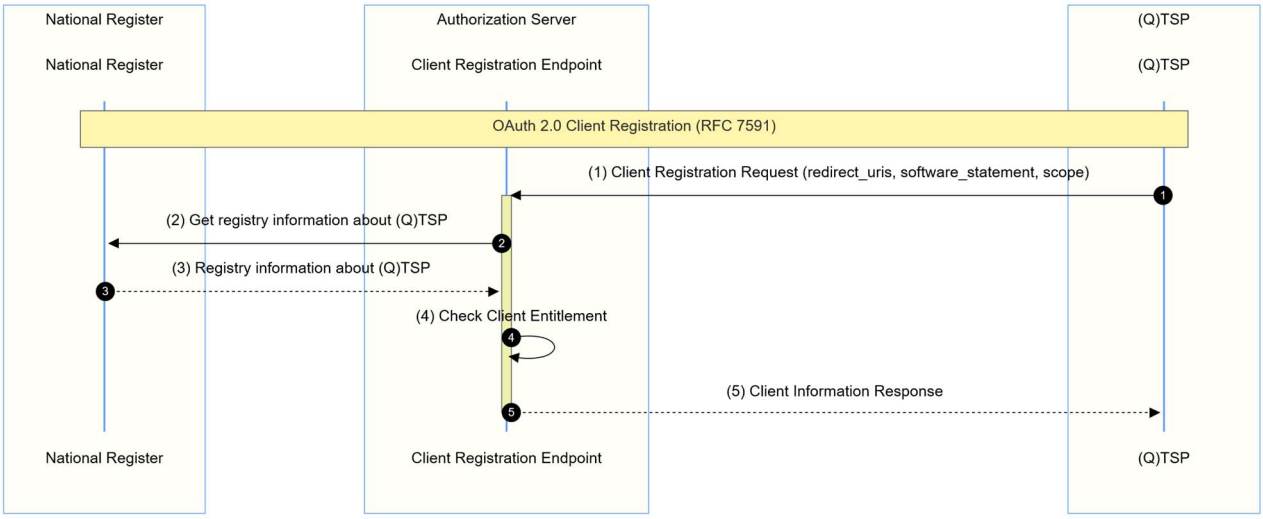

Figure 2: Dynamic Client Registration using IETF RFC 7591 [10] (informative)

The registration procedure for a client is depicted in the informative Figure 2 and explained in the following. It provides an informative illustration of requirement REQ-AZSP-6.1.3.2-01:

1) The (Q)TSP sends the signed or sealed client registration request according to clause 3.1.1 of IETF RFC 7591 [10] to the client registration endpoint of the authorization server, which contains the `redirect_uris`, the `software_statement` and the `scope`, which can comprise `verify`, `retrieve` or both. The `software_statement` contains the claim `registryURI`, which has been provided to the (Q)TSP upon registration in the national register according to Article 2 (10) and Article 3 of CIR (EU) 2025/848 [i.10].

2) The `registryURI` claim can be used by the authorization server to look up the relevant information about the (Q)TSP with respect to its entitlement using the API defined for this purpose in the technical specification of the Commission [i.21]. The information retrieved from the national register contains the `entitlement` element and possibly additional information which is deemed relevant for the registration of the client. The entitlement will be verified within the initial client registration and it will be verified again in regular intervals determined by the applicable policy.

3) In this step the requested information is returned from the national register to the authorization server.

4) The authorization server checks the entitlement and registers the (Q)TSP as client in case of success.

5) The authorization server sends a response to the client as specified in clause 3.2 of IETF RFC 7591 [10].

6.1.3.3 Authorization Code Flow (IETF RFC 6749)

The overall process for the issuance of an electronic attestation using the authorization code flow according to clause 4.1 of IETF RFC 6749 [6] is depicted in the informative Figure 3 below.

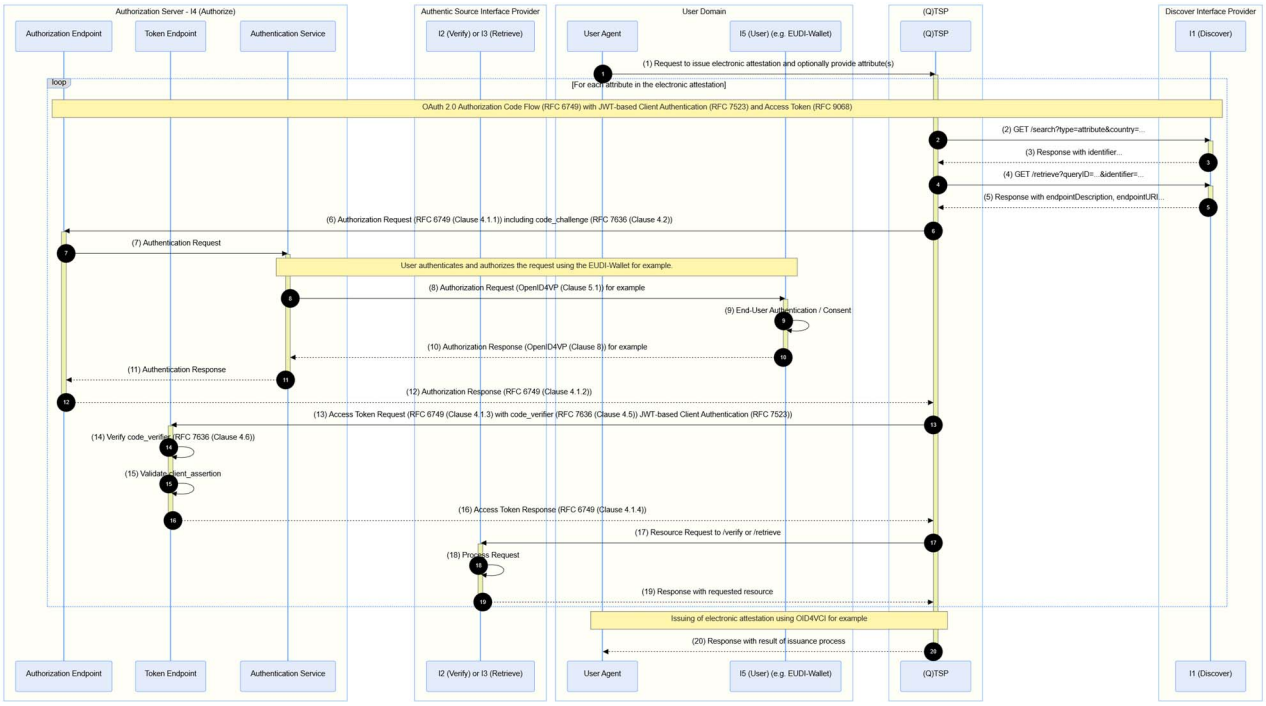

Figure 3: Overall process with authorization code flow using IETF RFC 6749 [6] (informative)

1) In the first step, the user requests the issuance of some electronic attestation at a (Q)TSP. If it is not possible to retrieve the attributes from the ASIP, the user will manually enter the attributes under consideration, such that they can be verified in step (18).

2) The (Q)TSP uses the **(I1) Discover** interface provided by the DIP to look up the required metadata about attributes registered in the catalogue of attributes and related data services for verification or retrieval. In a first step it will use the `search` function as specified in clause 5.2 to determine the `attributeIdentifier` of the attribute and additional metadata.

3) The requested information as specified in clause 5.2.3 is returned.

4) The (Q)TSP uses the **(I1) Discover** interface a second time to determine metadata about the available endpoints for verification or retrieval of attributes. As specified in clause 5.3, the `retrieve` function will essentially get the applicable meta data of the data service for verification or retrieval.

5) The requested information as specified in clause 5.3.3 is returned. This can include the `endpointDescription`, which in turn includes a reference to the endpoints of the authorization server in charge, and the `endpointURI` of the required verification or retrieval service, which is invoked in step (17).

6) The (Q)TSP sends an authorization request according to clause 4.1.1 of IETF RFC 6749 [6], which includes the `code_challenge` according to IETF RFC 7636 [11] to the authorization endpoint of the authorization server.

7) The authorization endpoint relays the request to the authentication service integrated in or attached to the authorization server, which will take care of the required authentication and identification of the user and the authorization of the later resource request by the user. The involvement of the user is required by Article 45e of Regulation (EU) No 910/2014 [i.33] and according to Article 9 (3) of CIR (EU) 2025/1569 [i.11] the request to the verification service needs to include the identification data of the user. The authentication, identification and authorization procedure can be implemented using the EUDI-Wallet or another suitable identity proofing mechanism according to ETSI TS 119 461 [i.15].

8) In this step, the authentication service performs the authentication and identification of the user. If the EUDI-Wallet is used for this purpose, the authentication service sends an authorization request according to clause 5.1 of the OpenID4VP specification [i.32] to the EUDI-Wallet.

9) The user approves the request and provides its person identification data. If the EUDI-Wallet is used for this purpose, the person identification data will be in form of a verifiable presentation.

10) In this step, the person identification data of the user will be returned to the authentication service. If the EUDI-Wallet is used for authentication and identification, it will return the authorization response according to clause 8 of the OpenID4VP specification [i.32], which includes the required person identification data.

11) The authentication service forwards the authentication response to the authorization endpoint.

12) The authorization endpoint answers the request from step (6) and returns an authorization response according to clause 4.1.2 of IETF RFC 6749 [6] to the (Q)TSP.

13) The (Q)TSP sends an access token request according to clause 4.1.3 of IETF RFC 6749 [6] to the token endpoint of the authorization server, which involves a suitable client authentication as specified in line with REQ-AZSP-6.1.3.1-05.

14) In this step the authorization server verifies the `code_verifier` according to clause 4.6 of IETF RFC 7636 [11].

15) In this step the authorization server validates the JWT-based `client_assertion`.

16) If the verification step (14) and the validation step (15) is successful, the authorization server creates a JWT-based access token according to IETF RFC 9068 [14], which contains the identification data obtained in step (10).

17) In this step the resource request to the verification service using the **I2 (Verify)** interface defined in clause 6.1.1, or the retrieval service using the **I3 (Retrieve)** interface as defined in clause 6.1.2, is invoked using the access token obtained in the previous step.

18) The ASIP is processing the verification or retrieval request.

19) The result of the request is returned in this step as specified in clauses 6.1.1 and 6.1.2 respectively.

20) After the issuing of the requested electronic attestation using OpenID4VCI [i.31] for example, the user is informed about the result of the issuance process.

6.1.3.4 Security considerations for the Authorization Server

REQ-AZSP-6.1.3.4-01: The authorization server shall take into account the security considerations related to OAuth 2.0 from IETF RFC 6819 [7], IETF RFC 7636 [11], IETF RFC 9700 [19] and the FAPI 2.0 Security Profile [28].

REQ-AZSP-6.1.3.4-02: The authorization server shall be operated in a trustworthy environment, which shall be embedded in an information security management system according to ISO/IEC 27001 [25] for example.

6.2 Authentic Source Interface based on ISO 15000

6.2.1 Overview

The following clauses specify the general protocol binding for the authentic source interface in clause 6.2.2, which is based on the relevant parts of the ISO 15000 standard, which includes ISO 15000-2 [23] and ISO 15000-3 [24].

REQ-ASIP-6.2.1-01: The **(I2) Verify** interface based on ISO 15000 shall be as specified in clause 6.2.3.

REQ-ASIP-6.2.1-02 [CONDITIONAL]: The **I3 (Retrieve)** interface based on ISO 15000 shall be as specified in clause 6.2.4, if it is supported.

The authentic source interface follows the request-response pattern and involves actors in the following two roles:

- **requester**
- **provider**

and it allows to:

- **verify** attributes using the **I2 (Verify)** interface as specified in clause 6.2.3; and

- **retrieve** attributes using the **I3 (Retrieve)** interface as specified in clause 6.2.4, if this functionality is supported.

The request contains data identifying the data subject and the attributes to be verified or retrieved as a set of attribute value pairs described by an attribute schema and some additional data as specified in clauses 6.2.3.1 and 6.2.4.2 below.

There are three types of responses, where the result type is expressed using the `status` attribute:

- A **successful** response is used in case the verification provider processed the request and no error occurred while processing the request. The successful response for the verify call is specified in clause 6.2.3.2.1 below. For an attribute verification query, the response indicates that the provider completed the matching for the query and determined whether the query data matches data held by the authentic source, does not match the data held by the authentic source or whether the no authentic source has the attribute data for the user.

- A **failure error** response is used in failure situations including situations where an error occurred while processing the request. It is specified for the verify call in clause 6.2.3.2.2 below. The failure error differs from the successful response in that no match result is available for the user for the attribute due to an error.

- An **unavailable** response indicates that the response is not yet available. This can be the case where the response is an immediate response to an asynchronous request, but an actual response is not yet available and could be provided later. In this case, the actual response is obtained in multiple request/response flows as specified in clause 6.2.3.2.3 below.

This verification interface is defined as an instance of the **QueryManager** interface defined in ISO 15000-3 [24]. The messages conform to the Regrep XML schemas for the Registry Information Model (ebRIM) and the Registry Services (ebRS).

To exchange messages in the authentic source interface, the protocol binding specified in clause 6.2.2 below is used.

Personal identification data is expressed using the *NaturalPerson*, *LegalPerson*, *NaturalPersonRepresentingNaturalPerson*, and *NaturalPersonRepresentingLegalPerson* elements defined in the XML schema specified in clause C.2. Details of the verification request or response message content are expressed using the `VerificationQueryDetails`, and `VerificationResponseDetails` elements defined in the XML schema specified in clause C.3 and the RIM binding elements defined in the XML schema specified in clause C.4.

6.2.2 Protocol Binding

REQ-ASIP-6.2.2-01: All messages in the authentic source interface based on ISO 15000 shall be exchanged using the OASIS ebXML Messaging Protocol Binding for RegRep Version 1.0 [26], which defines the use of ISO 15000-2 [23] as message exchange protocol for ISO 15000-3 [24] (RegRep) exchanges.

REQ-ASIP-6.2.2-02: For interoperability, a provider of an implementation of the authentic source interface or a community of providers may mandate support for additional implementation guidelines.

REQ-ASIP-6.2.2-03 [CONDITIONAL]: In case where additional guidelines are mandated for a specific context, as in REQ-ASIP-6.2.2-02, then all requestors within this context shall follow these guidelines.

6.2.3 I2 (Verify)

6.2.3.1 Verify Request

REQ-ASIP-6.2.3.1-01: The verification request shall be an XML document rooted in `query:QueryRequest`.

- **REQ-ASIP-6.2.3.1-01-01**: The `query:QueryRequest` shall have an attribute named `id` with a UUID value according to [18].

- **REQ-ASIP-6.2.3.1-01-02**: The `query:QueryRequest` shall have the following six named `rim:Slot` child elements:

  1) `SpecificationIdentifier`;

  2) `IssueDateTime`;

3) `Requester`;

4) `Provider`;

5) `IssuingPurposeQEAA`; and

6) `ExplicitRequestGiven`.

- **REQ-ASIP-6.2.3.1-01-02-01**: `SpecificationIdentifier` shall be of type *rim:StringValueType*, and shall be set to the fixed value "`https://uri.etsi.org/19478/v1.1.1`".

- **REQ-ASIP-6.2.3.1-01-02-02**: `IssueDateTime` shall be of type *rim:DateTimeValueType* and shall be set to the date and time the request was issued in the date and time format according to [22].

- **REQ-ASIP-6.2.3.1-01-02-03**: `Requester` shall identify the person making the request and shall be of type *NaturalPersonValueType*, in case the verification requester is a natural person, or *LegalPersonValueType*, in case the verification requester is a legal person.

NOTE 1: *NaturalPersonValueType* and *LegalPersonValueType* are defined in the XML-schema provided in clause C.2.

- **REQ-ASIP-6.2.3.1-01-02-04**: `Provider` shall identify the ASIP to which the request is issued and shall be of type *NaturalPersonValueType*, in case the ASIP is a natural person, or *LegalPersonValueType*, in case the ASIP is a legal person.

- **REQ-ASIP-6.2.3.1-01-02-05**: `IssuingPurposeQEAA` shall be of type *rim:BooleanValueType* and shall indicate whether the request is made to be able to issue a QEAA in case the verification yields a positive result.

- **REQ-ASIP-6.2.3.1-01-02-06**: `ExplicitRequestGiven` shall be of type *rim:BooleanValueType* and shall indicate whether the request is at the explicit request of a user of the verification service.

REQ-ASIP-6.2.3.1-02 [CONDITIONAL]: If the request is issued to obtain a deferred response to a previously issued request (see clause 6.2.3.2.3 below), it shall contain the additional slot `DeferredResponseIdentifier`.

- **REQ-ASIP-6.2.3.1-02-01 [CONDITIONAL]:** If present, the `DeferredResponseIdentifier` slot shall be of type *rim:StringValueType* and shall be set to the value of the `ResponseIdentifier` issued in the response with unavailable status to the original request.

These slots apply to all attributes for which verification is requested.

REQ-ASIP-6.2.3.1-03: The `query:QueryRequest` shall contain a `query:Query` element, which shall be of the fixed type "`VerificationQuery`" and which shall have the two `rim:Slot` child elements `Person` and `VerificationQueryDetails`.

- **REQ-ASIP-6.2.3.1-03-01**: The `Person` element shall identify the person which is the subject of the attribute on which behalf the verification request is issued and shall be of the abstract type *PersonType*, which provides the basis for the *NaturalPersonType*, *LegalPersonType*, *NaturalPersonRepresentingNaturalPersonValueType* and *NaturalPersonRepresentingLegalPersonValueType*.

NOTE 2: *PersonType*, *NaturalPersonType*, *LegalPersonType*, *NaturalPersonRepresentingNaturalPersonValueType* and *NaturalPersonRepresentingLegalPersonValueType* are defined in the XML-schema provided in clause C.2.

- **REQ-ASIP-6.2.3.1-03-02**: The `VerificationQueryDetails` shall provide details of the attributes for which verification is requested and shall be of type *VerificationsQueryDetailsType*, as defined in the XML schema specified in clause C.3.

- **REQ-ASIP-6.2.3.1-03-03**: The content of the `rim:SlotValue` element, which corresponds to the `VerificationQueryDetails` element, shall be of type *VerificationQueryDetailsValueType* defined in the XML schema specified in clause C.4, which enables the inclusion of a `VerificationsQueryDetails` element.

REQ-ASIP-6.2.3.1-04: For each attribute for which verification is requested, the `VerificationQueryDetails` element shall contain a separate `AttributeVerificationQuery` element.

- **REQ-ASIP-6.2.3.1-04-01**: All `AttributeVerificationQuery` elements in a `VerificationsQueryDetails` element shall relate to the same `Person` element.

REQ-ASIP-6.2.3.1-05: An `AttributeVerificationQuery` shall have a child element `AttributeIdentifier`, which shall be of type *xs:anyURI* and shall contain the identifier of the attribute for which values are requested to be verified in form of a URI according to IETF RFC 3986 [2].

REQ-ASIP-6.2.3.1-06: An `AttributeVerificationQuery` may contain a child element `Schema`.

- **REQ-ASIP-6.2.3.1-06-01 [CONDITIONAL]**: If it is present, a `Schema` element shall be of type *xs:anyURI* and shall contain the link to the data model schema of the attribute in the catalogue of attributes according to Article 2 (3) of CIR (EU) 2025/1569 [i.11], which defines the internal structure of the attribute, which is provided for verification.

REQ-ASIP-6.2.3.1-07: An `AttributeVerificationQuery` may contain a child element `SchemaMediaType`.

- **REQ-ASIP-6.2.3.1-07-01 [CONDITIONAL]**: If the `SchemaMediaType` element is present, it shall identify the media type according to IETF RFC 6838 [8] of the schema distribution.

- **REQ-ASIP-6.2.3.1-07-02 [CONDITIONAL]**: If the `AttributeVerificationQuery` element contains a `Schema` element, it shall also contain the `SchemaMediaType` element.

REQ-ASIP-6.2.3.1-08: An `AttributeVerificationQuery` shall include exactly one of a choice of the following three child elements:

- **REQ-ASIP-6.2.3.1-08-01 [CHOICE]**: `TextValue` shall, if present, contain the full data content of the attribute encoded as a string and shall be used with an attribute with JSON object content or with atomic text content.

  - **REQ-ASIP-6.2.3.1-08-01-01 [CONDITONAL]:** If the `TextValue` element is present with JSON content and `SchemaMediaType` is present, then `SchemaMediaType` shall be set to `application/json-schema` or `application/ld+json`.

  - **REQ-ASIP-6.2.3.1-08-01-02 [CONDITIONAL]**: If the `TextValue` element is present, it may have one or both of the following optional attributes:

    - `Encoding` - specifies, if present, that the text value content has been encoded to be included in the XML request. The only allowed value is "`base64`" to indicate that the content has been encoded using the base64 algorithm according to IETF RFC 4648 [3]. If the attribute is absent, no encoding has been applied.

    - `xml:lang` - specifies, if present, the language in which the text is provided, encoded as two letter language code according to ISO 639 [20].

- **REQ-ASIP-6.2.3.1-08-02 [CHOICE]**: `XMLValue` shall, if present, contain the full data content of the attribute encoded as XML element content. This value shall be used with an attribute with complex XML content.

  - **REQ-ASIP-6.2.3.1-08-02-01 [CONDITIONAL]**: If `XMLValue` is present and `SchemaMediaType` is present, then `SchemaMediaType` shall be set to `application/xml`.

- **REQ-ASIP-6.2.3.1-08-03 [CHOICE]**: `AttributeProperties` shall, if present, be used for verification of selected properties of an attribute.

Presence of each of these three elements is conditional to the absence of the other two.

REQ-ASIP-6.2.3.1-09 [CONDITIONAL]: If present, the `AttributeProperties` element shall have the child elements `ExpressionLanguage` and `AttributeProperty`.

- **REQ-ASIP-6.2.3.1-09-01**: The `ExpressionLanguage` element shall be present exactly once, it shall be of type *xs:anyURI* and it shall contain a URI according to IETF RFC 3986 [2], which specifies the query expression language used in the `QueryExpression` elements in the `AttributeProperty` child elements.
- **REQ-ASIP-6.2.3.1-09-02**: The present specification defines the following two URIs, which shall be used consistently with the `SchemaMediaType`:
  - **REQ-ASIP-6.2.3.1-09-02-01 [CONDITONAL]**: If the `SchemaMediaType` is equal to `application/xml`, the `ExpressionLanguage` element shall be the URI `https://www.w3.org/TR/xpath-31/` and it shall be used in the `ExpressionLanguage` element, in order to indicate that the query expression language is XPath 3.1 according to the W3C XPath specification [30].
  - **REQ-ASIP-6.2.3.1-09-02-02 [CONDITONAL]**: If the `SchemaMediaType` is equal to `application/json-schema` or `application/ld+json`, the `ExpressionLanguage` element shall be the URI `urn:ietf:rfc:9535` in order to indicate that the query expression language is JSONPath according to IETF RFC 9535 [17].
- **REQ-ASIP-6.2.3.1-09-03**: The `AttributeProperty` element shall be present one or more times, if the verification request is not for the full attribute, but for selected properties or other located substructures of the attribute.

REQ-ASIP-6.2.3.1-10: The `AttributeProperty` element shall have a child element `QueryExpression`, which shall contain a query expression locating a particular property of the attribute content. It shall express the path in the attribute structure from its root to the location of the value to be verified within the structure. The expression shall be expressed in the provided `ExpressionLanguage`.

REQ-ASIP-6.2.3.1-11: The `AttributeProperty` element shall include exactly one of a choice of the following two child elements:

- **REQ-ASIP-6.2.3.1-11-01 [CHOICE]**: `TextValue` shall, if present, contain the value against which the located property content is matched and shall be used for JSON property content or any atomic value property content. The `TextValue` element may have one or more of the two optional attributes `encoding` and `xml:lang`, which are specified above (see REQ-ASIP-6.2.3.1-08-01).
- **REQ-ASIP-6.2.3.1-11-02 [CHOICE]**: `XMLValue` shall, if present, contain the value of the located content as well-formed XML element content.
  - **REQ-ASIP-6.2.3.1-11-02-01:** If the `XMLValue` element is present and `SchemaMediaType` is present, then `SchemaMediaType` shall be equal to `application/xml`.

REQ-ASIP-6.2.3.1-12: The `QueryRequest` shall include a `query:ResponseOption` element which shall contain a `returnType` attribute set to the value "`RegistryObject`".

6.2.3.2 Verify Response

6.2.3.2.1 Successful Verify Response

REQ-ASIP-6.2.3.2.1-01: The successful verification response providing an immediate result shall be an XML document rooted in a `query:QueryResponse` element with `status` set to the value `urn:oasis:names:tc:ebxml-regrep:ResponseStatusType:Success`.

REQ-ASIP-6.2.3.2.1-02: The XML document shall have an attribute named `requestId` set to the value of the `id` attribute in the corresponding request.

REQ-ASIP-6.2.3.2.1-03: The `query:QueryResponse` root element for a successful Verify Response shall contain the five slots as child elements: `SpecificationIdentifier`, `SpecificationIdentifier`, `IssueDateTime`, `Requester` and `Provider`.

- **REQ-ASIP-6.2.3.2.1-03-01**: `SpecificationIdentifier` shall be of type *rim:StringValueType* and shall be the fixed value "`https://uri.etsi.org/19478/v1.1.1`".
- **REQ-ASIP-6.2.3.2.1-03-02**: `ResponseIdentifier` shall be of type *rim:StringValueType* and shall uniquely identify the response message using a UUID value in the URN namespace according to IETF RFC 9562 [18].
- **REQ-ASIP-6.2.3.2.1-03-03**: `IssueDateTime` shall be of type *rim:DateTimeValueType* and shall be set to the date and time the response was issued in the date and time format according to ISO 8601-1 [22].
- **REQ-ASIP-6.2.3.2.1-03-04**: `Requester` shall identify the person making the request.
  - **REQ-ASIP-6.2.3.2.1-03-04-01 [CONDITONAL]**: If the verification requester is a natural person, the `Requester` shall be of type *NaturalPersonValueType*.
  - **REQ-ASIP-6.2.3.2.1-03-04-02 [CONDITONAL]**: If the verification requester is a legal person, the `Requester` shall be of type *LegalPersonValueType*.
- **REQ-ASIP-6.2.3.2.1-03-05**: `Provider` shall identify the person that issues the response.
  - **REQ-ASIP-6.2.3.2.1-03-05-01 [CONDITIONAL]**: If the provider is a natural person, the `Provider` shall be of type *NaturalPersonValueType*.
  - **REQ-ASIP-6.2.3.2.1-03-05-02 [CONDITIONAL]**: If the provider is a legal person, the `Provider` shall be of type *LegalPersonValueType*.

REQ-ASIP-6.2.3.2.1-04: In addition, the `query:QueryResponse` shall contain a `rim:RegistryObjectList`.

- **REQ-ASIP-6.2.3.2.1-04-01**: This `rim:RegistryObjectList` shall contain one `rim:RegistryObject` for each `AttributeVerificationQuery` in the `query:QueryRequest` and shall not contain a `rim:Exception` element.
- **REQ-ASIP-6.2.3.2.1-04-02 [CONDTIONAL]**: In case of an immediate response, the element `query:QueryResponse` shall not contain the `rim:Slot` "`ResponseAvailableDateTime`".

NOTE 1: The "`ResponseAvailableDateTime`" is used in responses indicating deferral (see clause 6.2.3.2.3 below).

REQ-ASIP-6.2.3.2.1-05: Each `rim:RegistryObject` shall be of type *rim:RegistryObjectType* and shall contain the two slots `Person` and `VerificationResponseDetails`

- **REQ-ASIP-6.2.3.2.1-05-01**: `Person` shall be exactly as specified in REQ-ASIP-6.2.3.1-03-01 for the Verify Request.
- **REQ-ASIP-6.2.3.2.1-05-02**: `VerificationResponseDetails` shall provide details of an attribute for which verification is requested.
- **REQ-ASIP-6.2.3.2.1-05-03**: The content of the `rim:SlotValue` element, which corresponds to the `VerificationResponseDetails` element, shall be of type *VerificationResponseDetailsValueType* defined in the XML schema specified in clause C.4, which enables the inclusion of a `VerificationsResponseDetails` element.
- **REQ-ASIP-6.2.3.2.1-05-04**: The `VerificationsResponseDetails` element shall be of type *VerificationResponseDetailsType*, defined in the XML schema specified in clause C.3.

REQ-ASIP-6.2.3.2.1-06: The ASIP may support fuzzy matching, meaning it considers attribute values included in the attribute verification query as matching that differs in transliteration, blank spaces, hyphenation, concatenation, and similar orthographic variations from the corresponding attribute value held by the authentic source.

- **REQ-ASIP-6.2.3.2.1-06-01 [CONDITONAL]**: In the case, the ASIP supports fuzzy matching, it shall return the attribute value stored in the authentic source in the `AttributeVerificationResponse` element specified below.

NOTE 2: The decision whether the requesting provider uses the attribute value provided in the verification query or the matching attribute value held by the authentic source for issuing electronic attestations of attributes, is out of scope of the present specification.

NOTE 3: Article 2 of CIR (EU) 2025/846 [i.9] specifies general requirements for the unequivocal identity matching for natural persons in online cross-border services offered by or on behalf of a public sector body. Article 2 (6) of CIR (EU) 2025/846 [i.9] stipulates that the unequivocal identity matching "shall, to the extent possible, not be affected by differences in transliteration, blank spaces, hyphenation, concatenation, and similar orthographic variations that are required under Union law or national law of the Member State".

REQ-ASIP-6.2.3.2.1-07: The `rim:RegistryObject` for which an attribute verification is requested in an `AttributeVerificationQuery` shall contain a corresponding `AttributeVerificationResponse` element, which shall be of type AttributeVerificationResponseType.

REQ-ASIP-6.2.3.2.1-08: This `AttributeVerificationResponse` element shall contain the following mandatory or optional child elements:

- **REQ-ASIP-6.2.3.2.1-08-01**: The `AttributeVerificationResponse` element shall contain an `AttributeIdentifier`, which shall be of type xs:anyURI and shall contain the identifier of the attribute for which values are to be verified in form of a URI according to IETF RFC 3986 [2].

- **REQ-ASIP-6.2.3.2.1-08-02 [CONDITIONAL]**: If the `Schema` element as specified in clause 6.2.3.1 was present in the corresponding `AttributeVerificationQuery`, then the `AttributeVerificationResponse` element shall contain a `Schema` which shall contain a copy of the one present in the request.

- **REQ-ASIP-6.2.3.2.1-08-03 [CONDITIONAL]**: If the `SchemaMediaType` element as specified in clause 6.2.3.1 was present in the corresponding `AttributeVerificationQuery`, then the `AttributeVerificationResponse` element shall contain a `SchemaMediaType`, which shall contain a copy of the one present in the request.

- **REQ-ASIP-6.2.3.2.1-08-04**: The `AttributeVerificationResponse` element shall include exactly one child element of the choice `TextValue`, `XMLValue` or `AttributeProperties`.

  - **REQ-ASIP-6.2.3.2.1-08-04-01 [CONDITIONAL]**: If the `AttributeVerificationQuery` (see REQ-ASIP-6.2.3.1-08) contained a child element `TextValue`, then the `AttributeVerificationResponse` element shall include a child element `TextValue` which shall contain the text value as stored in the authentic source.

NOTE 4: In case of fuzzy matching in the sense of REQ-ASIP-6.2.3.2.1-06 it is possible that the content is not exactly the same as in the request, but an orthographic variant.

  - **REQ-ASIP-6.2.3.2.1-08-04-02 [CONDITIONAL]**: If the `AttributeVerificationQuery` (see REQ-ASIP-6.2.3.1-08) contained a child element `XMLValue`, then the `AttributeVerificationResponse` element shall include a child element `XMLValue` which shall contain the XML value stored in the authentic source.

NOTE 5: In case of fuzzy matching in the sense of REQ-ASIP-6.2.3.2.1-06 it is possible that the content is not exactly the same as in the request, but an orthographic variant.

  - **REQ-ASIP-6.2.3.2.1-08-04-03 [CONDITIONAL]**: If the `AttributeVerificationQuery` (see REQ-ASIP-6.2.3.1-08) contained a child element `AttributeProperties`, then the `AttributeVerificationResponse` element shall include a child element `AttributeProperties`, which shall contain the attribute properties as stored in the authentic source.

NOTE 6: In case of fuzzy matching in the sense of REQ-ASIP-6.2.3.2.1-06 it is possible that the content is not exactly the same as in the request but an orthographic variant.

EXAMPLE: If a verification request was issued for two attributes X and Y, and if for X the attribute verification query used a `TextValue` for X and an `AttributeProperties` for Y, then the attribute verification response for X shall use a `TextValue` and an `AttributeProperties` for Y.

- **REQ-ASIP-6.2.3.2.1-08-05 [CONDITONAL]**: If and only if the ASIP is different from the authentic source and the output is different from `http://uri.etsi.org/19478/VerificationResult/Unknown`, then the `AttributeVerificationResponse` element shall contain a child element `AuthenticSource`, which shall be of type PersonType as defined in the XML-schema provided in clause C.2 and which shall represent the authentic source.

- **REQ-ASIP-6.2.3.2.1-08-06**: The `AttributeVerificationResponse` element shall contain a child element `VerificationResult`, which specifies the result of the matching of attribute by the provider. It shall have exactly one of the following four values:

  1) `http://uri.etsi.org/19478/VerificationResult/Match` shall indicate that the value held for this user by the authentic source matches the value claimed by the (Q)TSP.

  2) `http://uri.etsi.org/19478/VerificationResult/NoMatch` shall indicate that the value held for this user by the authentic source does not match the value claimed by the (Q)TSP.

  3) `http://uri.etsi.org/19478/VerificationResult/MatchWithVariation` shall indicate that the value held for this user by the authentic source matches the value claimed by the (Q)TSP with an admissible orthographic variation in the sense of REQ-ASIP-6.2.3.2.1-06 above.

  4) `http://uri.etsi.org/19478/VerificationResult/Unknown` shall indicate that no authentic source data was available to determine a match for the attribute for the user.

6.2.3.2.2 Failure Error Response

REQ-ASIP-6.2.3.2.2-01: The failure error response shall be an XML document rooted in `query:QueryResponse` where the status attribute shall be set to `urn:oasis:names:tc:ebxml-regrep:ResponseStatusType:Failure`.

REQ-ASIP-6.2.3.2.2-02: The XML document shall have an attribute named `requestId`, which shall be set to the value of the `id` attribute in the corresponding request.

REQ-ASIP-6.2.3.2.2-03: The failure error response shall have the following mandatory slots as child elements:

1) `SpecificationIdentifier`;

2) `ResponseIdentifier`;

3) `IssueDateTime`;

4) `Requester`; and

5) `Provider`.

REQ-ASIP-6.2.3.2.2-03-01: The `SpecificationIdentifier` element shall be of type *rim:StringValueType* and shall be set to the fixed value "`https://uri.etsi.org/19478/v1.1.1`".

- **REQ-ASIP-6.2.3.2.2-03-02**: The `ResponseIdentifier` element shall be of type *rim:StringValueType* and shall uniquely identify the response message using a UUID in the URN namespace according to IETF RFC 9562 [18].

- **REQ-ASIP-6.2.3.2.2-03-03**: The `IssueDateTime` element shall be of type *rim:DateTimeValueType* and shall be set to the date and time the response was issued in the date and time format according to ISO 8601-1 [22].

- **REQ-ASIP-6.2.3.2.2-03-04**: The `Requester` element shall be of type *PersonType* as defined in the XML-schema provided in clause C.2 and shall identify the (Q)TSP making the request.

- **REQ-ASIP-6.2.3.2.2-03-05**: The `Provider` element shall be of type *PersonType* as defined in the XML-schema provided in clause C.2 and shall identity the ASIP that issues the response.

REQ-ASIP-6.2.3.2.2-04 [CONDITIONAL]: In case the entity creating the exception is different from the ASIP, the error response may contain the slot `ErrorProvider`.

- **REQ-ASIP-6.2.3.2.2-04-01 [CONDITIONAL]**: If the `ErrorProvider` is present and the entity creating the exception is a natural person, the `ErrorProvider` shall be of type *NaturalPersonType*.

- **REQ-ASIP-6.2.3.2.2-04-02 [CONDITIONAL]**: If the `ErrorProvider` is present and the entity creating the exception is a legal person, the `ErrorProvider` shall be of type *LegalPersonType*.

REQ-ASIP-6.2.3.2.2-05: The contents of the `Requester` and `Provider` slots shall match the contents of the corresponding slots in the request.

REQ-ASIP-6.2.3.2.2-06: The message shall contain a `rim:Exception` element instead of a `rim:RegistryObjectList` element.

- **REQ-ASIP-6.2.3.2.2-06-01 [CONDITIONAL]**: In case the error is raised due to a non-match, the type of the `rim:Exception` shall be *rs:ObjectNotFoundExceptionType*.

- **REQ-ASIP-6.2.3.2.2-06-02 [CONDITIONAL]**: In error situations different to a no-match, appropriate exception types defined in ISO 15000-3 [24] shall be used.

REQ-ASIP-6.2.3.2.2-07: The `rim:Exception` shall contain the slot `TimeStamp` which shall be of type *rim:DateTimeValueType* and shall be set to the date and time at which the exception was raised, expressed in the date and time format according to ISO 8601-1 [22].

6.2.3.2.3 Deferred Response

REQ-ASIP-6.2.3.2.3-01 [CONDITIONAL]: If the outcome of processing of the verification request is not immediately available, a `rim:QueryResponse` shall be returned with a value `urn:oasis:names:tc:ebxml-regrep:ResponseStatusType:Unavailable` for the status attribute instead of `urn:oasis:names:tc:ebxml-regrep:ResponseStatusType:Success`.

REQ-ASIP-6.2.3.2.3-02 [CONDITIONAL]: If the outcome of processing of the verification request is not immediately available , the content of the `rim:QueryResponse` element shall contain an empty `rim:RegistryObjectList`.

REQ-ASIP-6.2.3.2.3-03 [CONDITIONAL]: If the outcome of processing of the verification request is not immediately available, the content of the `rim:QueryResponse` element shall not contain a `rim:Exception` element.

REQ-ASIP-6.2.3.2.3-04 [CONDITIONAL]: If the outcome of processing of the verification request is not immediately available, the content of the `rim:QueryResponse` element shall contain:

- the mandatory slots defined in clause 6.2.3.2.1 above; and

- the `rim:Slot ResponseAvailableDateTime`, which shall indicate the expected time of availability in the date and time format according to ISO 8601-1 [22].

6.2.4 I3 (Retrieve)

6.2.4.1 Overview

The **I3 (Retrieve)** protocol is a variant of the **I2 (Verify)** protocol specified in clause 6.2.3 above.

REQ-ASIP-6.2.4.1-01: The specification defined in clause 6.2.3 shall apply, except where specified in the following clauses.

NOTE: Clause 6.2.4.2 specifies the Retrieve Request and clause 6.2.4.3 specifies the Retrieve Response.

6.2.4.2 Retrieve Request

REQ-ASIP-6.2.4.2-01: The verification request shall be an XML document rooted in `query:QueryRequest` and shall have an attribute named `id` with a UUID value in the URN Namespace [18].

- **REQ-ASIP-6.2.4.2-01-01:** The verification request shall have the same six named `rim:Slot` child elements as the Verify Request as specified in specified in REQ-ASIP-6.2.3.1-01 (see clause 6.2.3.1).
- **REQ-ASIP-6.2.4.2-01-02:** The verification request shall include a `query:ResponseOption` element with a `returnType` attribute set to the value "`RegistryObject`".

REQ-ASIP-6.2.4.2-02 [CONDITIONAL]: If the request is issued to obtain a deferred response to a previously issued request (see clause 6.2.3.2.3 below), it shall contain the additional slot `DeferredResponseIdentifier`, which shall be of type *rim:StringValueType* and shall be set to the value of the `ResponseIdentifier` issued in the response with unavailable status to the original request.

These slots apply to all attributes for which retrieval is requested.

REQ-ASIP-6.2.4.2-03: The `query:Query` element shall be of the fixed type "`RetrieveQuery`" and shall have two `rim:Slot` child elements `Person` and `RetrieveQueryDetails`.

- **REQ-ASIP-6.2.4.2-03-01**: The `Person` child element shall be exactly as specified in REQ-ASIP-6.2.3.1-03-01 for the Verify Request.
- **REQ-ASIP-6.2.4.2-03-02**: The `RetrieveQueryDetails` child element shall provide details of the attributes for which retrieval is requested and shall be of type *RetrieveQueryDetailsType* as defined in the XML schema specified in clause C.3.

REQ-ASIP-6.2.4.2-04: The content of the `rim:SlotValue` element, which corresponds to the `RetrieveQueryDetails` element and enables its inclusion, shall be of type *RetrieveQueryDetailsValueType* defined in the XML schema specified in clause C.4.

REQ-ASIP-6.2.4.2-05: For each attribute for which retrieval is requested, a separate `AttributeRetrieveQuery` element shall be included.

- **REQ-ASIP-6.2.4.2-05-01**: All `AttributeRetrieveQuery` elements in a `RetrieveQueryDetails` element shall relate to the same `Person` element.

REQ-ASIP-6.2.4.2-06: An `AttributeRetrieveQuery` shall have the following child elements:

- **REQ-ASIP-6.2.4.2-06-01**: The `AttributeRetrieveQuery` shall contain the child element `AttributeIdentifier` which shall be of type *xs:anyURI* and shall contain the identifier of the attribute which is requested to be retrieved in form of a URI according to IETF RFC 3986 [2].
- **REQ-ASIP-6.2.4.2-06-02**: The `AttributeRetrieveQuery` shall contain the child element `Schema.`
  - **REQ-ASIP-6.2.4.2-06-02-01 [CONDITONAL]**: If present, the `Schema` element shall be of type *xs:anyURI* and shall contain the link to the data model schema of the attribute in the catalogue of attributes according to Article 2 (3) of CIR (EU) 2025/1569 [i.11], which defines the internal structure of the attribute, which is requested to be retrieved.
  - **REQ-ASIP-6.2.4.2-06-02-02 [CONDITONAL]**: If the `Schema` element is not present, the requested attribute shall be returned in all available data model schema distributions provided by the ASIP.
  - **REQ-ASIP-6.2.4.2-06-02-03 [CONDITONAL]**: If the `Schema` is present element, the `AttributeRetrieveQuery` shall contain the child element `SchemaMediaType`, which shall be of type *xs:string* and shall identify the media type of the `Schema` distribution according to IETF RFC 6838 [8].

6.2.4.3 Retrieve Response

6.2.4.3.1 Successful Retrieve Response

REQ-ASIP-6.2.4.3.1-01: The successful retrieve response providing an immediate result shall be an XML document rooted in a `query:QueryResponse` element with `status` set to the value `urn:oasis:names:tc:ebxml-regrep:ResponseStatusType:Success` and shall have an attribute named `requestId` set to the value of the `id` attribute in the corresponding request.

REQ-ASIP-6.2.4.3.1-02: The `query:QueryResponse` root element for a successful Retrieve Response shall contain exactly the same five slots as child elements as the successful Verify Response specified in REQ-ASIP-6.2.3.2.1-03.

REQ-ASIP-6.2.4.3.1-03: In addition, the `query:QueryResponse` shall contain a `rim:RegistryObjectList`.

- **REQ-ASIP-6.2.4.3.1-03-01:** This `rim:RegistryObjectList` shall contain one `rim:RegistryObject` for each `AttributeRetrieveQuery` in the `query:QueryRequest`.
- **REQ-ASIP-6.2.4.3.1-03-02:** This `rim:RegistryObjectList` shall not contain a `rim:Exception` element.
- **REQ-ASIP-6.2.4.3.1-03-03 [CONDTIONAL]**: In the case of an immediate response `rim:Slot` "`ResponseAvailableDateTime`" used in responses indicating deferral (see clause 6.2.4.3.3 below) shall not be used.

REQ-ASIP-6.2.4.3.1-04: Each `rim:RegistryObject` shall be of type *rim:RegistryObjectType* and shall contain the two slots `Person` and `RetrieveResponseDetails`.

- **REQ-ASIP-6.2.4.3.1-04-01**: The `Person` slot shall be exactly as specified in REQ-ASIP-6.2.3.1-03-01 for the Verify Request.
- **REQ-ASIP-6.2.4.3.1-04-02**: The `RetrieveResponseDetails` slot shall provide details of an attribute for which retrieval is requested.
  - **REQ-ASIP-6.2.4.3.1-04-02-01**: The `RetrieveResponseDetails` element shall be of type *RetrieveResponseDetailsType*, defined in the XML schema specified in clause C.3.
  - **REQ-ASIP-6.2.4.3.1-04-02-02**: The content of the `rim:SlotValue` element, which corresponds to the `RetrieveResponseDetails` element and enables its inclusion, shall be of type *RetrieveResponseDetailsValueType* defined in the XML schema specified in clause C.4.
  - **REQ-ASIP-6.2.4.3.1-04-02-03**: For the attribute for which retrieval is requested in an `AttributeRetrieveQuery` the `rim:RegistryObject` shall contain a corresponding `AttributeRetrieveResponse` element.
- **REQ-ASIP-6.2.4.3.1-04-03**: An `AttributeRetrieveResponse` element shall have the following child elements:
  - **REQ-ASIP-6.2.4.3.1-04-03-01**: The `AttributeRetrieveResponse` shall include an `AttributeIdentifier` child element, which shall be of type *xs:anyURI* and which shall contain the identifier of the attribute which is requested to be retrieved in form of a URI according to IETF RFC 3986 [2].
  - **REQ-ASIP-6.2.4.3.1-04-03-02 [CONDITONAL]**: If the `Schema` child element was present in the corresponding `AttributeRetrieveQuery` in the Retrieve Request specified in clause 6.2.4.2, then the `AttributeRetrieveResponse` shall contain a copy of this `Schema` child element.
  - **REQ-ASIP-6.2.4.3.1-04-03-03 [CONDITONAL]**: If the `SchemaMediaType` child element was present in the corresponding `AttributeRetrieveQuery` in the Retrieve Request specified in clause 6.2.4.2, then the `AttributeRetrieveResponse` shall contain a copy of this `SchemaMediaType` child element.

- **REQ-ASIP-6.2.4.3.1-04-03-04**: The `AttributeRetrieveResponse` element shall include exactly one of a choice of the two child elements `TextValue` and `XMLValue`.
  - **REQ-ASIP-6.2.4.3.1-04-03-04-01 [CHOICE, CONDITONAL]**: If present, the `TextValue` element shall contain the full data content of the attribute encoded as a string and this value shall be used with an attribute with JSON content.
    - **REQ-ASIP-6.2.4.3.1-04-03-04-01-01 [CHOICE, CONDITONAL]**: If `TextValue` is present and `SchemaMediaType` is present, then `SchemaMediaType` shall be set to `application/json-schema` or `application/ld+json`.
    - **REQ-ASIP-6.2.4.3.1-04-03-04-01-02 [CHOICE, CONDITONAL]**: If present, the `TextValue` element may have one or more of the two optional attributes `encoding` and `xml:lang`, which are specified above (see REQ-ASIP-6.2.3.1-08-01).
  - **REQ-ASIP-6.2.4.3.1-04-03-04-02 [CHOICE, CONDITONAL]**: If present, the `XMLValue` element shall contain the full data content of the attribute encoded as an XML element and this value shall be used with an attribute with XML content.
  - **REQ-ASIP-6.2.4.3.1-04-03-04-03 [CHOICE, CONDITONAL]**: If `XMLValue` is present and `SchemaMediaType` is present, then `SchemaMediaType` shall be set to `application/xml`.
- **REQ-ASIP-6.2.4.3.1-04-03-05**: The `AttributeRetrieveResponse` shall include a `RetrieveResult` element, which shall specify the result of the retrieval of the attribute content and which shall have one of the two values:
  1) `http://uri.etsi.org/19478/RetrieveResultTypes/Success` shall indicate the attribute value held for this user by the authentic source was successfully retrieved.
  2) `http://uri.etsi.org/19478/RetrieveResultTypes/Failure` shall indicate that the attribute value held for this user by the authentic source was not retrieved.

6.2.4.3.2 Failure Error Response

REQ-ASIP-6.2.4.3.2-01: The failure error response shall follow the specification for verify failure error response, as specified above in clause 6.2.3.2.2.

6.2.4.3.3 Deferred Response

REQ-ASIP-6.2.4.3.3-01: The deferred error response shall follow the specification for verify deferred response, as specified above in clause 6.2.3.2.3.

Annex A (normative): OpenAPI Specification for Discover Interface

REQ-DIP-A-01: The DIP shall provide the Discover Interface as specified in clause 5 based on the OpenAPI 3.0 specification in the file `19478-discover-interface-openapi.json` at https://forge.etsi.org/rep/esi/x19_478_authentic_source_interface/raw/v1.1.1/19478-discover-interface-openapi.json.

This OpenAPI 3.0 specification file refers to and integrates the following two JSON-schema files:

- The file `19478-attribute-schema.json` at https://forge.etsi.org/rep/esi/x19_478_authentic_source_interface/raw/v1.1.1/19478-attribute-schema.json contains the JSON-schema of the `search` response as specified in clause 5.2.3.
- The file `19478-dataservice-schema.json` at https://forge.etsi.org/rep/esi/x19_478_authentic_source_interface/raw/v1.1.1/19478-dataservice-schema.json contains the JSON-schema of the `retrieve` response as specified in clause 5.3.3.

Annex B (normative): OpenAPI Specification for Authentic Source Interface

REQ-ASIP-B-01: The ASIP shall provide the HTTP / OAuth based Authentic Source Interface as specified in clause 6.1 based on the OpenAPI 3.0 specification in the file `19478-authentic-source-interface-openapi.json` at https://forge.etsi.org/rep/esi/x19_478_authentic_source_interface/raw/v1.1.1/19478-authentic-source-interface-openapi.json.

This OpenAPI 3.0 specification file refers to and integrates the following JSON-schema file:

- The file `19478-dataservice-schema.json` at https://forge.etsi.org/rep/esi/x19_478_authentic_source_interface/raw/v1.1.1/19478-dataservice-schema.json contains the JSON-schema of the `retrieve` response as specified in clause 5.3.3, which in particular contains the specification of the `provider` object.

Annex C (normative): XML-Schemata for PID and RIM binding

C.1 General

REQ-ASIP-C-01: The ASIP shall provide the ISO 15000-based Authentic Source Interface specified in clause 6.2 based on the XML-schemata provided in the present Annex C.

C.2 Person Identification Data (PID) schema

The file `19478-pid-schema.xsd` at https://forge.etsi.org/rep/esi/x19_478_authentic_source_interface/raw/v1.1.1/19478-pid-schema.xsd contains the XML-schema used for Person Identification Data (PID) of the data subject used within the ISO 15000-based Authentic Source Interface as specified in clause 6.2.

C.3 Interface Details schema

The file `19478-interfaces-schema.xsd` at https://forge.etsi.org/rep/esi/x19_478_authentic_source_interface/raw/v1.1.1/19478-interfaces-schema.xsd contains the XML-schema used for the `VerificationQueryDetails`, `VerificationResponseDetails`, `RetrieveQueryDetails` and `RetrieveResponseDetails` elements used within the ISO 15000-based Authentic Source Interface as specified in clause 6.2.

C.4 Interface Details RIM Binding schema

The file `19478-interfaces-rim-binding-schema.xsd` at https://forge.etsi.org/rep/esi/x19_478_authentic_source_interface/raw/v1.1.1/19478-interfaces-rim-binding-schema.xsd contains the XML-schema elements defined to use the structured defined in clauses C.2 and C.3 schemas as values for `rim:Slot` elements used within the ISO 15000-based Authentic Source Interface as specified in clause 6.2.

Annex D (informative): XML-Examples

The file `19478-verify-request.xml` at https://forge.etsi.org/rep/esi/x19_478_authentic_source_interface/raw/v1.1.1/19478-verify-request.xml contains an example of a Verify Request according to clause 6.2.3.1.

The file `19478-verify-response-success.xml` at https://forge.etsi.org/rep/esi/x19_478_authentic_source_interface/raw/v1.1.1/19478-verify-response-success.xml contains an example of a successful Verify Response according to clause 6.2.3.2.1.

The file `19478-verify-response-failure.xml` at https://forge.etsi.org/rep/esi/x19_478_authentic_source_interface/raw/v1.1.1/19478-verify-response-failure.xml contains an example of a not successful Verify Response according to clause 6.2.3.2.2.

History

Version	Date	Status
V1.1.1	January 2026	Publication