ETSI TS 103 486 V1.1.1 (2024-03)

CYBER;
Identity Management and Discovery for IoT

Reference
DTS/CYBER-0014

Keywords
authentication, authorization, confidentiality,
identification, trust services

ETSI

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

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

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

Contents

Intellectual Property Rights

Essential patents

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

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™** and **LTE™** 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 Cyber Security (CYBER).

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.

Introduction

Identity, identity management, and identifiers are intended to distinguish between instances of a thing. In human discourse the name is often used as the primary distinguisher but as two people can have the same name other characteristics are then used to distinguish between them. In the ICT world for a long time it was considered sufficient to have a single unique name associated to an endpoint and this is best seen in the telephone number. However, the telephone number does not identify either the nature of the device or the user of the device, but best practice has for many years meant that users have chosen to associate themselves to their telephone number.

The present document defines mechanisms for interoperable discovery protocols over sets of role, identity and attribute-based taxonomies, and with which devices can determine what identity information to expose and what to request. This allows decisions to expose Personal Identifying Information (PII) to be made with the best available knowledge of the recipient devices, and can also underlie any techniques such as digital rights management, that try to preserve privacy beyond the immediate recipient.

1 Scope

The present document describes a model of secure and attestable identity management applicable to IoT devices. The present document defines a data structure, described as Authority-Attribute Trees, for managing identifiers and properties of a device (as attributes), where identifiers and properties are exposed in a number of use cases including discovery, attachment and communication.

The present document outlines the requirements for cryptographic methods to establish trust in Authority-Attribute Trees.

The present document does not define or make recommendations about policies for users or devices with regards to revealing identity information, or accepting or rejecting discovery based on this information.

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 at https://docbox.etsi.org/Reference.

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.

Not applicable.

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 are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] ETSI TR 187 010: "Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN); NGN Security; Report on issues related to security in identity management and their resolution in the NGN".

[i.2] ETSI TS 102 165-2: "Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN); Methods and protocols; Part 2: Protocol Framework Definition; Security Counter Measures".

[i.3] ETSI TS 103 264: "SmartM2M; Smart Applications; Reference Ontology and oneM2M Mapping".

[i.4] 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).

[i.5] ETSI TR 103 305-5: "Cyber Security (CYBER); Critical Security Controls for Effective Cyber Defence; Part 5: Privacy and personal data protection enhancement".

[i.6] ETSI TR 103 370: "Practical introductory guide to Technical Standards for Privacy".

[i.7] Recommendation ITU-T X.509: "Information technology - Open Systems Interconnection - The Directory: Public-key and attribute certificate frameworks".

[i.8] Council Regulation (EEC) No 2658/87 of 23 July 1987 on the tariff and statistical nomenclature and on the Common Customs Tariff.

3 Definition of terms, symbols and abbreviations

3.1 Terms

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

attribute provider: 3rd party that attests to the ownership of any attribute of the principal (see attribute authority as defined in requirement 4)

Authority-Attribute Tree (AAT): tree like data structure that encodes identity information

AAT provider: owner of the AAT

AAT requestor: entity that wishes to receive identity/identifier information from the Principal's AAT

identity provider: entity giving authority to the identifier(s) of the principal

NOTE: This also refers to the 3rd party that attests to the root/canonical identifier of the principal (see the Identity authority defined in requirement 5 of clause 4).

principal: entity being identified

NOTE: A principal is also an AAT provider.

private discovery: discovery of devices or resources without public revealing of identity information

relying party: organization providing a service to the Principal

semantic discovery: discovery of devices or resources using plain language terms

3.2 Symbols

Void.

3.3 Abbreviations

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

AAT	Authority-Attribute Tree
DNS	Domain Name System
EEC	European Economic Community
EU	European Union
GDPR	General Data Protection Regulation
ICT	Information and Communications Technology
IdM	Identity Management
IMEI	International Mobile Equipment Identity
IMSI	International Mobile Subscriber Identity
IoT	Internet of Things
M2M	Machine to Machine
NGN	Next Generation Network
NoO	Notice of Obligation
OID	Object IDentifier

OoT	Obligation of Trust
PII	Personal Identifying Information
PKI	Public Key Infrastructure
RDF	Resource Description Framework
SAO	Signed Acceptance of Obligations
SAREF	Smart Appliances REFerence ontology
SIM	Subscriber Identity Module
SPARQL	SPARQL Protocol and RDF Query Language

NOTE: SPARQL is recognized as a recursive acronym.

TMSI	Temporary Mobile Subscriber Identity
vSIM	virtual SIM

4 Identity Management problem

The requirements outlined in Table 1 apply for the purposes of the present document as core requirements for identity management.

Table 1: Core identity management requirements

Requirement id	Text
1	A <<device>> shall be distinguished by an <<identifier>>
2	A <<device>> shall be discoverable using a <<discovery mechanism>>
3	A <<device>> shall belong to a <<device class>>
4	A <<device class>> shall be managed and attested to by a <<device class authority>>
5	An <<identifier>> shall be assigned, attested to, and managed, by an <<Identity authority>>

The classical model of Identity Management (IdM), as identified in ETSI TR 187 010 [i.1] and replicated in part below (see also Figure 1), using the simplified model of 3 actors (Principal, Relying Party, Identity Provider), only addresses requirement 1 and requirement 5 of the list presented above. The role of discovery is addressed in requirement 2 and can be extended for semantic discovery in requirements 3 and 4. Thus, combining the classical identity management model and the discovery model addresses all 5 requirements.

NOTE 1: The terms device, thing, and entity are almost but not entirely synonymous. Many ontologies of identity have "thing" at the root or centre. In SAREF this is extended to "device", with the term entity being an alternative to "thing".

The actors of the classical model of Identity Management from ETSI TR 187 010 [i.1] are as follows:

- Principal:
  - Often synonymous with the end-user or an electronic agent of the end-user.
  - The entity being identified.

NOTE 2: The Principal can be a person or a device, or a piece of software or some other thing. The general term used here is entity unless it can be explicitly identified as a person or device or some other thing.

- Identity Provider:
  - The organization generally required to authenticate the Principal and to provide an assertion of this authentication to the Relying Party.
  - The entity giving authority to the identifier(s) of the Principal.
- Relying Party:
  - An organization or entity providing a service to, or receiving a service from, the Principal.
  - The Relying Party can rely on an assertion of the identity of the Principal provided by the Identity Provider.

NOTE 3: This is the normal practice where identity is asserted within a public key architecture and the Principal offers his identifier within a public key certificate that has been verified by the Identity Provider.

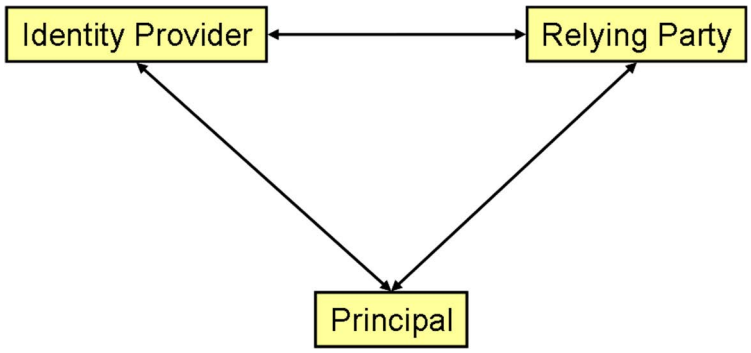

Figure 1: The three primary roles in the classical Identity Management model

In many cases the Principal will be represented by an alias. In some cases the Identity Provider will be aware of and manage the alias (see example 1 below), in other cases the alias shall be applied by the Principal or by the Relying Party (see example 2 below). If the alias is not shared between the Principal and the Relying Party the provisions in the present document shall not apply.

EXAMPLE 1: In cellular radio the Temporary Mobile Subscriber Identity (TMSI) is a network managed alias assigned to mask the IMSI when used in signalling and is assigned by an agent of the Identity Provider.

EXAMPLE 2: The identification of a specific instance of a printer on a network can contain or be associated to multiple functions, each with a different identifier, each acting as an alias. Thus the printer can be known to its manufacturer by its serial number ("AB12345YZ"), to the network management entity by its management tag ("PRT123"), to the end user by its friendly name ("printer in Dave's office") and so forth. Thus, whilst the serial number can be viewed as the canonical identifier assigned by the Identity Provider and known to all parties, the management tag and the friendly name are locally managed aliases and are not shared with the Identity Provider.

The conventional IdM model has been reconsidered in a number of environments and these address aspects of discovery and complexity that are also addressed in the present document. In Object Oriented programming the *entity-attribute-value* structure extends the IdM model towards association of attributes to the Principal, and the Resource Description Framework (RDF) used in many ontologies adopts *subject-predicate-object* expressions, which also extend the richness of expression of the principal from the conventional IdM model.

NOTE 4: There are a number of problems in terminology that have been addressed in part in ETSI TR 187 010 [i.1], in particular in stating that an identity is a composite of many identifiers. In the context of IoT and with respect to discovery the necessary data to uniquely distinguish one entity from another does not necessarily map to the identity, nor to any chosen representation or mask of identity in the form of a persona.

5 Identity management model - Authority-Attribute Tree structure

5.1 General overview

The present document defines Authority-Attribute Trees as data structures that encode identity information. The structure is "tree like" containing a persistent root node and an arbitrary number of attribute leaf nodes. The root node of the tree shall contain a canonical identity of the entity attested to by a recognized authority such as its manufacturer. Each leaf node shall contain an attribute of known type and a corresponding attribute value attested to by a recognized authority. An identity offered by one party to another, or discovered by a party as a representation of another party, is a directed graph linking leaves in the tree back to the root node.

NOTE 1: Some attributes may be linked and this should be considered in how each attribute is assigned within a leaf structure.

EXAMPLE 1: A single SIM smartphone device uses its IMEI as its canonical identity and this can be attested to by the manufacturer (in current smartphones there is no cryptographic assertion or proof of authenticity of the IMEI), whilst the smartphone user uses the IMSI as the canonical identity attested to by the provider by the IMSI-K authentication protocols.

EXAMPLE 2: In an industrial context, say a factory, the root node uses the device's audit-code as its canonical identifier attested to by the owner (the factory).

EXAMPLE 3: In dual SIM smartphones there is an independent IMEI for each SIM-slot and independence of each IMSI-K relationship.

EXAMPLE 4: When using a virtual SIM (vSIM) there is a single machine IMEI for each instance of the vSIM.

A root node shall consist of the authority name and the attestation to the canonical identifier of the tree owner. The root node may contain metadata that the device associates to the authority according to its policies. The metadata of the root node should contain security credentials that attest to the identity of the authority.

The leaf node holding an attribute shall be defined as a triplet {type; value; [SecurityCredential]} where type is an attribute for the device class (see requirement 3 from clause 4), and SecurityCredential is the attestation to the attribute value by the device class authority (see requirement 4 from clause 4). The data set of each leaf should be independent of all other leaves, thus if a party shares knowledge of the content of any leaf node it should be difficult for any adversary to infer the existence of any other leaf node owned by that party.

Whilst SecurityCredential is shown as optional the primary purpose of the present document is to describe the means of securely attesting to an attribute within a node of the tree thus the remainder of the present document addresses non-null SecurityCredentials.

An Authority-Attribute tree is a representation of the connectivity of a "chain" of leaves to the root. Such trees can be pre-created (e.g. at device configuration) or be established on the fly (see figures 2, 3 and 4).

EXAMPLE 5: A conventional PKI structure of attribute certificates linked to an identity certificate can be viewed as a static implementation of the Authority-Attribute tree. Thus, the root node contains the identity of the entity asserted to via a public-key certificate; e.g. {type = AuthenticationIdentity, Value = SubjectName, SecurityCredential = PublicKeyCertificate}. The leaf node contains the attribute asserted to via an attribute certificate that links back to the public-key certificate; e.g. {type = AttributeType, Value = AttributeValue, SecurityCredential = AttributeCertificate}.

EXAMPLE 6: A single symmetric key pairing such as the {type = AuthenticationIdentity, Value = IMSI, SecurityCredential = K} acts as the root node and the telephone number {type = PublicTelephoneNumber; Value = +447700900000, SecurityCredential = Null} as a leaf node.

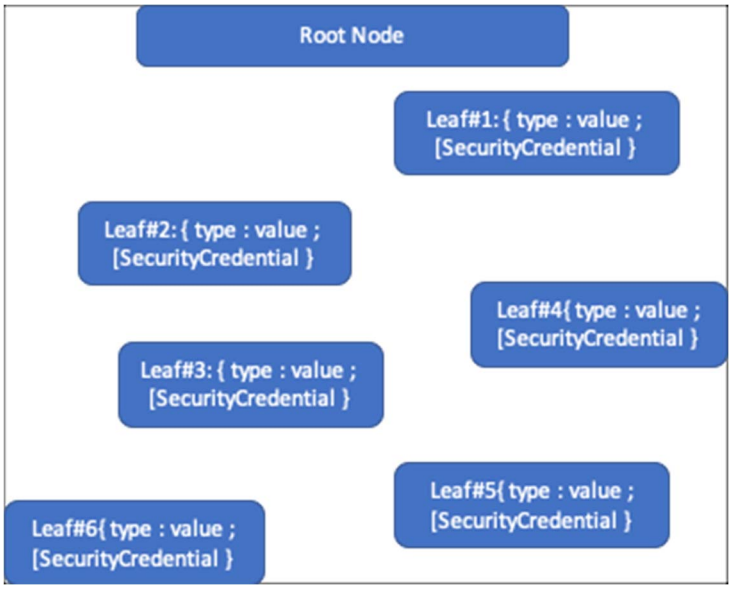

Figure 2: Unconnected set of attribute nodes

Figure 2 illustrates a set of attributes for a party that are attested to by a recognized authority, but which are not organized into an Authority-Attribute tree. Figure 3 illustrates an Authority-Attribute tree that links a subset of the party's attributes so that they can be shared with a potential correspondent.

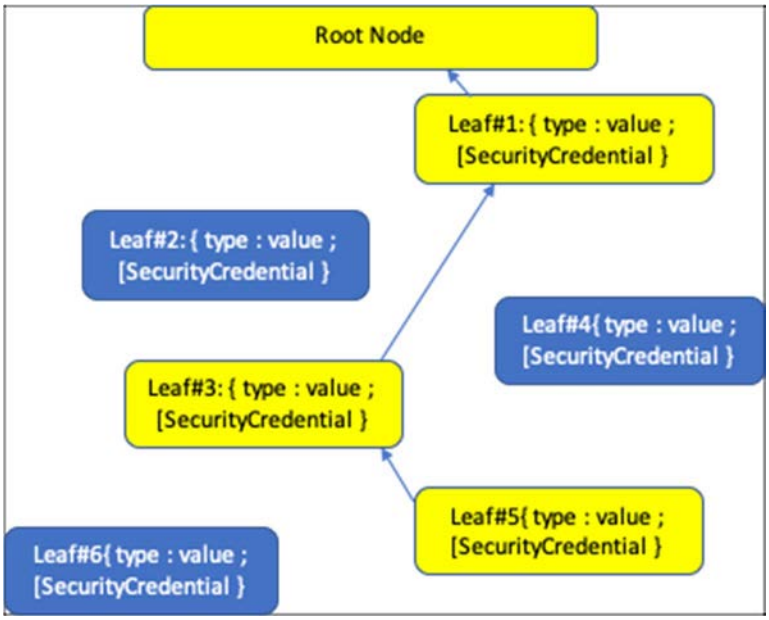

Figure 3: One example of an Authority-Attribute tree

Figure 4 extends the example of building Authority-Attribute trees showing a single leaf shared between 2 trees (leaf 3 is shared).

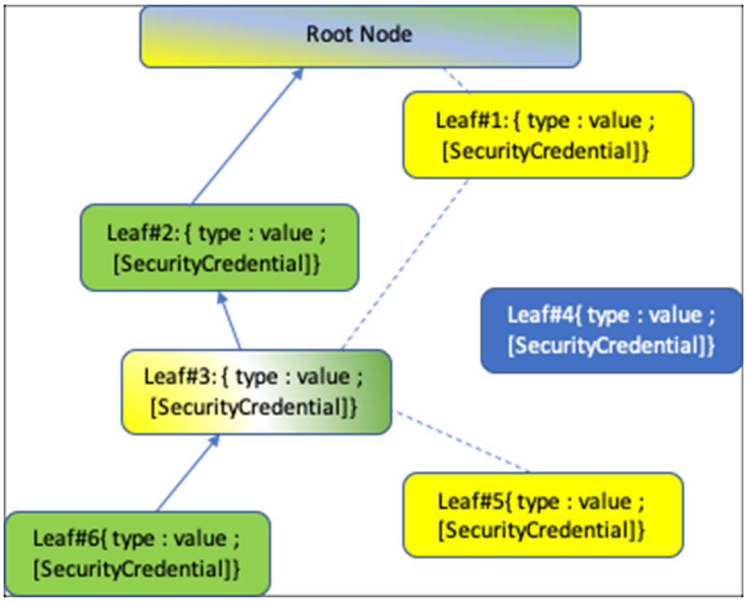

Figure 4: Example of an Authority-Attribute tree where a leaf is shared

Each entity in the identity system shall maintain at least one Authority-Attribute tree, where the root of each tree is an authority the device trusts. This trusted authority may attest to the attributes in the tree. Where an entity has more than one Authority-Attribute tree each tree shall have a unique root authority.

EXAMPLE 7: A device associates an authentication token (such as a public key) to the authority, and the root node stores this metadata as:

- Root: authority; {metadata}

EXAMPLE 8: A device recognizes all of the following as authorities: the device manufacturer, the enterprise owner, and one or more authorities who own or administer installed capabilities.

NOTE 2: The model number of a device can naturally be a node in more than one tree - that is, the value fields of the nodes for model number in two trees can be identical. In this case the device policy, out of scope of the present document, can record the fact for either or both nodes that there is a corresponding node under another authority. This can in some circumstances result in, for example, a decision by the requestor after one round of a discovery protocol with a potential partner, to attempt to discover additional information about the potential partner through further discovery, if it has been determined that the potential partner "shares" one of the nodes in question with the requestor.

The tree construct gives greater flexibility than can be attained by using simple fixed hierarchies alone.

EXAMPLE 9: A tablet is one kind of display-capable device, and in some cases, the attribute of "display" will be what a peer device discovers.

In general, each device has a policy (the contents of the policy are out of scope of the present document) which, given knowledge of one or more partial Authority-Attribute trees of another device influences or determines the extent to which each device trusts the other.

NOTE 3: The capabilities and requirements of Obligation of Trust are attribute trees and are examined in more detail in Annex B.

In a device with *n* leaf nodes, where *n* is greater than or equal to 1, and where an Authority-Attribute tree contains at least 1 leaf node and the root node, the number of possible Authority-Attribute trees is equal to sum of the number of permutations of each siz:

$$P = \sum_{r=1}^{n-1} \frac{n!}{(n-r)!}$$

An entity can be identified by the content of each permutation. The intent of the attribute-authority tree is that any one permutation should not uniquely identify any entity but, rather, any permutation is sufficient to contextually distinguish one party from another.

EXAMPLE 10: In a room of people it may be sufficient to identify someone as the woman wearing glasses standing at the window, in which case only a small set of attributes are sufficient to contextually identify that person (in this case, their sex, their location, and a distinguishing feature). In an IoT environment it may be possible to distinguish a set of window opening sensors from each other by their location, such as by identifying the sensor on the window in the library closest to the door.

Each leaf node as described above is represented as a triplet: {type ; value ; [Security-Credential]}. The security credential in a leaf node shall attest to the type and value of the node. In addition to the persistent attributes in a leaf node, any device or entity has context, representing its mutable attributes (e.g. location, current time). Mutable attributes are self-asserted from the same authority representing the root node as an enabled reporting capability of the device.

Identity in the context of Authority-Attribute trees shall be measured across each permutation of leaves and the root node, and the integrity of that identity measured across the specific permutation.

NOTE 4: A permutation in the IoT context can be likened to a persona in human discourse, thus in a similar way that humans do not give their entire life history to every other person every time they meet, but rather give just enough identity information to enable the relationship, so in IoT using AATs allows release of only sufficient identity information to enable the relationship.

The security claims of the Authority-Attribute trees shall be assessed by verification of the integrity of the tree, with the specific mechanism being a refinement of a Merkle tree that utilizes hashes and digital signatures to provide proofs of integrity and authenticity of the data in the tree.

The structure of each leaf node, i.e. the {type ; value ; [Security-Credential]} triplet, contains a security credential which is a signed attestation of the type and value. This concept, a generalization of a hash chain, is in Figure 5. Integrity proof is commonly created by generating a cryptographic hash of the data (action of the trusted Principal, or indirectly through the identity manager) and making that hash available to the Relying Party. See ETSI TS 102 165-2 [i.2] for a more detailed definition of technical means to achieve proof of integrity.

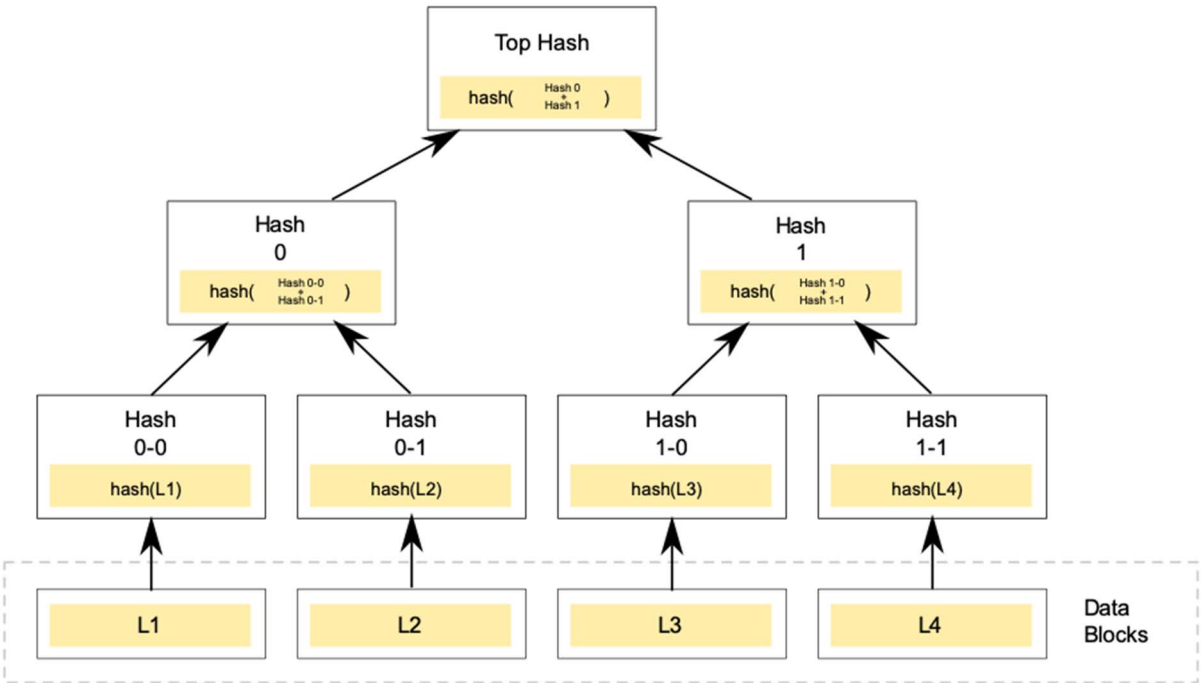

Figure 5: Illustration of Merkle tree (cryptographic integrity tree) from https://en.wikipedia.org/wiki/Merkle_tree#/media/File:Hash_Tree.svg

EXAMPLE 11: A Relying Party wants to find a device with a 'camera' attribute. The Principal publishes the root hash of its Authority-Attribute tree with a permutation that includes the leaf containing the 'camera' attribute. The Relying Party can verify if the 'camera' attribute is a known leaf node in the Principal's Authority-Attribute tree.

5.2 Contextual and persistent attributes

It is recognized that many attributes of an instance of an object are ephemeral, temporary, and contextual. In addition the value assigned to an attribute may be referential (i.e. depends on the value of another attribute). Such non-persistent attribute values do not have persistent attestation but shall be included in the calculation of the permutation.

6 Discovery protocols and Identity Management

6.1 Terminology

The terminology of conventional identify management applies with the additional scope for having multiple authorities for attribute assertion:

- Principal, or AAT provider - owner of the AAT
- Identity Provider - a 3rd party that attests to the root/canonical identifier of the principal (see the Identity authority defined in requirement 5)
- Attribute provider - a 3rd party that attests to the ownership of any attribute of the principal (see attribute authority as defined in requirement 4)
- Relying Party, or AAT requestor - the entity that wishes to receive identity/identifier information from the Principal's AAT

The concern for terminology is that all entities will be a principal at some time, and all entities will also be the relying party at some time (i.e. will be both principal and relying party). In the examples that follow and which inform the discovery protocol messages the use of the terms AAT provider, and AAT requestor are used.

6.2 Discovery use cases

6.2.1 Remote file discovery and access under single authority

In this use case the requestor is unsure of exactly where a particular file is stored on a private cloud, but all assets on the cloud are access restricted, thus the requestor sends a discovery poll request with the triplet to be matched (where the triplet indicates the type of asset the requestor is searching for and the value of the asset (the file name in this case). For the purposes of this example the assumption is made that the requestor has no a-priori knowledge of the AAT provider, but does have confidence that somewhere reachable is an AAT provider that has the requested file. It is also assumed that the requestor is not bypassing the discovery process by use of an intermediate search engine (it is recognized that a search engine could perform this form of simple discovery):

```
{type = File; value = FileName; Null}
```

The requestor can choose to restrict the scope of the polling request (see below). Depending on the value requested in the request the likelihood of multiple matching responses can be significant.

When any instance of an AAT provider receives the request it may choose to say "yes, I have a copy of that file" but this can be insufficient. If the file name is something generic, e.g. "readme.doc", then the likelihood of collision (i.e. a file of that name existing on multiple servers) is high, in which case it can be useful to narrow down the search. What data should the AAT provider with an attribute of file storage entity reveal to the AAT requestor to ensure that the AAT requestor is served with the appropriate asset? An alternative view is, what data is necessary for the AAT requestor to discover that the file storage entity holds the particular file that the AAT requestor wishes to access? There is a concern that any adversary was also a recipient of the initiating messages from the AAT requestor then the same questions can be asked of the adversary: What can an adversary send in response to get revealing data from the AAT requestor, and how can the AAT requestor defend themselves from an adversary?

The AAT requestor has thus far revealed the following:

Who the AAT requestor is;	what AAT attribute is being looked for
{pseudonym associated to the canonical identity; Proof-of-identity};	{type = File; value = FileName; Null}

6.2.2 Home discovery

Home discovery is a refinement of the consumer IoT use-case: discovery and trust establishment between devices from different manufacturers. Current solutions to this use access to a secure home access point as a proxy to anchor trust in the home authority. Alternatives rely on the manufacturer authority to allow trust establishment, through the user account - this can be relied upon when the devices are of the same manufacturer, but is less likely to be interoperable between devices of different manufacturer.

6.3 Discovery and information hiding

The present document describes protocols for nuanced and flexible private discovery of controlled portions of identity information codified in Authority-Attribute trees, as defined in clause 5; in addition, the present document describes protocols for private service advertisement, offer and acceptance. Before they can fully negotiate the type, extent and parameters of their obligations to each other, with optional non-repudiation evidence generated as part of the protocol, communicating parties will complete a discovery phase.

It is not necessary, nor desirable, to make public the content of every leaf of an AAT to a potential correspondent. Not all elements of a thing are offered to a correspondent, rather the thing shall only release information (a specific permutation of the AAT) necessary to establish an identified association.

6.4 Discovery models

6.4.1 Discovery by polling (pull model)

In this case a device or service (the AAT requestor) wants to connect to another device or service (the AAT provider). The AAT requestor polls all devices to determine if they are of the right type, i.e. that attributes of match those required. The AAT requestor shall include the public key and request all responders to sign and encrypt responses.

If the AAT requestor has no prior knowledge of the existence of a matching entity (AAT provider) the initial polling will be seen by all possible instances of a matching entity, and all instances of an adversary. Any instance of an AAT provider with the attribute requested by the AAT requestor is able to respond with the proof of ownership of the attribute and its value (i.e. the attested to leaf node content).

It is not considered mandatory to respond to every request. Rather responses should only be sent if the potential AAT provider can satisfactorily service any future use of the attribute or capability (i.e. the AAT provider has sufficient bandwidth to accommodate the service). If a response is sent by an entity then that entity becomes an AAT provider for the duration of the transaction.

The AAT requestor should only release data regarding its request to limit the likelihood of an adversary to use such data (knowledge of the relation of the AAT requestor to the attributes identified by the AAT requestor in the request) to launch an attack impacting the AAT requestor. The AAT provider shall respond to the AAT requestor's request with the permutation of the leaf-node data and the root node signed *(and* by the key given by the requestor, countersigned by the private key of the target).

The AAT requestor and the AAT provider shall share a common root authority for the keys used in the signature process.

6.4.2 Discovery by advertisement (push model)

In this case an AAT provider periodically publishes its capability for any AAT requestor to find. For restricted channel advertisement the AAT requestor shall subscribe to the AAT provider's subscription channel. For unrestricted channel advertisement the AAT provider is not able to restrict visibility to named entities but may be able to apply some restriction of scope as described in clause 6.4.3.

As with the polling model there can be leakage of data to the adversary.

6.4.3 Restriction of scope of discovery

The nature of the network that underpins the IoT assumes global connectivity. In theory this would suggest that an advertisement, or polling request, can propagate to every node of the connected world. All signalling of capability, either by advertisement or by polling, should restrict the scope of responses. Forms of restriction that can be applied include:

- Geo-fencing;

EXAMPLE 1: The AAT provider or the AAT requestor restricts the scope of discovery, i.e. the limit on advertisement or attribute queries, to specific geographic locations.

- Restriction to a specific sub-net or IP address range;

EXAMPLE 2: The AAT provider or the AAT requestor restrict the scope of discovery, i.e. the limit on advertisement or attribute queries, to specific IP addresses or address ranges.

- Hop restriction (on relaying network topologies);
- Timeout for response to messaging;
- Pre-configuration using DNS or equivalent as a discovery proxy.

6.5 Discovery protocol initialization

For further study.

6.6 Discovery protocol messages

With respect to the AAT structure the following message types are defined for attribute discovery in an AAT environment.

Table 2: AAT message types in discovery scenario

Message	Topology	Acknowledged	Content
AAT_ServiceAdvertisement	Broadcast, Multicast	No	
AAT_ServiceAdvertisement_Bid	Unicast	Yes	
AAT_ServiceAdvertisement_BidAccept	Unicast	Yes	
AAT_ServiceRequest	Broadcast, Multicast		
AAT_ServiceRequest_Offer	Unicast		
AAT_ServiceRequest_OfferAccept	Unicast		

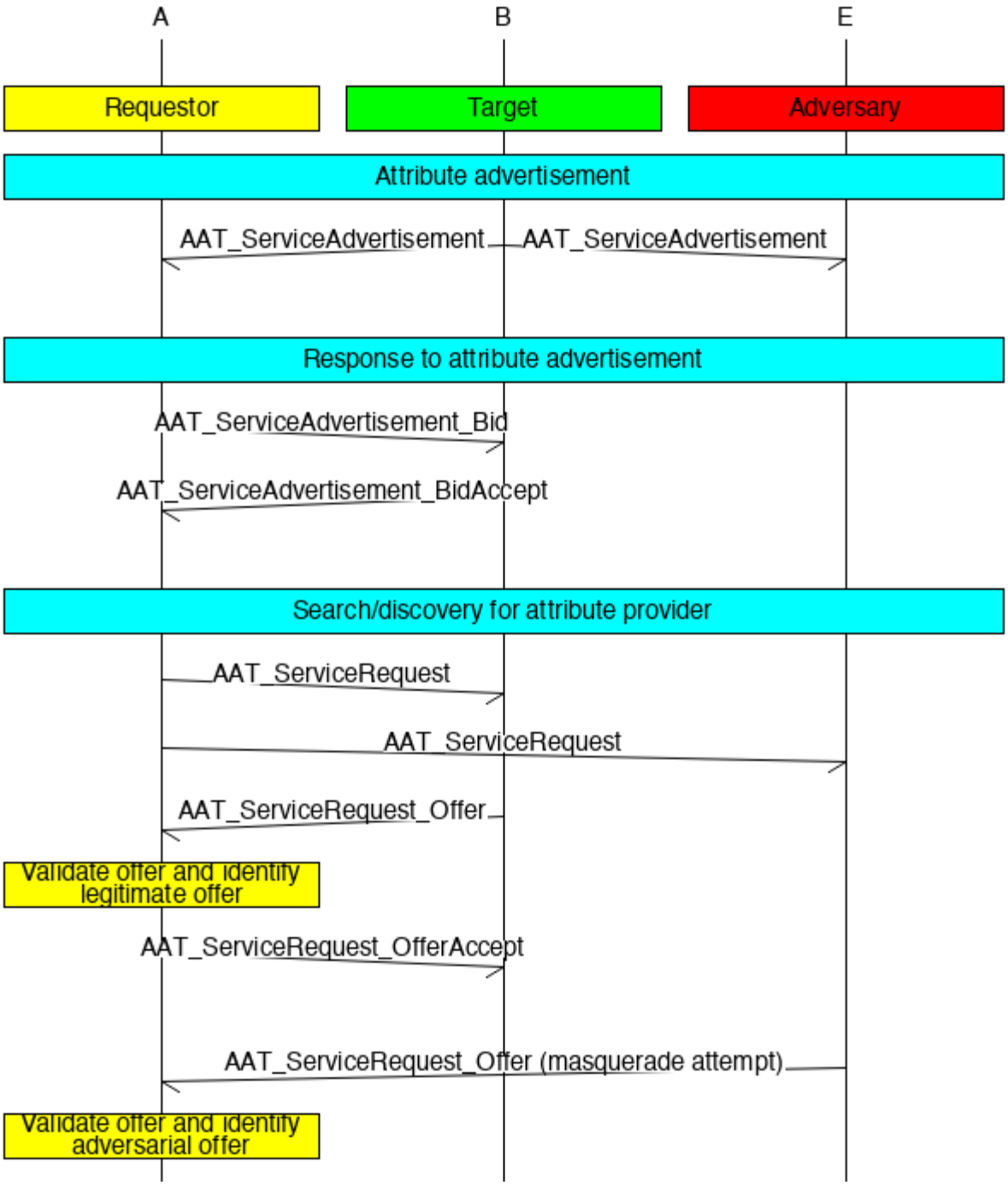

Figure 6: Message sequences for entity discovery

6.7 Application to SAREF

A suitable ontology for IoT/M2M can be found in the Smart Appliances REFerence (SAREF) Ontology in ETSI TS 103 264 [i.3]. SAREF defines a device as "a tangible object designed to accomplish a particular task in households, common public buildings or offices" although SAREF is extensible and each extension may simplify the definition to "a tangible object designed to accomplish a particular task in the defined environment". Each item in SAREF (as in the list below) can be mapped as a leaf attribute in an AAT.

The SAREF ontology is based on the following main items:

- Building Object (Door, Window)
- Building Space

- Command (including OnCommand, OffCommand, PauseCommand, GetCommand, NotifyCommand, SetLevelCommand)
- Commodity (including Electricity, Gas, Water)
- Device (including Switch, Meter, Sensor, Washing Machine)
- Device Category
- Duration Description
- Function (including Actuating Function, Event Function, Metering Function, Sensing Function)
- Function Category
- Profile
- Property (including Energy, Humidity, Light, Motion, Occupancy, Power, Pressure, Price, Smoke, Temperature, Time)
- Service
- State
- Task (including Cleaning, Safety, Entertainment)
- Temporal Entity
- UnitOfMeasure (including Currency, EnergyUnit, Power Unit, Temperature Unit)

It is possible to extend SAREF beyond these items and its definition of devices, to apply more generally to identity management as described in the present document in the form of Authority Attribute Trees.

NOTE: SAREF allows interoperability within a wider Resource Description Framework (RDF), which interoperates with SPARQL for search and discovery, and which consists of a set of *subject-predicate-object* expressions.

The use of semantic discovery as part of identity management is considered in SAREF.

EXAMPLE: A temperature sensor is defined in SAREF as a device that has category saref:Sensor, performs the function saref:SensingFunction and is used for sensing values of the type saref:Temperature. Instead of requesting a device with a non-readable globally unique identity that is known to be a temperature sensor, the sensor can be requested using the combination of these attributes. Extending SAREF to the concept defined in the present document requires the RDF constructs to describe a concept to be extended to the triplet {type : value ; SecurityCredential}

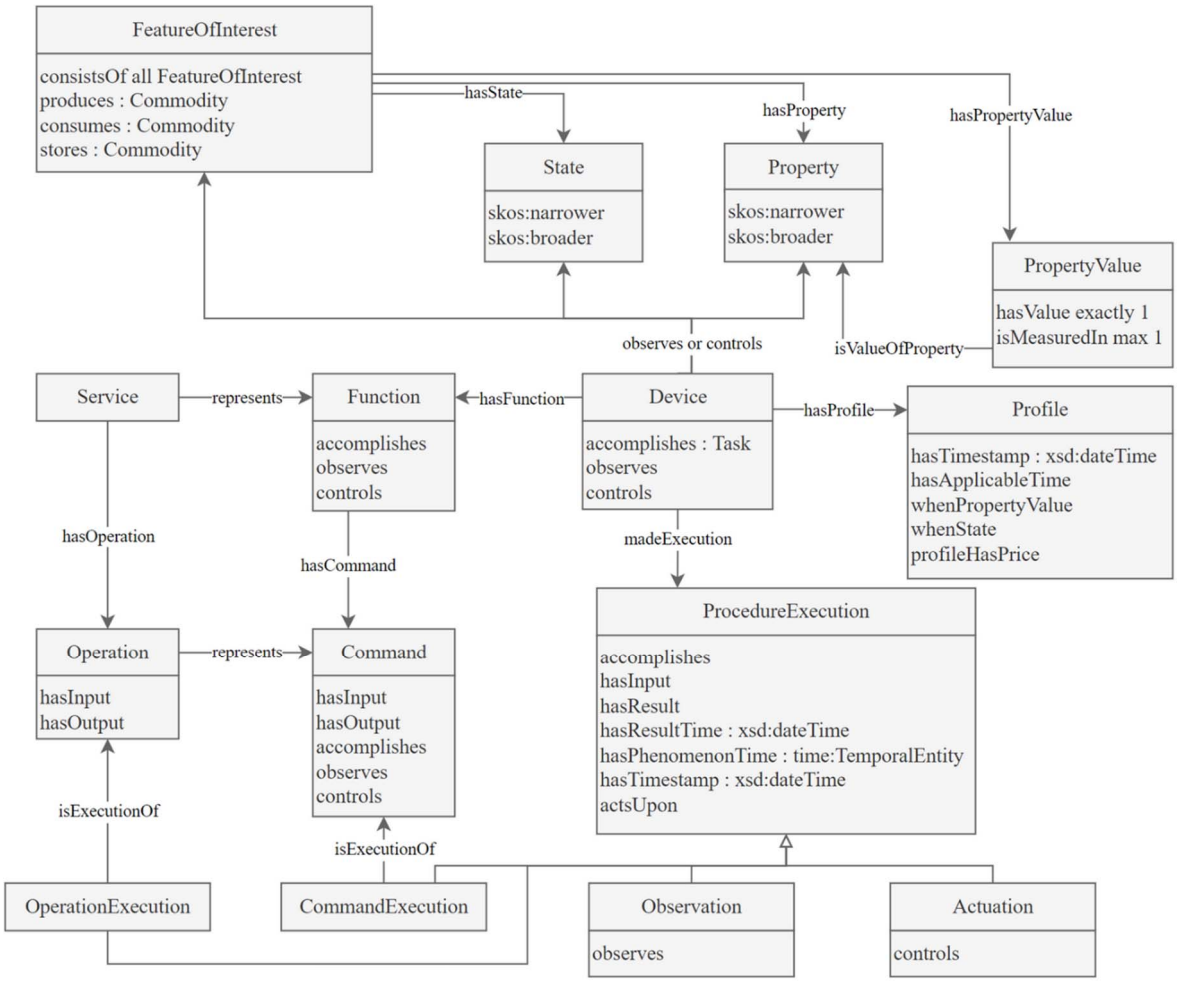

Figure 7: Overview of the SAREF ontology (from ETSI TS 103 264 [i.3])

6.8 Role of Object Identifiers

Object Identifiers (OIDs) may be used to identify attributes of an entity. In such cases, and where the entity is defined in an ETSI standard, the OID should be assigned using the deliverable form as follows:

- itu-t(0) identified-organization(4) etsi(0) document-id (dddd) object (nn)
  - in the example above "dddd" refers to the 4 least significant digits of the ETSI Deliverable number, so for the present document (ETSI TS 103 486) this would be "3486", and nn refers to a specific object defined in the ETSI document, so for the present document a canonical identifier object may be identified as "0".

NOTE 1: The deliverable form of OIDs is described in https://portal.etsi.org/pnns/oidlist.

NOTE 2: A device may also be identified using the scheme of the Combined Nomenclature of the EU (Council Regulation (EEC) No 2658/87) [i.8] but for the purposes of the present document this is not addressed.

OIDs are used in Recommendation ITU-T X.509 [i.7].

Annex A (normative): Cryptographic requirements for attribute-authority trees

A.1 Identity Authentication mechanisms

A.1.1 Overview

A Principal's identity and its identifiers shall be authenticated by the Relying Party by one of the mechanisms defined in ETSI TS 102 165-2 [i.2]. At least one identifier of the Principal shall always be authenticated by the Relying Party before the identifier is used by the Relying Party.

If an identifier has been authenticated, then all its associated identifiers in a single identity should be trusted by association.

A.1.2 Symmetric authentication

To perform symmetric authentication between the Principal and the Relying Party, each party uses a shared secret to authenticate the other. See ETSI TS 102 165-2 [i.2] for a detailed definition of technical means to achieve symmetric authentication.

A.1.3 Asymmetric authentication

To perform asymmetric authentication, the Principal proves that it holds a private key that corresponds to a public key that is tied to its identity. This is commonly achieved by binding the proof in a public key certificate signed by a shared trust authority. See ETSI TS 102 165-2 [i.2] for detail definition of technical means to achieve asymmetric authentication.

A.2 Identity integrity mechanisms

Identity integrity in the context of Authority-Attribute trees shall be measured across each permutation of leaves and the root node.

Integrity proof is commonly created by generating a cryptographic hash of the data (action of the trusted Principal, or indirectly through the identity manager) and making that hash available to the Relying Party. See ETSI TS 102 165-2 [i.2] for detail definition of technical means to achieve proof of integrity.

The integrity mechanism is a refinement of a Merkle tree that utilizes hashes to provide integrity of the data in the tree.

The Merkle tree structure and its properties can be used for identity integrity. Any Relying Party can verify if a known leaf node is part of a Principal's Merkle tree, when presented with the tree's 'root' hash.

EXAMPLE: A Relying Party wants to find a device with a 'camera' attribute. The Principal publishes the root hash of its Authority-Attribute tree (as in clause 5). The Relying Party can verify if the 'camera' attribute is a known leaf node in the Principal's Authority-Attribute Tree.

A.3 Structure of AAT leaf security credential

An AAT leaf credential can take the form of an extended X.509 [i.7] public key credential. The AAT leaf shall identify the attribute authority (see clause 5.1).

A.4 Structure of AAT root security credential

An AAT root credential can take the form of an extended X.509 [i.7] public key credential. The AAT root shall identify the attestation authority (see clause 5.1).

Annex B (informative): Application of attribute-authority trees in obligation of trust protocols

B.1 Introduction to Obligation of Trust protocols

An Obligation of Trust exchange allows parties in a relationship to negotiate the conditions (constraints) that apply to data they share.

In the Obligation of Trust protocol, the Principal and Relying Party (see Figure 1 in the main body of the present document) exchange difficult-to-repudiate Notifications of Obligations (NoO). Each NoO details the scope of one party's requirements for the relationship, informing the other party of the obligations that they are required to accept. Describing these obligations is core to protection of PII.

EXAMPLE 1: A device (the requestor) requests a cloud service. The service provider responds with a Notification of Obligations. This NoO indicates that the requestor needs to provide a username to use the service, which the service provider will delete after 24 hours. The device can send its own NoO, but in this example does not.

The Principal and Relying Party then exchange similarly difficult-to-repudiate Signed Acceptances of Obligations (SAO). Creating the SAO means the Principal and Relying Party assign trust to the relationship in the context of the mutually agreed set of obligations.

EXAMPLE 2: The device sends a SAO that agrees to provide a username to the service provider, agreeing to the condition that it will be deleted after 24 hours. If the device had sent its own NoO, the service provider would respond with its own SAO in this stage too.

In public telecommunications networks, some data is needed to provide a service, such as the type of service requested, where withholding such data means the service does not work. Therefore, that data is made available but can be restricted from further use in the terms and conditions of the service.

EXAMPLE 3: A user issues a NoO that requires that phone identity data is only used in the provision of services to the phone. When the smartphone uses a third-party location service, that service provides its own NoO to the user, as the third party is an extra and separate service.

A NoO can include descriptions of data handling obligations consistent with any applicable regulation or mores derived from one or more of the following principles (summarized from the privacy principles found in the GDPR [i.4] and from ETSI TR 103 305-5 [i.5]) where specific attributes are identified to and attested to using the AAT approach:

- Collection limitation:
  - Limits to data collection
  - Data collection methods
  - Data collection without consent
- Data quality principle
- Purpose specification principle
- Use limitation principle:
  - Use limitation
  - Restriction of disclosure to third parties
  - Use without consent
- Security safeguards principle

A Signed Acceptance of Obligations (SAO) can contain expiry notification(s) of obligation. A NoO can describe release of the signing party from obligations when data is deleted or anonymized.

B.2 Authority-Attribute trees within Obligation of Trust

For the purposes of this annex the example of attributes used to illustrate the obligation of trust are based on attributes of a device. Each attribute of the device is represented as an attested attribute in an AAT leaf (node).

In a NoO the specific permutation of the AAT that the requestor identifies as sufficient to perform the service is made available to the provider, and the agreement of the provider to that permutation of the AAT as an NoO is endorsed in the SAO it sends back.

B.3 Obligation principles

OoT aims to support non-repudiation of consent to processing data. The devices and service providers build a relationship based on SAOs that consent to specific processing of precisely defined data. NoOs can provide a framework to describe the data collection and limitation principles that will be applied to the data (ETSI TR 187 010 [i.1] and ETSI TR 103 370 [i.6]).

History

Document history		
V1.1.1	March 2024	Publication