**ATIS-0x0000x**

ATIS Technical Report on

**Signature-Based Handling of Asserted Information Using Tokens (SHAKEN): Enterprise Identity and Telephone Number (TN) allocation utilizing Distributed Ledger Technology for Originating Service Provider (OSP) Attestation**

**Alliance for Telecommunications Industry Solutions**

Approved April 13th ,2021

**Abstract**

Abstract text here.

**Foreword**

The Alliance for Telecommunications Industry Solutions (ATIS) serves the public through improved understanding between carriers, customers, and manufacturers. The DLT Focus Group initiated to validate key aspects of DLT as it applies to real-world challenges facing today’s communications industry. From a number of potential use cases, one was selected for a more in-depth analysis and proof of concept. The Enterprise Identity Network use case addresses current challenges differentiating robocalls from legitimate calls placed by enterprises.

The mandatory requirements are designated by the word *shall,* and recommendations by the name *should*. Where both a mandatory requirement and a recommendation are specified for the same criterion, the recommendation represents a goal currently identifiable as having distinct compatibility or performance advantages. The word *may* denotes an optional capability that could augment the standard. The standard is fully functional without the incorporation of this optional capability.

Suggestions for improvement of this document are welcome. They should be sent to the Alliance for Telecommunications Industry Solutions, [DLT Focus Group], 1200 G Street NW, Suite 500, Washington, DC 20005.

At the time of consensus on this document, DLT Focus Group which was responsible for its development, had the following leadership organizations:

* Charter Communication
* First Orion
* IBM
* Inteliquent
* Metaswitch
* Numeracle
* TNSI
* T-Mobile
* Somos

The DLT Focus Group was responsible for the development of this document.

**Revision History**

| **Date** | **Version** | **Description** | **Author** |
| --- | --- | --- | --- |
| 6th August 2020 | 1.0 | Initial Contribution for IPNNI | Ian Deakin |
| 26th March 2021 | 1.1 | Updates   * Consideration of Toll Free Numbers * Inclusion of Verifiable credentials for ‘rcd’ claims * Change PASSporT format to remove ‘tnauth’, TN verification is from the Enterprise Identity DID | Ian Deakin |

**Table of Contents**

[1 Scope, Problem 1](#_Toc69994345)

[1.1 Scope 1](#_Toc69994346)

[1.2 Problem Statement 1](#_Toc69994347)

[2 Normative References 3](#_Toc69994348)

[3 Definitions, Acronyms, & Abbreviations 4](#_Toc69994349)

[3.1 Definitions 4](#_Toc69994350)

[3.2 Acronyms & Abbreviations 6](#_Toc69994351)

[4 Overview 8](#_Toc69994352)

[4.1 Enterprise Identity Distributed Ledger Network Interworking with SHAKEN Architecture 13](#_Toc69994353)

[4.2 Overview of Self-Sovereign Identity (SSI) 13](#_Toc69994354)

[4.3 Overview of Decentralized Identifier 14](#_Toc69994355)

[4.4 Overview of Verifiable Credentials 16](#_Toc69994356)

[5 Enterprise Identity Management on Distributed Ledger 17](#_Toc69994357)

[5.1 Creation of a Decentralized Identifier DID and recording of the EIDL 17](#_Toc69994358)

[5.1.1 Identity Creation API 18](#_Toc69994359)

[5.2 Identity Vetting Trust Authority Hierarchy 19](#_Toc69994360)

[5.3 Identity Vetting Authorization Verifiable Credential 20](#_Toc69994361)

[5.3.1 Enterprise Identity Vetting Status 23](#_Toc69994362)

[5.3.2 Organization ‘rcd’ Identifiable Information Verifiable Credential 24](#_Toc69994363)

[5.3.3 Organization Vetting API 26](#_Toc69994364)

[6 TN Authorizations 27](#_Toc69994365)

[6.1 Assignment of a TN by TNSP to TNR or BRAND 30](#_Toc69994366)

[6.2 Delegation of a TN by TNR to Brand / Brand to BPO 33](#_Toc69994367)

[6.2.1 Delegation of a TN by Brand to BPO 37](#_Toc69994368)

[6.2.2 TN Authorization API 41](#_Toc69994369)

[6.3 TN Cancel and Revocation - Credential Claim Status 42](#_Toc69994370)

[7 OSP Attestation of a call using EIDL Network 43](#_Toc69994371)

[7.1 Enterprise PASSporT Encoding 45](#_Toc69994372)

[7.1.1 Use of a modified base PASSporT encoding for Enterprise Identity 45](#_Toc69994373)

[7.1.2 Use of a modified “rcd” PASSporT encoding for Enterprise Identity 46](#_Toc69994374)

[7.2 ‘Enterprise’ PASSporT Verification Procedure 47](#_Toc69994375)

[7.2.1 OSP Verification using a modified base PASSporT 47](#_Toc69994376)

[7.2.2 OSP Verification using a modified “rcd” PASSporT 49](#_Toc69994377)

[8 Traceback using Enterprise Identity 51](#_Toc69994378)

[8.1 Principle #5. Confirm the Identity of Commercial Customers. 51](#_Toc69994379)

[8.2 Principle #6. Require Traceback Cooperation in Contracts 53](#_Toc69994380)

[9 ANNEX A: Example of Open API for EIDL 54](#_Toc69994381)

[9.1 Organization Accounts 54](#_Toc69994382)

[9.1.1 createAccount 54](#_Toc69994383)

[9.1.2 getAccount 56](#_Toc69994384)

[9.2 TNAuthorization 57](#_Toc69994385)

[9.3 Vetter 60](#_Toc69994386)

[9.3.1 createVettedOrganisationOnDLT 60](#_Toc69994387)

[9.3.2 getVettedOrganisationAuthorization 62](#_Toc69994388)

**Table of Figures**

[INSERT]

**Table of Tables**

[INSERT]

# Scope, Problem

## Scope

This specification extends the capabilities of SHAKEN to enable an enterprise to establish ‘Enterprise Identity’ credentials by applying Distributed Ledger Technology and its cryptographic principles. A Know Your Customer (KYC) verified ‘Enterprise Identity' allows the enterprise to prove their identity to any participant of the VoIP eco system to be assigned or delegated 'Trusted' telephone numbers by authorized Telephone Number Service Providers (TNSPs) or Telephone Number Resellers (TNRs). Using a KYC verified ‘Enterprise Identity’ with an authorized telephone number, the enterprise can place calls signed with their ‘Enterprise Identity’ credentials, enabling any Origination Service Provider (OSP) receiving the call, to authenticate the 'Enterprise Identity' of the calling enterprise. The OSP can also determine the signing entity's authorization for using the telephone number from the distributed ledger to mark a call with the SHAKEN “Full” or “A-level” attestation. This distributed ledger infrastructure is called the 'Enterprise Identity Network'.

The ‘Enterprise Identity’ credential is a W3C Decentralized Identifier (DID) recorded on the distributed ledger, authenticated by Public/Private Key pair cryptography. The proof that an ‘Enterprise Identity’ has been KYC vetted will be recorded on the distributed ledger by the issuing authority using signed verifiable credentials, recorded on the distributed ledger according to the W3C Verifiable Credential format.

All authorized TN assignments or delegations will be recorded on the distributed ledger by the issuing authority using signed verifiable credentials recorded on the distributed ledger according to the W3C Verifiable Credential format.

An enterprise will create a SIP Identity header on its outgoing calls containing a PASSporT signed with their ‘Enterprise Identity’ private key and including a reference to the DID credential. The signature and reference will enable any OSP connected to the distributed ledger to authenticate the signed PASSporT using the DID/public key stored on the DLT. The OSP can then verify that the Originating TN being used for the outgoing call is authorized for use by the ‘Enterprise Identity' by checking the signed verifiable credential for the TN.

When the OSP can authenticate the calling ‘Enterprise Identity' and the originating TN is authorized for use, the OSP can apply SHAKEN A-level attestation to the call.

## Problem Statement

The Federal Trade Commission (FTC) and Federal Communications Commission (FCC) regularly cite “unwanted and illegal robocalls” as their number-one complaint category, and caller ID spoofing increases the harm from these unwanted calls. The ATIS-SIP Forum IP-NNI Task Force has developed a specification explicitly designed to mitigate unsolicited and unwanted robocalls by reducing the impact of illegitimate caller ID spoofing. This specification, SHAKEN (Signature-based Handling of Asserted information using tokens), is based on the STIR (Secure Telephone Identity Revisited) protocol developed by IETF. The SHAKEN specification allows the originating service provider to generate a digital signature that securely signals the caller's right to use a phone number to the terminating service provider.

When a call is originated, the originating service provider will use the SIP identity header to contain a digital signature, or token, that will accompany the call as it is being completed. At call termination, the terminating service provider verifies that nothing was tampered with and ensures that the call came from an entity that has a legitimate right to use that number. The verification from SHAKEN can be displayed directly to the user or fed into a "call-blocking app" that provides a rating system that will essentially identify calls as good, questionable, or likely fraudulent. On behalf of the user, the call-blocking app can take action to stop unwanted calls from getting through.

SHAKEN was never intended to be a complete solution for addressing the robocalling problem. It is an important tool in a multi-layered approach. What SHAKEN specifically does is to allow an OSP to populate a SIP Identity header on its calls destined to other service providers. This SIP Identity header contains a cryptographically signed token (the SHAKEN PASSporT) that authenticates the OSP's identity for network traceability and non-repudiation, and protects the integrity of SIP parameters such as the calling TN. One of the parameters, or claims, provided in the SHAKEN PASSporT is an attestation indicator (“attest”) that the OSP marks on the call, indicating its relationship to a “customer” entity and the customer's association with the calling TN. A “Full Attestation” level specifically indicates that a call was received from a known and authenticated customer and that there is a verified “association” of the calling TN to the call. The SHAKEN framework, ATIS-1000074, defines the criteria for Full Attestation as follows:

**A. Full Attestation:** The signing provider shall satisfy all of the following conditions:

1. Is responsible for the origination of the call onto the IP based service provider voice network.
2. Has a direct authenticated relationship with the customer and can identify the customer.
3. Has established a verified association with the telephone number used for the call.

Many call scenarios today will not satisfy these conditions through information locally available to the OSP and may, therefore, receive “partial attestation" (Attestation level “B”) at best. Third-party call centers are a great example of a situation that may not allow full attestation via information locally known to the OSP. Full attestation is not provided in this case due to the original assignment or delegation of a TN to a different entity than the one placing the call. Full attestation may not be provided when the assignment and/or delegation of the TN is coming from a TNSP that is not also the OSP.

SHAKEN does not define what service providers should do with the delivery of a call. Call delivery is governed by existing regulations that generally (with a few exceptions) require that telephone service providers complete all calls. Note, this does not apply to call-blocking apps, which allow the end-user to opt-in and have the app screen, characterize, and potentially block calls on the user’s behalf. In March 2020, the FCC issued a Declaratory Ruling allowing service providers to apply call blocking at the network level on an opt-out basis.

SHAKEN does not characterize calls, and therefore it cannot identify calls correctly or incorrectly. SHAKEN merely creates a digital signature at the origin with the information the OSP knows about the call (customer, and potentially their right to use the number) and sends this signature to the terminating service provider, which verifies that no one has tampered with the information. SHAKEN does not identify calls as “SPAM". Call Validation Treatment (CVT) is the logical function that tries to identify the call intent, and as a result, sometimes label “legitimate” calls as spam.

Robocalls can sometimes be quite valuable to customers in communicating important, need-to-know information that otherwise may not have been received in real-time. A few examples include flight or school cancellations, appointment reminders, and credit card fraud alerts. However, current analytics deployments can label these legal callers as FRAUD/SCAM.

As previously mentioned, these typical call scenarios are complex; an enterprise may contract with a call center provider, who will use another vendor (or vendors) for the calling platform and obtain TNs for the campaign from a telephone number reseller(s) and originate the calls using multiple OSPs (some examples actually use a dozen OSPs) based on time of day and other factors to minimize costs. With this scenario, the OSP has absolutely no confidence in the phone number in the caller ID, or anything else about the call and can only provide “partial attestation”. A solution is needed to provide the ability for an OSP to associate the calling number of a call with the verified 'Enterprise identity' responsible for or sponsoring the content of the call.

# Normative References

The following standards contain provisions which, through reference in this text, constitute provisions of this Standard. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this Standard are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below.

ATIS-10000074, *Signature-based Handling of Asserted Information using Tokens (SHAKEN). [[1]](#footnote-2)*

ATIS-10000080, *SHAKEN: Governance Model and Certificate Management, 1*

ATIS-I-0000076, *Enterprise Identity on Distributed Ledger for Authenticated Caller Use Cases. 1*

IPNNI-2021-00013R00x, SHAKEN: Calling Name and Rich Call Data Handling Procedures. *1*

OpenAPI, *OpenAPI Specification (OAS) Version 3.0.3*

RFC 3261, *SIP: Session Initiation Protocol.[[2]](#footnote-3)*

RFC 4122, *A Universally Unique Identifier (UUID) URN Namespace.**2*

RFC 4949, *Internet Security Glossary, Version 2. 2*

RFC 7044, *An Extension to the Session Initiation Protocol (SIP) for Request History Information*.*2*

RFC 7095, *jCard: The JSON Format for vCard*. *2*

RFC 7515, *JSON Web Signature (JWS). 2*

W3C, Decentralized Identifiers (DIDs) v1.0 [[3]](#footnote-4)

W3C, Verifiable Credentials Data Model 1.0 *3*

W3C, Verifiable Credentials Use Cases *3*

ITU-T Recommendation X.811 (04/1995) | ISO/IEC 10181-2:1996, Information technology – Open Systems Interconnection – Security frameworks for open systems: Authentication framework

NIST SP 800-63-3 - NIST Special Publication 800-63-3 Digital Identity Guidelines

CA/Browser Forum Guidelines for The Issuance and management of Extended Validation Certificates Version 1.6.8).

# Definitions, Acronyms, & Abbreviations

For a list of common communications terms and definitions, please visit the *ATIS Telecom Glossary*, which is located at < <http://www.atis.org/glossary> >.

## Definitions

**BPO Enterprise:** Is a Business Process Outsource (BPO) organization (Call Center) placing calls on behalf of a Brand Enterprise.

**Brand Enterprise:** Is the business entity who is representing the calling party TN, even when the call is placed by a BPO, the caller identity in the SIP call is using the Brand Enterprise name and intended purpose for the call when contacting consumers.

**Caller ID:** The originating or calling party’s telephone number used to identify the caller carried either in the P-Asserted-Identity or From header fields in the Session Initiation Protocol (SIP) [RFC 3261] messages.

**Company Code:** A unique four-character alphanumeric code (NXXX) assigned to all Service Providers [ATIS-0300251].

**Decentralized Identifiers:** (DIDs) are a new type of identifier that enables verifiable, decentralized digital identity (DIDs), and can be authenticated using proofs (for example, digital signatures, privacy-preserving biometric protocols, and so on)

**Distributed Ledger Technology:** (DLT) is a protocol that enables the secure functioning of a decentralized digital database. Distributed networks eliminate the need for a central authority to keep a check against manipulation. DLT allows for storage of all information in a secure and accurate manner using cryptography. Information is stored and verified on the DLT by using asymmetric key cryptographic signatures. Once the information is stored, it becomes an immutable database and is governed by the rules of the network.

**Decentralized Identifier Document:** Also referred to as a DID document, this is a document that is accessible using a verifiable data registry and contains information related to a specific decentralized identifier, such as the associated repository and public key information.

**End-Entity:** An entity that participates in the Public Key Infrastructure (PKI). Usually a Server, Service, Router, or a Person. In the context of this document, an end-entity is a Service Provider, TN Service Provider, or VoIP Entity.

**Enterprise**: For the purposes of this document, a business, governmental body, or non-governmental body that participates in VoIP calling and TN assignments and delegations. In some instances, an “enterprise” may also be an entity providing VoIP calling services and/or infrastructure.

**Identity:** Unless otherwise qualified, an identifier that unambiguously distinguishes an entity for authentication and other security and policy application purposes (compare to “distinguished identifier” as used in X.811 [Ref ITU-T Rec. X.811 (1995 E)]). For the purposes of this report, an identity may or may not be a "“TN-based caller identity",” depending on the context.

**KYC Vetter:** The Know Your Customer (KYC) vetter is the business function used to verify client’s/business identity. The KYC Vetter is a trusted actor of the EIDL managed by the EIDL Governance Authority (GA) to perform verification service according to the policy of the EIDL service. The process of KYC, is to verify the identity of clients before doing business with the entity and on an ongoing basis with them. An EIDL KYC vetter will perform the process of KYC on all TNRs, enterprise Brands, and BPOs with an ‘Enterprise Identity’, to verify that they are compliant with KYC policies of the EIDL governance. A KYC vetter provides a qualified ‘Enterprise Identity’ that can be authenticated by all other actors of the EIDL. This equates to the security management process of “Identity Proofing.” As referenced in NIST SP 800-63-3 - NIST Special Publication 800-63-3 Digital Identity Guidelines, and CA/Browser Forum Guidelines For The Issuance And Management Of Extended Validation Certificates Version 1.6.8).

**National/Regional Regulatory Authority (NRRA):** A governmental entity responsible for the oversight/regulation of the telecommunication networks within a specific country or region.

Note: Region is not intended to be a region within a country (e.g., a region is not a state within the US).

**Numbering Authority:** The entity designated as authoritative over the allocation of a TN to a TNSP for their assignment to end users or other customer entities such as TN resellers or VoIP providers. For full codes and thousands blocks, it is the North American Numbering Plan Administrator (NANPA) under contract to the FCC. For Toll-Free Numbers, it is the Toll-Free Number Administrator (TFNA), which manages the toll-free number registry and the allocation of Toll-Free Numbers (TFNs) to a RespOrg (for their assignment to end users or other customer entities such as TN resellers or other entities), under a tariff from the FCC. For Local Number Portability, it is the Local Number Portability Administrator (LNPA), under contract from the North American Portability Management (NAPM), LLC., and with oversight from the North American Numbering Council (NANC).

**Private Key:** In asymmetric cryptography, the private key is kept secret by the end-entity. The private key can be used for both encryption and decryption [RFC 4949].

**Public Key:** The publicly disclosable component of a pair of cryptographic keys used for asymmetric cryptography [RFC 4949].

**Real-world identity**: Identifiers and identifying characteristics of a principal outside of the telecommunications services domain, including but not limited to personal or business name, physical location or postal address, government-issued identifiers, credit information, etc. Compare to the use of “real-life identity” and “real-world identity” in [NIST SP 800-63-3].

**Responsible Organization (RespOrg):** Entity designated as the agent for the Toll-Free subscriber to obtain, manage and administer Toll-Free Numbers and provide routing reference information in the SMS/800 Toll-Free Number Registry. RespOrgs are the only parties who assign, manage and administer Toll- Free numbers in the Toll-Free Number Registry.

**RespOrg Identification (RespOrg ID):** A 5-character code that designates or points to the Responsible Organization (RespOrg) associated with a specific Toll-Free number [ATIS-0417001-003].

**Secure Telephone Identity (STI) Certificate:** A public-key key certificate used by a service provider to sign and verify the SHAKEN PASSporT.

**Service Provider Code:** In the context of this document, this term refers to any unique identifier that is allocated by a Regulatory and/or administrative entity to a service provider. In the US and Canada, this would be a Company Code as defined in [ATIS-0300251], or a Resp Org ID assigned to a Resp Org as defined in [ATIS-0417001-003].

Editor’s note: Further analysis is required to determine if Resp Org should be included as part of the service provider code or somewhere else.

**Signature:** Created by signing the message using the private key. It ensures the identity of the sender and the integrity of the data [RFC 4949].

**Telephone Identity:** An identifier associated with an originator of a telephone call. In the context of the SHAKEN framework, this is a SIP identity (e.g., a SIP URI or a TEL URI) from which a telephone number can be derived.

**TN Allocation:** The granting of a TN numbering block or individual TN to a TNSP by a TN numbering authority (E.g. NANPA, TFNA).

**TN Assignment:** The procedure for a TNSP to grant the right to use a block or individual TN to a TNR, Brand Enterprise or other entity.

**TN Delegation:** The procedure for a TNSP, TNR, Brand Enterprise or other TN assignee to grant the right to use an individual TN to a Brand Enterprise, BPO or other entity.

**Toll-Free Number Administrator (TFNA):** An entity that is authoritative over the assigning, reserving and releasing of TFNs for public use.

**Toll-Free Number Registry (TFNR):** The main administrative support system of Toll-Free service. It is used to create and update subscriber Toll-Free records that are then downloaded to Service Control Points (SCPs) for handling subscriber’s Toll-Free calls. The system is also used by RespOrgs to reserve and assign TFNs.

**Trust Authority:** The Trust Authority in the Enterprise Identity network is an assigned 3rd party providing three functions. 1. Vetting and authorization of the Enterprise Identity TNSPs and Know Your Customer (KYC) vetters. 2. Vetting and authorization of the EIDL network nodes. 3. Implements the TN authorization rules and process on the EIDL.

**Verifiable Credential:** are the electronic equivalent of the physical credentials that we all possess today, such as: plastic cards, passports, driving licenses, qualifications, and awards, etc. Verifiable Credentials (VC) may be expressed using [JSON](https://en.wikipedia.org/wiki/JSON). A VC is typically composed of: the context of the VC, the identity of the issuer, the date and time of issuing, the expiry date and time, the type of VC, the subject of the VC, one or more identity attributes of the VC subject, and finally the cryptographic proof created by the issuer to ensure the integrity and authenticity of the VC.

**VoIP Entity:** A non-STI-authorized customer entity that purchases (or otherwise obtains) delegated telephone numbers from a TNSP

## Acronyms & Abbreviations

|  |  |
| --- | --- |
| 3GPP | 3rd Generation Partnership Project |
| ATIS | Alliance for Telecommunications Industry Solutions |
|  |  |
| BPO | Business Process Outsource |
| CSCF | Call Session Control Function |
| CVT | Call Validation Treatment |
| DID  DLT | Decentralized Identifier  Distributed Ledger Technology |
| EIDL | Enterprise Identity Distributed Ledger |
| HTTPS | Hypertext Transfer Protocol Secure |
| IETF | Internet Engineering Task Force |
| IP | Internet Protocol |
| JSON | JavaScript Object Notation |
| JWS | JSON Web Signature |
| KYC | Know Your Customer |
| LNPA | Local Number Portability |
| NANPA | North American Numbering Plan Administrator |
| NNI | Network-to-Network Interface |
| OOB | Out of Band |
| OSP | Originating Service Provider |
| PASSporT | Persona Assertion Token |
| PBX | Private Branch Exchange |
| PKI | Public Key Infrastructure |
| RCD | Rich Call Data |
| RespOrg | Responsible Organization |
| SHAKEN | Signature-based Handling of Asserted information using toKENs |
| SIP | Session Initiation Protocol |
| SKS | Secure Key Store |
| SP | Service Provider |
| STI | Secure Telephone Identity |
| STI-AS | Secure Telephone Identity Authentication Service |
| STI-CA | Secure Telephone Identity Certification Authority |
| STI-CR | Secure Telephone Identity Certificate Repository |
| STI-VS | Secure Telephone Identity Verification Service |
| STIR | Secure Telephone Identity Revisited |
| TA | Trust Authority |
| TFNA | Toll-Free Number Administrator |
| TN | Telephone Number |
| TNSP | TN Service Provider |
| TNR | TN Reseller |
| TSP | Terminating Service Provide |
| URI | Uniform Resource Identifier |
| UUID | Universally Unique Identifier |
| VC | Verifiable Credential |
| VoIP | Voice over Internet Protocol |

# Overview

This document presents a method for participants in the VoIP service provider network ecosystem, including TNSPs, TNRs, OSPs, Enterprises, KYC Vetter’s, to record identity information, KYC vetting status and TN assignment and delegation information on a distributed ledger. This service is described as the Enterprise Identity Distributed Ledger (EIDL)

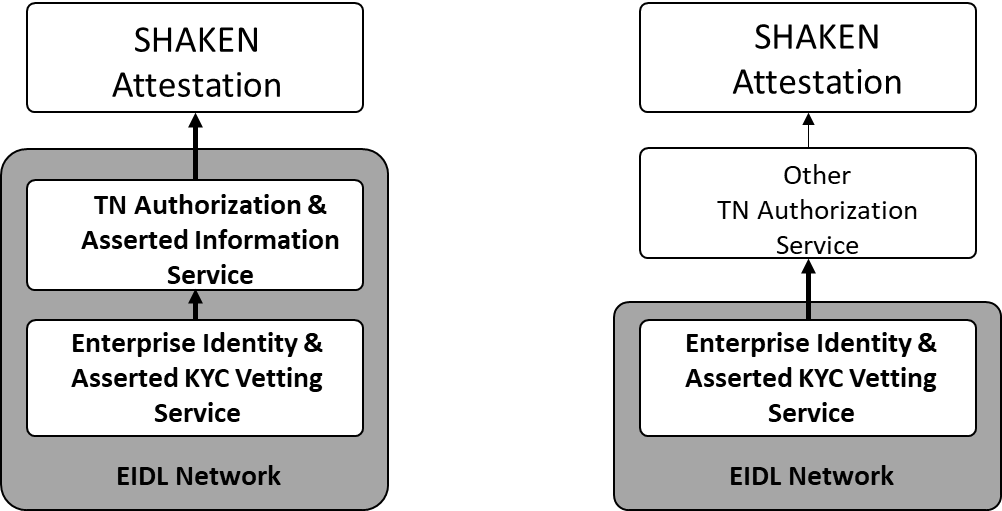
The EIDL performs two key functional services:

* **Enterprise Identity and Asserted KYC Vetting**

A participating organization records their digital identity called the ‘Enterprise Identity’, to be verified and authorized by a KYC Vetter. As described in clause 5 of this document. Proving to all participants of the EIDL, that they have been KYC Vetted in accordance with the EIDL-GA ‘Governance Authority’ policy.

* **TN Authorizations ‘Right to Use’**

Signed TN assignment and delegation information by an authoritative participant is recorded on the EIDL, proving that an ‘Enterprise Identity’ has the ‘Right to Use’ a TN, which can be used to attest calls being placed. The TN authorization function is underpinned by a vetted ‘Enterprise Identity’, which allows the participant authorizing a TN to an ‘Enterprise Identity’ to authenticate that the business representing the ‘Enterprise Identity ‘ is legitimate. As described in clause 6 of this document.

******

***Figure 4.1 EIDL Services supporting SHAKEN attestation***

**Note:** The EIDL ‘Enterprise Identity and Asserted KYC vetting’ service can be used to provide proof of enterprise organization identity and KYC vetting status to other business processes of the VoIP service provider ecosystem. This could be another TN authorization solution that is supporting SHAKEN attestation as illustrated in figure 4.1. through the EIDL APIs as described in Annex A of this document.

**Enterprise Identity Management and KYC Vetting Service**

A participating organization ‘Enterprise Identify’ and their proof of the KYC vetting authorization may be read from an instance of the ledger to determine, authentication of the organization during customer onboarding, TN Authorization and the SHAKEN attestation level to be assigned by the OSP for a call from an originating entity whose information is recorded on the ledger.

The method will illustrate how a participating organization, will create and register an ‘Enterprise Identity’ on the EIDL. How the proof of KYC vetting of an organization’s identity is registered and signed by an authorized Vetter on the EIDL. Allowing any participating organization to verify an organization’s ‘Enterprise Identify’, the vetting status and proof it was performed by which vetting organization from the EIDL.

The method of KYC vetting of a participating organizations identity on the EIDL, is depending on the organization’s role, see clause 5 of this document for role definitions. Some organization roles on the EIDL require the EIDL Trust Authority to KYC verify and authorize their identity. These organizations roles are TNSP’s and KYC vetter’s. Other organization roles on the EIDL have their vetting performed by the ‘KYC Vetter’ role of the EIDL. The organizations that are vetted by the ‘KYC Vetter’ are TNR, Brand Enterprise and BPO Enterprise’.

If the vetting status changes for an organization to ‘unauthorized or ‘revoked’ as covered in clause 6.3 for any reason, this is recorded on the EIDL and is available to be read by all participants.

**TN Authorizations ‘Right to Use’ Service**

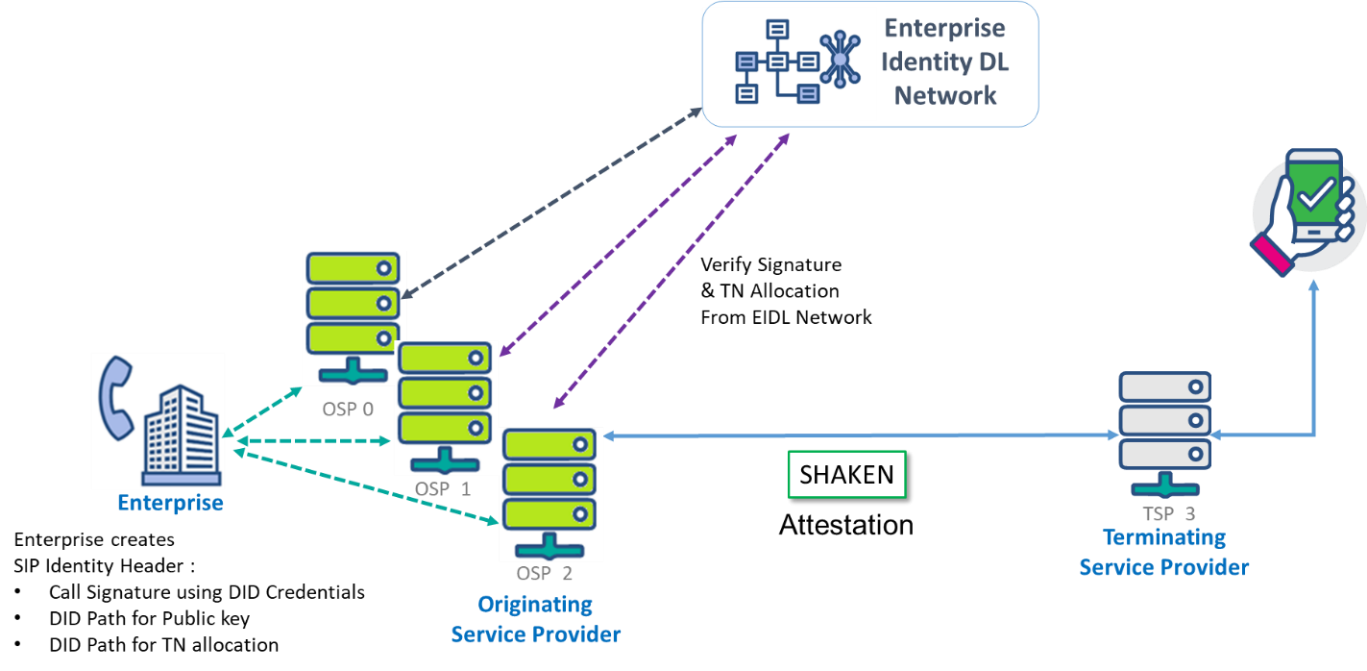
The TN assignment and delegation information may be read from an instance of the ledger to determine the SHAKEN attestation level for a call from an originating entity whose information is recorded on the ledger. This document contains an example PASSporT format that may be used to assert an Enterprise Identity and authorization of the holder of that identity to use a calling TN, and the references in the PASSporT may be used by OSPs to resolve that information from the distributed ledger.

The method will be illustrated by a complex use case involving multiple layers of TN assignment and delegation and calling initiated by a party without a direct relationship to the TNSP or OSPs. This scenario demonstrates a range of capabilities that this method supports, although the method may be used for many different scenarios. The outline of the scenario is as follows:

1. The trust authority will ensure all TNSP’s stay in good standing and will have the ability to revoke any TNSP’s credentials.
2. A TNSP assigns TNs to a TNR or to a Brand Enterprise directly for the Brand’s own general calling purposes
3. The TNR delegates TNs to a Brand/Enterprise for the Brand’s own general calling purposes
4. The Brand Enterprise may further delegate TNs to a Business Process Outsourcing (BPO) Enterprise to conduct an outbound calling campaign on the Brand’s behalf.
5. The calling Brand or BPO initiates calls using one or more OSPs. An OSP will authenticate the contents of a PASSporT that the calling Brand/BPO populates using information from the EIDL. The Brand/BPO may place calls indirectly to the OSPs through another entity such as a Call Processing as a Service (CPaaS) platform provider, which holds the direct relationship with the OSPs.
6. Based on the verification of the Brand/BPO’s PASSporT identity and TN right to use authorization information read from the EIDL, an OSP determines that the calling Brand/BPO’s calls marked with the calling TN are entitled to Full Attestation marking designation in an outgoing “shaken” PASSporT.

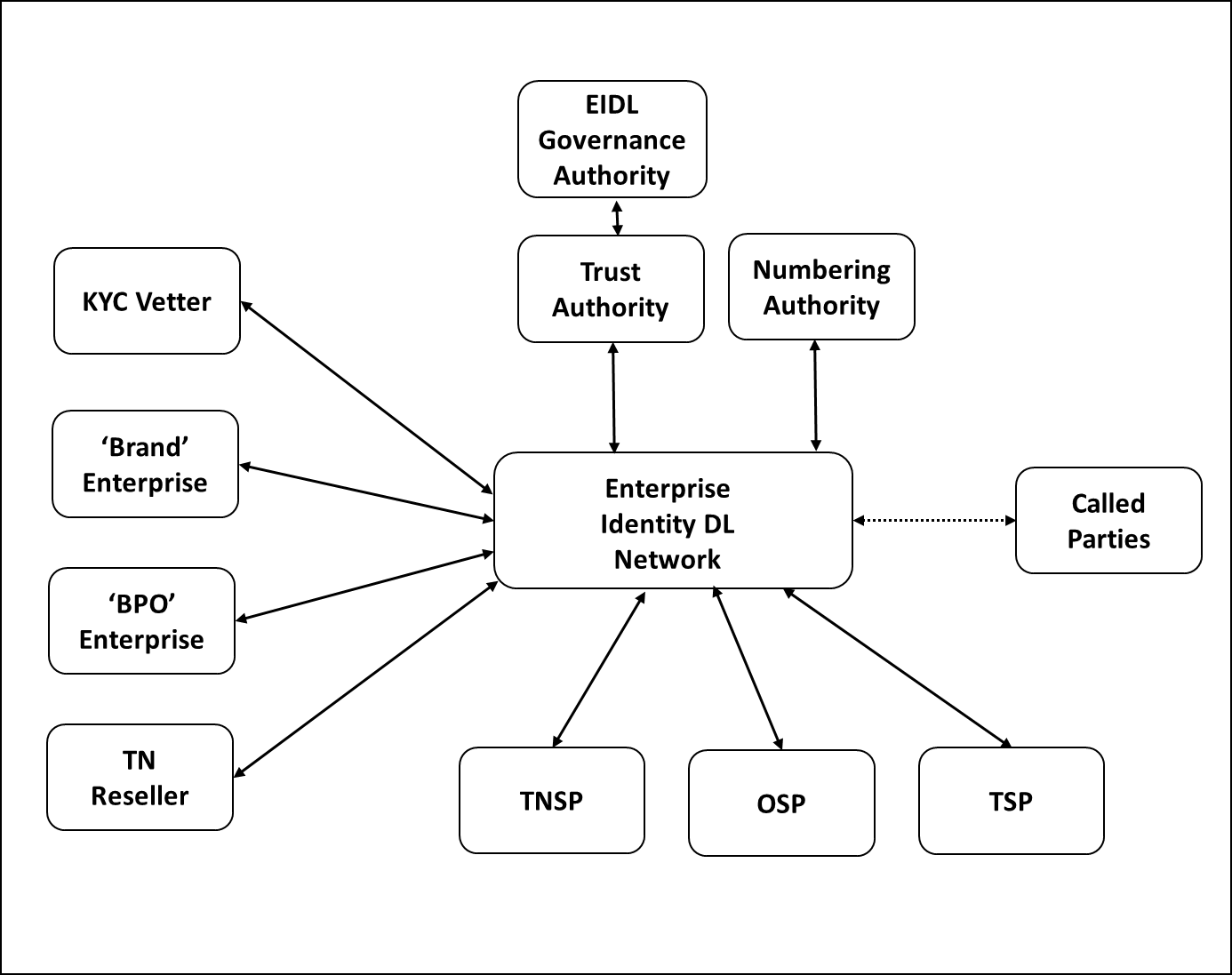
**Enterprise Identity Distributed Ledger Network**

Using distributed ledger technology, the Enterprise Identity Distributed Ledger (EIDL) Network provides control and scale for establishing an Enterprise Identity digital trust foundation, or an identity trust fabric (ITF) to all participating organizations. The EIDL Network enables any participating OSP in the call path to fully attest to a calling enterprise’s SIP call, with an authorized EIDL identity using an authorized TN assignment or delegation. A full description of the EIDL Network is explained in ATIS-I-0000076, *Enterprise Identity on Distributed Ledger for Authenticated Caller Use Cases.*



***Figure 4.2 Enterprise Identity Distributed Ledger Network to attest a Telephone Number***

The diagram illustrated in Figure 4.3 list all organization actors of the EIDL network. These actors may have roles in the EIDL network's Enterprise authentication and calling TN authorization process. For a full description of the EIDL actor roles and functions, refer to ATIS-I-0000076.



***Figure 4.3 Enterprise Identity (EIDL) Network Organization Actors***

In addition to enterprise callers, all organization actors of the EIDL network must have an ‘Enterprise Identity‘ to transact on the network. An organization's ‘Enterprise identity' on the EIDL is implemented using decentralized identifiers according to the [W3C Decentralized Identifiers]

The Decentralized Identifiers credential is generated with a public/private key pair by the organization actor. The private key is securely stored within the organization actors SKS (secure Key Store). The Public Key is stored within the DID on the EIDL.

The benefit of using DID for identities is that it enables the underlying distributed ledger technology to be agnostic and allows interoperability across other identity networks as a federated identity service to support cross border identity and TN authentication.

Once an Enterprise organization DID is created and registered on the EIDL, the enterprise can use a network of KYC vetters to verify and prove their identity. When the KYC Vetter has validated the Enterprise’s real-world identity according to the governance policy of the EIDL, the KYC Vetter will sign a verifiable claim for the organization’s DID indicating they are now 'approved'.

With a KYC vetted and approved DID, a TN can be assigned by a TNSP or Delegated by a TNR to a Brand to place calls, or delegated by a Brand to a KYC vetted and approved BPO to place calls on their behalf.

All TN allocations (assignments or delegation) on the EIDL are represented and contained within verifiable credential (VC) claims according to [W3C Verifiable Credentials]. These verifiable credential claims are digitally signed assignments or TN delegation from the authorized TNSP to a TNR to a Brand Enterprise and to one or more BPO (call Centers). Each TN is encapsulated in a single VC. As the TN is assigned from the TNSP to TNR, TNR to Brand, Brand to BPO, each organization assignment or delegation is cryptographically signed within the VC to prove that claim. A single TN DID provides the full chain of custody for the TN by a TNSP; who assigned it, at what time, and by whom. As these Verifiable Credential claims are on the distributed ledger, they are all true at any point in time as an authoritative source for all stakeholders against which they can verify.

***Note:*** Throughout this document when the term TNSP is used, it also refers to a RespOrg.

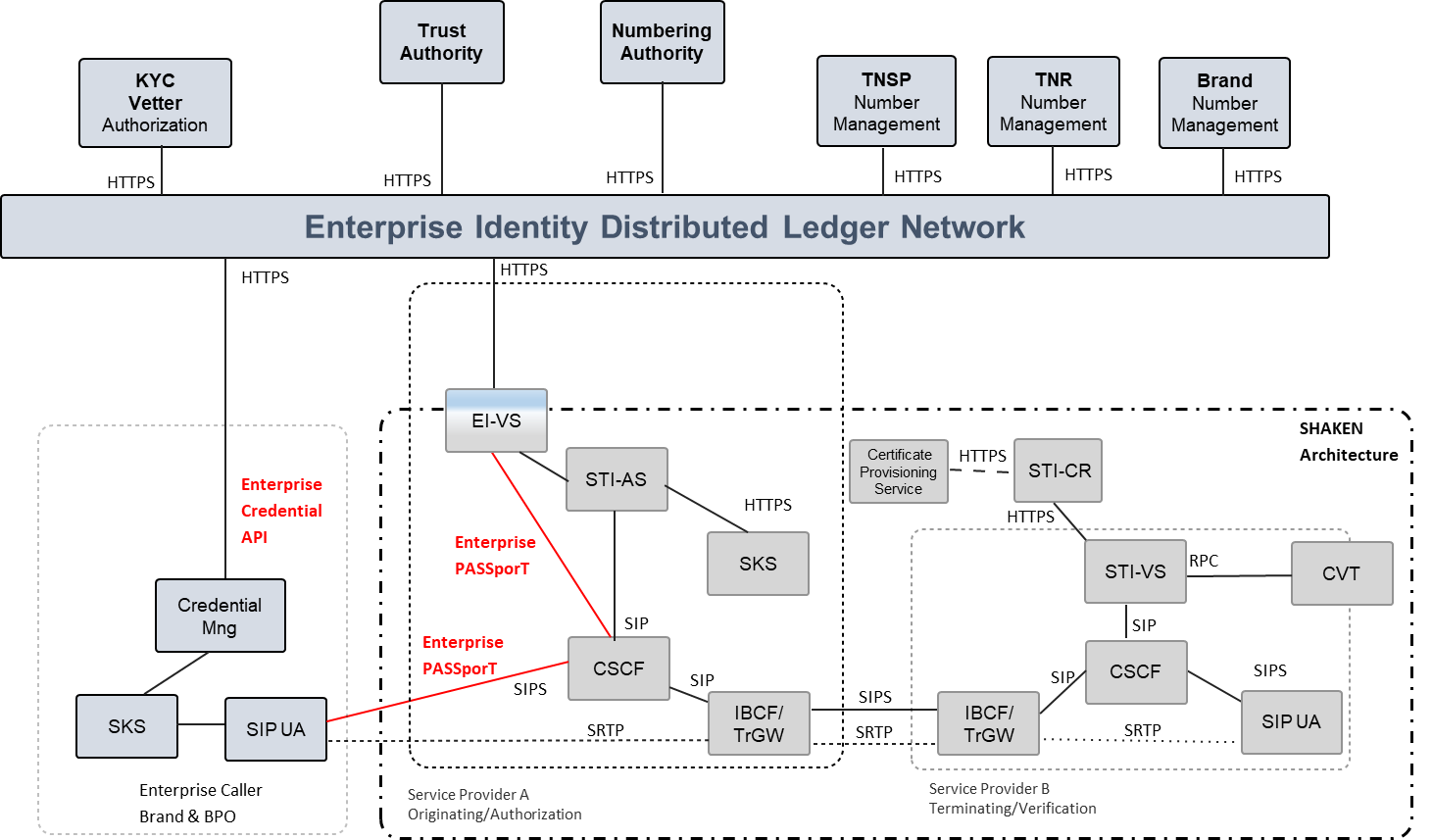
Using their DID, a KYC vetted enterprise will sign originating phone calls using the SIP Identity header. This signature enables any OSP connected to the EIDL Network to verify the calling Enterprise’s DID and prove that the call is being placed only by a 'Trusted' enterprise.  Using this proof of identity, the OSP can then verify that the originating TN is being used by an Enterprise that has the authoritative right to use it by checking the TN’s verifiable credential on the EIDL Network. This VC enables an OSP to authenticate an Enterprise Identity and prove that they are entitled to make calls from the calling TN.  With this proof of identity and the number being used, the OSP can attest to the call using SHAKEN A-level attestation.

An 'Enterprise Identity' implemented with distributed ledger technology, using DID and Verifiable credentials, will significantly enhance the capability for Originating Service Providers to validate and authenticate the calling party identity and the authority to place a call with an authorized TN. OSP’s will be able, to provide SHAKEN A-level attestation for enterprise calls in complex ~~the~~ call scenarios described above.

This document will cover the implementation of the DID for Enterprise identity verification and TN authorization. For more information on the EIDL implementation, see the ATIS-I-0000076, *Enterprise Identity on Distributed Ledger for Authenticated Caller Use Cases,* technical report.

## Enterprise Identity Distributed Ledger Network Interworking with SHAKEN Architecture

The following architecture illustrates the Enterprise Identity Distributed Ledger network architecture interworking with the SHAKEN architecture to provide '‘Enterprise Identity' services to OSPs.



***Figure 4.4 Enterprise Identity (EIDL) Network Interworking with SHAKEN Architecture***

## Overview of Self-Sovereign Identity (SSI)

Self-sovereign identity (SSI) is a decentralized user-centric model that gives the user full control of their own identity and information. In other words, the user is the owner and the administrator of his digital identity and their identity is independent of any service provider. The implementation of SSI requires three concepts, decentralized identifiers, decentralized identifiers documents, and verifiable claims. SSI implies that an individual or organization with one or more identifiers or DIDs, can present specific claims or credentials relating to those DIDs without going through an intermediary. An SSI is not bound to a single identity issuer, therefore enabling the use of SSI to be portable across multiple credential issuers.

An EIDL Network Actor SSI can create an account on the EIDL, generating a decentralized identifier credential with a public-private key pair. The DID and Public key are published to the EIDL for authentication when transacting with other actors on the EIDL network.

## Overview of Decentralized Identifier

The concept of a globally unique decentralized identifier is not new. Universally Unique Identifiers (UUIDs) were first developed in the 1980s and later became a standard feature of the Open Software Foundation’s Distributed Computing Environment. UUIDs achieve global uniqueness without a centralized registry service by using an algorithm that generates 128-bit values with sufficient entropy that the chance of collision is infinitesimally small. UUIDs are formally specified in [RFC4122] as a specific type of Unified Resource Name (URN).

A DID is similar to a UUID except that:

* Like a URL, it can be resolved or dereferenced to a standard resource describing the subject. That is a DID document.
* Unlike most resources returned when dereferencing URLs, a DID document usually contains cryptographic material enabling authentication of the DID subject.

A Decentralized Identifier (DID) is a globally unique identifier that does not require a centralized registration authority and is created in a common trust domain called an Identity Trust Fabric (ITF). The ITF stores the proof of identities and their verifiable credentials cryptographically and immutably on the blockchain. The ITF is where supply-chain partners can verify the authenticity of an identity as well as related verifiable credentials. The ITF is the component that circumvents the need for a central identity provider to manage trust. Once a decentralized identity is established, any supply-chain partner can verify relevant attributes regarding another supply-chain partner with which it is about to engage in a business interaction, either by granting access or conducting a transaction.

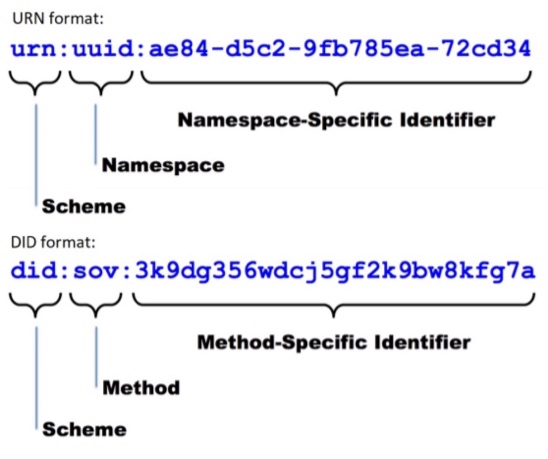
Entities are identified by decentralized identifiers (DIDs); and authenticated using proofs (for example, digital signatures, privacy-preserving biometric protocols, and so on). DIDs point to DID documents. A DID document contains a set of service endpoints for interacting with the entity that the DID identifies (that is, the DID subject). Following the guidelines of Privacy by Design, any entity can have as many DIDs (and corresponding DID documents and service endpoints) as necessary to respect the entity’s desired separation of identities, personas, and contexts (in the ordinary sense of these words).

DID methods are the mechanism by which a DID and its associated DID document are created, read, updated, and deactivated on a specific distributed ledger or network. DID methods are defined using separate DID method specifications.

This design eliminates dependence on centralized registries for identifiers and centralized certificate authorities for key management, which is the standard in hierarchical PKI (public key infrastructure). In cases where the DID registry is a distributed ledger, each entity can serve as its own root authority.

**DID Path format**

DID format is similar to the Uniform Resource Name (URN) format. The key difference is that we are specifying a method instead of a namespace in the DID format. The method is referring to the specification ‘W3C, Decentralized Identifiers (DIDs) v1.0' used to register and manage DIDs on a particular blockchain or DLT (Distributed Ledger Technology).



***Figure 4.5: DID format (Example Sovrin network method) compared to URN format***

**DID Document**

We can see DID infrastructure as a global key-value database where the database consists of all DID-compatible distributed ledgers. In this global and distributed database, the key is a DID, and the value is a DID document. A DID document is a JSON-LD file which stands for JavaScript Object Notation for Linked Data. The main advantage of JSON-LD is interoperability.

The DID specified in the DID Document includes the following six parts:

1. The DID itself, so the DID document is fully self-describing, and we can link other documents to this DID.

2. A set of public keys or other proofs that can be used for authentication or interaction with the identified entity.

3. A set of service endpoints that describe where and how to interact with the identified entity. A service endpoint could, for example, link to a telephone number of the DID’s owner.

4. The owner of the DID can also decide to authorize other entities to make changes to the DID Document. This can become particularly important and useful in the case of private key loss.

5. Timestamps can be added for auditing the date of creation or update of a particular DID.

6. A JSON-LD signature is added if there is a need to verify the integrity of the document.

The format of an enterprise identity DID document is:-

{

"@context": ["https://www.w3.org/ns/did/v1", "https://w3id.org/security/v1"],

"id": "did:example:<BRAND1>",

" verificationMethod ": [{ "id": "did:example:<BRAND1>#key-1",

"type": "Ed256VerificationKey2018",

"controller": "did:example:<BRAND1>,",

"publicKeyJwk": {

“kty":"EC",

"crv":"P-256",

"x":"f83OJ3D2xF1Bg8vub9tLe1gHMzV76e8Tus9uPHvRVEU",

"y":"x\_FEzRu9m36HLN\_tue659LNpXW6pCyStikYjKIWI5a0" }]

}

***Note:*** in this format of a self-sovereign identity, the controller DID is the subject of the DID.

## Overview of Verifiable Credentials

A Verifiable Credential claim is a qualification, achievement, quality, or piece of information about an entity's background, such as a name, government ID, payment provider, home address, or university degree. Such a claim describes a quality or qualities, property or properties, of an entity that establish its existence and uniqueness. The use cases outlined here are provided to make progress toward possible future standardization and interoperability of both low and high-stakes claims with the goals of storing, transmitting, and receiving digitally verifiable proof of attributes such as qualifications and achievements.

There are four roles supported by verifiable credentials: Issuer, Verifier, Subject, and Holder.

**Issuer:** The entity that creates a claim and associates it with a particular subject.

**Verifier:** The entity verifying a claim about a given subject.

**Subject:** The entity about whom a claim is issued.

**Holder:** A role an entity may perform by possessing one or more verifiable credentials. A holder is usually, but not always, the subject of the verifiable credentials that they are holding. Holders store their credentials in credential repositories.

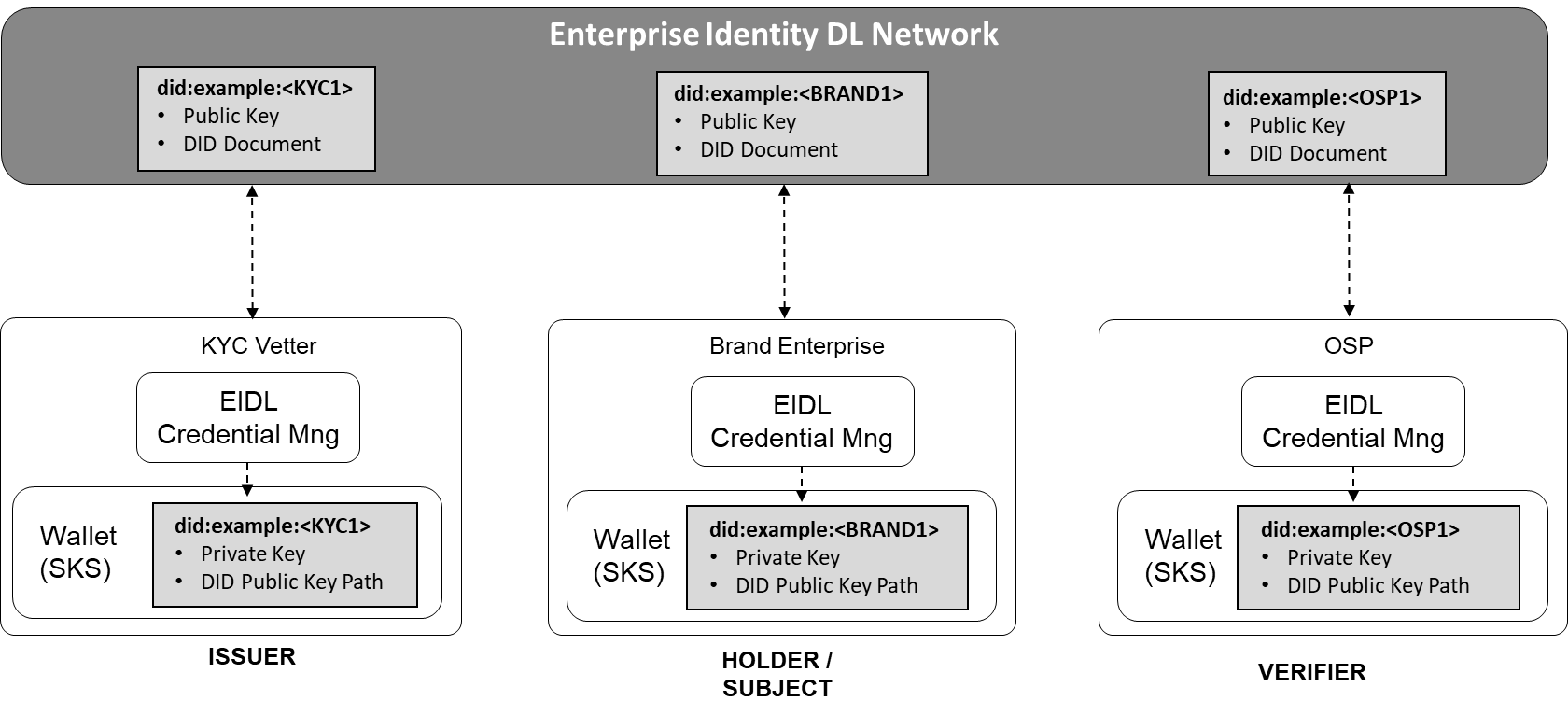
For more details on the Verifiable Credential claim syntax see 'W3C, Verifiable Credentials Data Model 1.0.'

# Enterprise Identity Management on Distributed Ledger

This clause describes the specific implementation of the W3C Decentralized Identifiers and Verifiable Credentials on the Enterprise Identity distributed ledger network to provide organization actor identification and proof of KYC vetting status for participants of the EIDL and caller TN attestation for the OSP. For more details on the Enterprise identity use cases, architecture, actors and functional flows, see ATIS-I-0000076.

## Creation of a Decentralized Identifier DID and recording of the EIDL

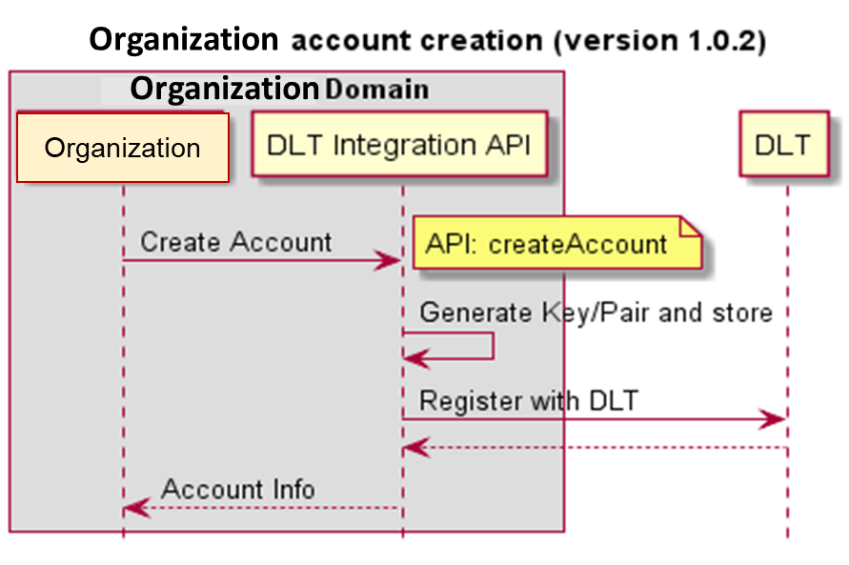
Each Actor of the EIDL Network must generate a Decentralized Identifier. The Actor will create a DID identity and publish this onto the EIDL as a DID Document with its public key. The EIDL will return the DID Path, which is the address for the public key on the EIDL.



***Figure 5.1 Creation of DID on EIDL Network***

### Identity Creation API

Using the Enterprise Identity API definition described in Annex A of this document, an organization's client-side ~~side~~ credential manager will integrate with the EIDL to generate their DID credentials and a corresponding key pair. Here is an example of the organization's client-side API generating a key pair and registering with the EIDL.



***Figure 5.2 Creation of DID on EIDL Network***

## Identity Vetting Trust Authority Hierarchy

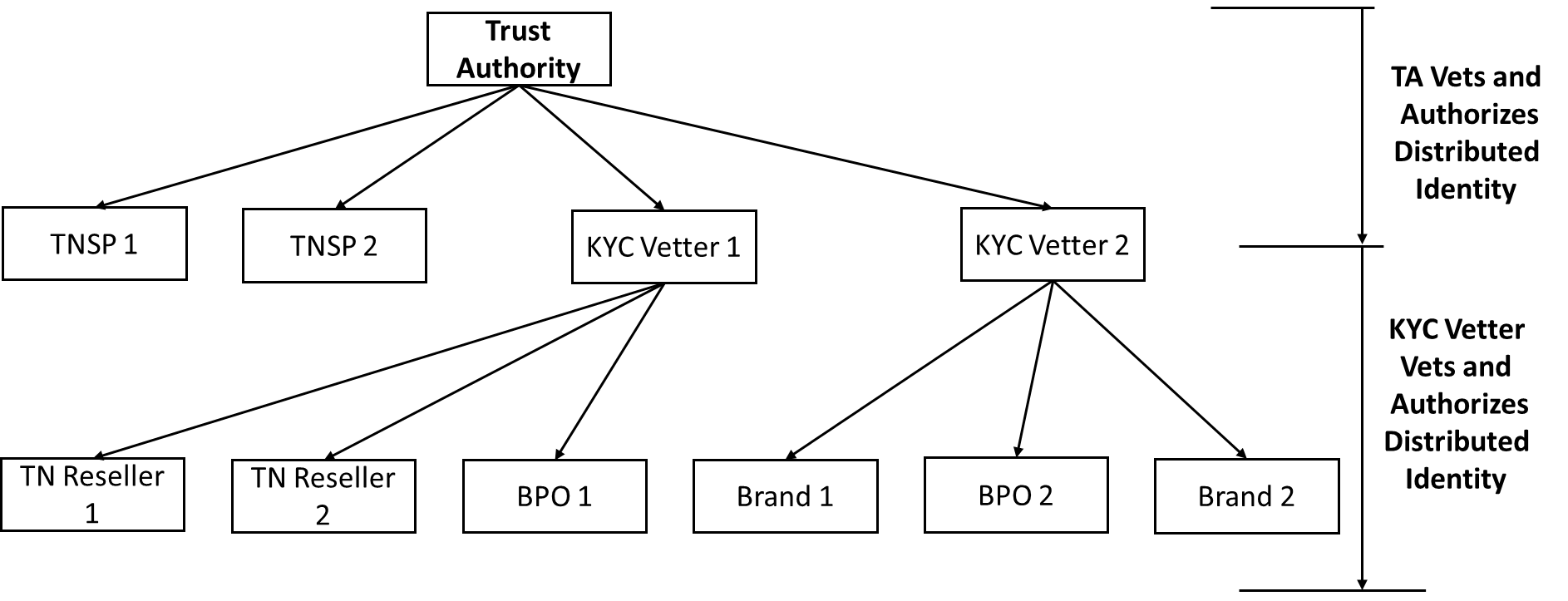
Every actor of the EIDL Network must have their DID authorized to participate in the Enterprise Identity service. Using the organization DID, the requesting organization Actor will apply to the appropriate vetting authority to have their DID authorized to participate.

The Trust Authority (TA) role is to verify the organizations that represent the TNSP and KYC Vetter actors of the EIDL Network. The vetting authorization process is according to the policy defined by the EIDL Network Governance Authority. When the TA has authorized an organization, the TA will create a verifiable credential claim on the EIDL for that organization's DID to indicate that they are authorized.

The KYC Vetter’s role is to verify the organizations that represent the TNR, Brand Enterprise, and BPO Enterprise actors of the EIDL Network. The vetting authorization process is according to the policy defined by the EIDL Network Governance Authority. When the KYC Vetter has authorized an organization, the KYC will create a verifiable credential claim on the EIDL for that organization's DID to indicate that they are authorized.

The TNSP role is to assign TNs that is under their control to TNR and Brand organization that have been KYC vetted and authorized on the EIDL. The right to assignment of TN’s by a TNSP must comply with the policy of the appropriate Numbering authority for the given TN. When a TNSP assigns a TN to a TNR or Brand organization, the TNSP will create a verifiable credential claim on the EIDL for that organization's DID to indicate that they are authorized to use that TN.

***Note:*** a KYC Vetter must be approved by the TA before they themselves can authorize other organizations



***Figure 5.3 EIDL Network Trust Hierarchy***

The issuer (TA or KYC Vetter) of the Verifiable claim for the organization will contain data attributes that indicate the organization's business function, for example: -

* Name of Business = “Company Name” – Assigned in KYC Process
* Nature of Business (TNSP, KYC Vetter, TN Reseller, Brand, BPO)
* Vetting Identity = Identity (Unique Allocation i.e., UUID) – Assigned by DLT
* Current Authentication Status = Authorized, Not Authorized, Revoked
* Vetter Name = "“Vetter Business Name"” – Assigned by TA

## Identity Vetting Authorization Verifiable Credential

The Enterpriser Identity vetting status by authorized vetting actors on the EIDL network are contained within W3C verifiable credential claims. A verifiable credential is a tamper-evident credential with authorship that can be cryptographically verified proof of the Enterprise Identity using the public key of the actors Enterprise Identity DID. When an authorized KYC vetter or TA on the EIDL has vetted the Enterprise Identity, a VC claim is created for that indicating the vetting status for that Enterprise Identity and is added to the Enterprise Identity DID for proof.

**Vetting Status Verifiable Credential**

A vetting Status verifiable credential contains the following claims:

**Issuer:** The actor enterprise identity DID that provides vetting of Enterprise Identities on the EIDL and creates a verifiable claim of the SUBJECT Enterprise Identity DID vetting status. This will be the Enterprise Identity DID of either the KYC Vetter or TA actor

Example of the “issuer” claim: -

"issuer": "did:example:<**KYC1**>",

**Subject:** The subject Enterprise Identity DID that the vetting credential is for and the vetting status.

Example of the “credentialSubject” claim:-

"credentialSubject": {

"id": "did:example:<**BRAND1**>,vettingstatus",

"organisationVettingStatus": "**Authorized**"

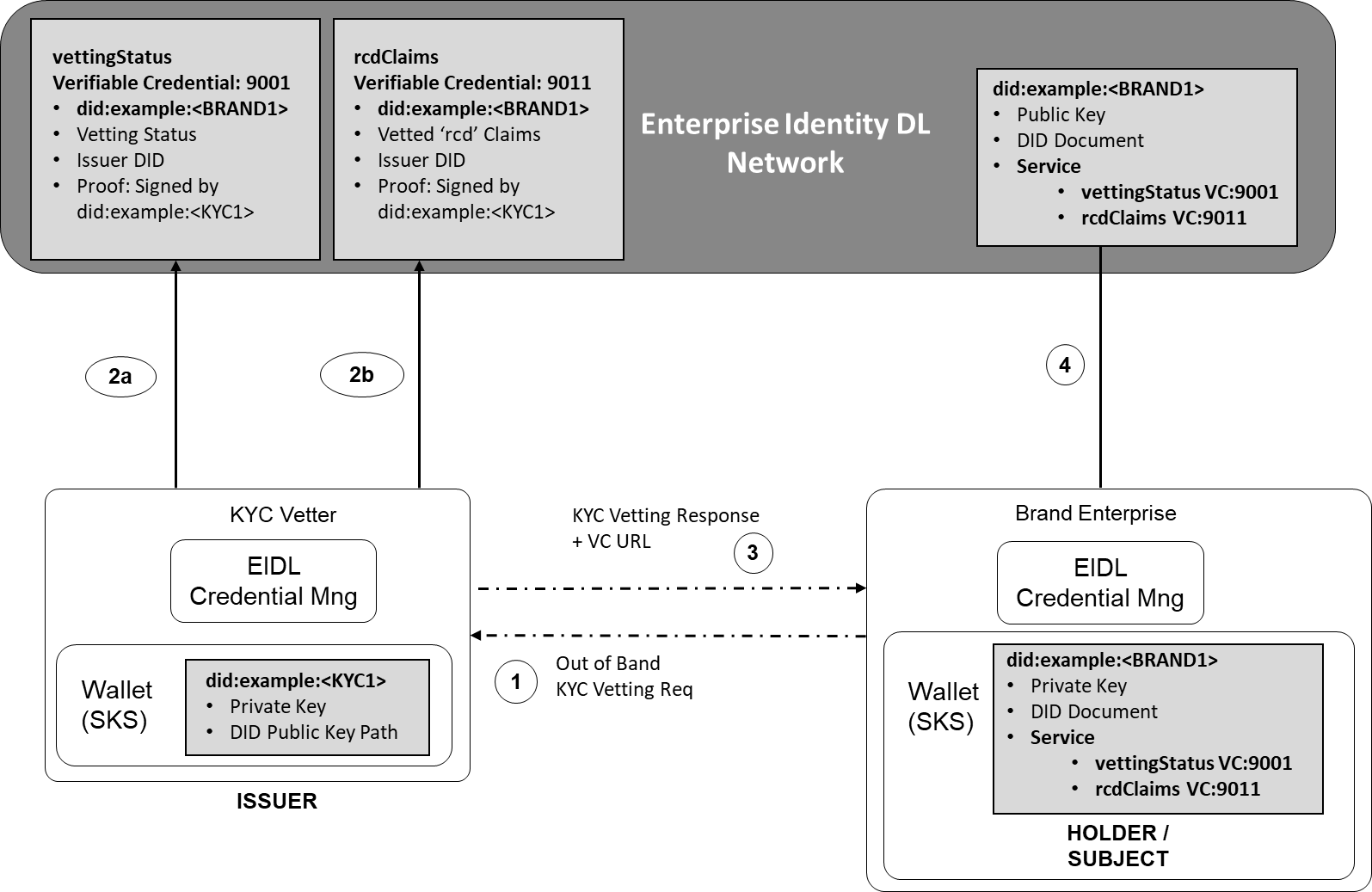
**Vetting Authorization Process**

As indicated in the Trust Hierarchy, the TA must authorize a TNSP, KYC vetter, and OSP to perform their function on the EIDL network. Only an Authorized KYC vetter can then vet and authorize TNR’s, Brand Enterprise’s, and BPO enterprise actors to perform their function on the EIDL.

1, Using an out-of-band (OOB) request (as indicated in option 1 below), the Organization will request the Vetter Actor to perform vetting of their business for authorization to use the EIDL Network services, according to the agreed policy of the EIDL (out of scope for this document at this time).

2a, The vetting authority (TA or KYC Vetter), will perform vetting of the requesting organization according to the agreed EIDL policy. If authorized, the vetting authority will generate and sign a Verifiable Credential claim, indicating the organization DID and that their authorization status in now 'authorized.' The vetter returns the status to the requesting organization together with the VC claim identity to support this.

2b, Optionally, if the requesting organization has ‘rcd’ information that is to be associated with their Enterprise Identity DID. The vetted authority ( TA or KYC Vetter ) , will perform vetting of ‘rcd’ information to be associated with the requesting organization. If authorized, the vetting authority will generate and sign a Verifiable Credential claim for the supporting ‘rcd’ information claims for the organization DID. The vetter returns the status to the requesting organization together with the VC claim identity to support this.



***Figure 5.4 Request Org to be KYC Vetted and create Vetted VC Claim***

In the example outlined in figure 5.4, the Brand Enterprise (BRAND1) has been authorized by the KYC Vetter (KYC1). KYC Vetter (KYC1) has generated a signed VC Claim to indicate this on the EIDL which can be checked by any other actor on the EIDL. As part of the authorization process, the KYC Vetter will verify the Enterprise Brand Name that will be used to identify them. If the Enterprise passes the KYC vetting process, then the KYC vetter will generat~~e~~ a verifiable credential claim on the EIDL, indicating that the DID of the enterprise is authorized. Signing the VC with their private key for proof.

**Organization Vetting Status Verifiable Credential Claim**

Example of a W3C verifiable credential claim representing the vetting status of an organization.

{

"@context": [

"https://www.w3.org/2018/credentials/v1",

"https://access.atis.org/apps/group\_public/dlt/v1/"

],

"id": **"https://access.atis.org/credentials/9001"**,

"type": ["VerifiableCredential", "VettingStatus"],

"holder": "did:example:<**BRAND1**>",

"issuer": "did:example:<**KYC1**>",

"issuanceDate": "2020-01-01T19:73:24Z",

"credentialSubject": {

"id": "did:example:<**BRAND1**>,vettingstatus ",

"organisationVettingStatus": "**Authorized**"

},

"proof": {signed by KYC vetter}

}

3, The KYC vetter will reply to enterprise organization indicating the URL of the verifiable credential on the EIDL that supports their KYC vetting approval. In this example "https://access.atis.org/credentials/9001", the issuer KYC1, has authorized the enterprise BRAND1.

4, The enterprise will update their DID document on the EIDL with the service that indicates that they have been KYC vetted, using the URI of the verifiable credential supporting the vetting authorization.

"@context": ["https://www.w3.org/ns/did/v1", "https://w3id.org/security/v1"],

"id": "did:example:<**BRAND1>**",

"publicKey": [{

"id": "did:example:123456789abcdefghi#key-1",

"type": "Ed25519VerificationKey2018",

"controller": "did:example:<BRAND1>",

"publicKeyJwk": {

“kty":"EC",

"crv":"P-256",

"x":"f83OJ3D2xF1Bg8vub9tLe1gHMzV76e8Tus9uPHvRVEU",

"y":"x\_FEzRu9m36HLN\_tue659LNpXW6pCyStikYjKIWI5a0"

}]

"service": [{

"id":"did:example:<BRAND1>#Key-1;vettingstatus",

"type": "VerifiableCredentialService",

"serviceEndpoint": "**https://access.atis.org/credentials/9001”**

}]

}

### *Enterprise Identity Vetting Status*

There are three possible identity vetting status options recorded in an organization’s identity verifiable credential claim that is declared in the "organisationVettingStatus" Claim.

*‘Not Authorized’,*

* Where the organizations identity has conducted a vetting process, but has not been authorized by the TA or a KYC vetter.
* Changed from *‘Authorized’* to *‘Not Authorized’* by the TA or a KYC vetter, when vetting authorization has expired and the TA or KYC vetter needs to perform a new vetting process on the origination identity. The valid vetting authorization period of an organization vetting status is defined by the EIDL Governance Authority policy.

*‘Authorized’*

* Where the organizations identity has been vetted and is authorized by the TA or a KYC Vetter. The vetting authorization criteria of an organizations vetting status is defined by the EIDL Governance Authority policy.

*‘Revoked’.*

* Where the organizations identity has been revoked by the TA or a KYC vetter. The reason for revocation of an organizations vetting status is defined by the EIDL Governance Authority policy.

Example of a W3C verifiable credential claim representing the ‘Revoked’ vetting status of an organization.

{

"@context": [

"https://www.w3.org/2018/credentials/v1",

"https://access.atis.org/apps/group\_public/dlt/v1/"

],

"id": **"https://access.atis.org/credentials/9001"**,

"type": ["VerifiableCredential", "VettingStatus"],

"holder": "did:example:<**BRAND1**>",

"issuer": "did:example:<**KYC1**>",

"issuanceDate": "2020-01-01T19:73:24Z",

"credentialSubject": {

"id": "did:example:<**BRAND1**>",

"organisationVettingStatus": "**Revoked**"

},

"proof": {signed by KYC vetter}

}

### *Organization ‘rcd’ Identifiable Information Verifiable Credential*

During the vetting process an organization will provide identifiable information about their business and function, that can be used a verified rich call data ‘rcd’. For example, the business brand name, business logo, business function, call back number. The vetted organization identifiable information can be included into a ’rcd’ verifiable credential which is used for the verification of ‘rcd’ enhanced calling party data presented in a modified rcd PASSporT described in clause 7, such as name, address, photos, logos, and other information that may be extended in the future. The following claims are used to prove that the organizations identifiable information has been vetted.

“organizationType”:

The vetted organization actor type on the EIDL. Either TNSP, KYC Vetter, TN Reseller, Brand or BPO

“organizationName”:

The name is the vetted organization brand name that will be used for enhanced calling party data. This claim can be assigned to a TN authorization VC as described in clause 6 to provide verification of the calling party name presented in a modified rcd PASSporT covered in clause 7.

“organizationLogo”:

The URL of the vetted organization brand identity logo, that will be used for enhanced calling party data. This claim can be assigned to a TN authorization VC as described in clause 6 to provide verification of the calling party ‘jcd’ claim including the URL for the logo in a modified rcd PASSporT covered in clause 7.

“organizationJcard”:

The URL of the vetted JSON encoded jCard of the organization brand identity information, that will be used for enhanced calling party data. The JSON format or vCard defined in [RFC7095] which is itself an extensible JSON object for the transport of personal identifiable types of information. This claim can be assigned to a TN authorization VC as described in clause 6 to provide verification of the calling party ‘jcl’ key value presented in a modified rcd PASSporT covered in clause 7.

“callingDataDigest”

Contains a SHA256 digest of the “rcd” claims that will be used by the calling party, calculated by the vetter. [draft-ietf-stir-passport-rcd] specifies how the " claim of the "rcd" PASSporT is used to protect the integrity of the rich call data from being maliciously modified. The "rcdi" claim contains a digest that is calculated across all of the rich call data; i.e., the input to the digest calculation is the “rcd” claim contents, plus any resources referenced by the "rcd" claim contents, plus any resources referenced by the referenced resources, and so on.

As described in clause 5.4 , the requesting organization will request a KYC vetter to verify and authorize their associated ‘rcd’ information to be associated to the actors DID.

1, Using an out-of-band (OOB) the organization will request the Vetter Actor to perform vetting of their business ‘rcd’ information for authorization to use the EIDL Network services, according to the agreed policy of the EIDL (out of scope for this document at this time).

2b, The vetting authority (TA or KYC Vetter), will perform vetting of the requesting organizations ‘rcd’ information according to the agreed EIDL policy. If authorized, the vetting authority will generate and sign a Verifiable Credential claim, indicating the organization DID and that their vetted ‘rcd’ information. The vetter returns the status to the requesting organization together with the VC claim identity to support this.

Example of a W3C verifiable credential claim representing the vetted organizations ‘rcd’ identification information.

{

"@context": [

"https://www.w3.org/2018/credentials/v1",

"https://access.atis.org/apps/group\_public/dlt/v1/"

],

"id": **"https://access.atis.org/credentials/9011"**,

"type": ["VerifiableCredential", "vettedcrd"],

"holder": "did:example:<**BRAND1**>",

"issuer": "did:example:<**KYC1**>",

"issuanceDate": "2020-01-01T19:73:24Z",

"credentialSubject": {

"id": "did:example:<**BRAND1>;rcdclaims**>",

“organizationType”: “**Brand**”,

“organizationName”: “**Big Bank**”,

“organizationLogo”: "**https://vetter.org/bigbankLogo.jpg**",

“organizationJcard**”: "https://vetter.org/bigbank.json",**

“callingDataDigest”: “**A9……”**

},

"proof": { signed by KYC vetter }

}

3, The KYC vetter will reply to enterprise organization indicating the URL of the verifiable credential on the EIDL that supports their rcd’ vetting approval. In this example "https://access.atis.org/credentials/9011", the issuer KYC1, has authorized the enterprise BRAND1 ‘rcd’ information.

4, The organization will update their DID document on the EIDL with the service that indicates that they have vetted ‘rcd’ information, using the URI of the verifiable credential supporting the vetting authorization.

"@context": ["https://www.w3.org/ns/did/v1", "https://w3id.org/security/v1"],

"id": "did:example:<**BRAND1>**",

"publicKey": [{

"id": "did:example:<BRAND1>#key-1",

"type": "Ed25519VerificationKey2018",

"controller": "did:example:<BRAND1>",

"publicKeyJwk": {

“kty":"EC",

"crv":"P-256",

"x":"f83OJ3D2xF1Bg8vub9tLe1gHMzV76e8Tus9uPHvRVEU",

"y":"x\_FEzRu9m36HLN\_tue659LNpXW6pCyStikYjKIWI5a0"

}]

"service": [{

"id":"did:example:<BRAND1>;vettingstatus",

"type": "VerifiableCredentialService",

"serviceEndpoint": "https://access.atis.org/credentials/9001”

}, {

"id":"did:example:**<BRAND1>;rcdclaims**",

"type": "VerifiableCredentialService",

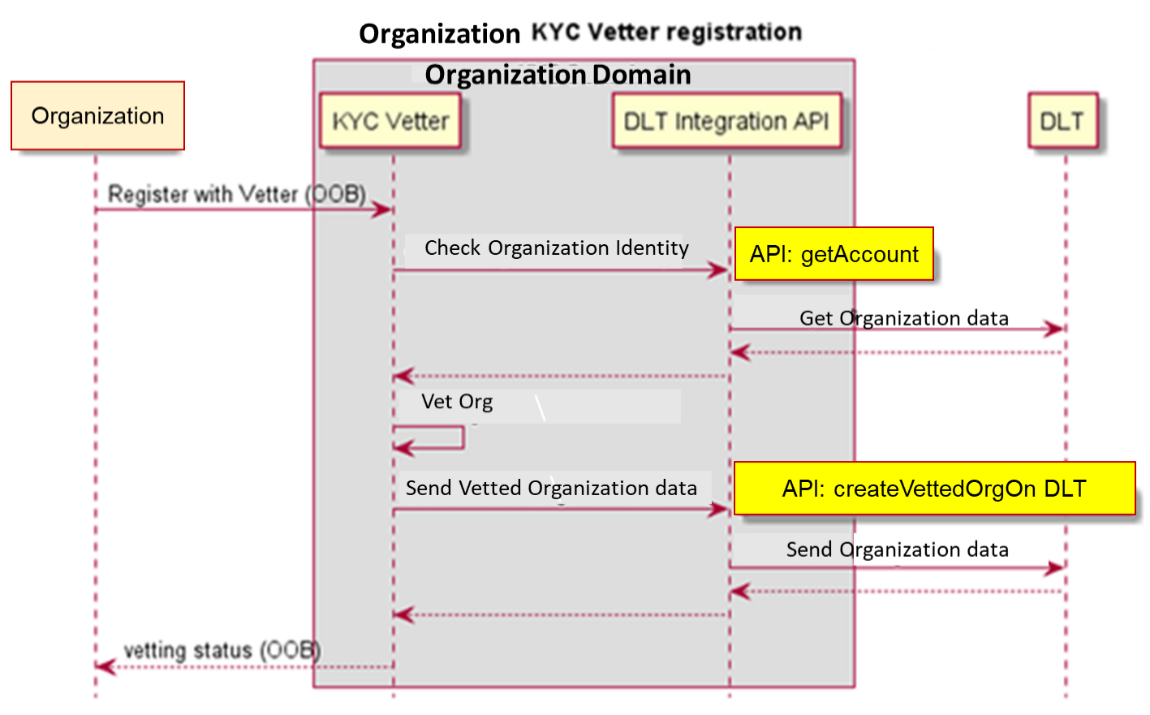
"serviceEndpoint": "**https://access.atis.org/credentials/9011”**

}]

}

### *Organization Vetting API*

Using the Enterprise Identity API definition described in Annex A of this document, organizations will request vetting from a vetter that will generate the VC claim to support the vetting status.



***Figure 5.5 API Flow for Out of Band (OOB) request to be vetted***

# TN Authorizations

All TN authorizations by actors on the EIDL network are contained within W3C verifiable credential claims. A verifiable credential is a tamper-evident credential with authorship that can be cryptographically verified proof of the TN authorization using the public key of the actors Enterprise Identity DID. When a TN unique identity is created for the first time by a TNSP, a corresponding TN Identity DID is created on the EIDLN to support the TN identity. The TN Identity DID is used to manage the control of use and chain of custody for a given TN unique identity.

**TN Authorization Verifiable Credential**

A TN authorization verifiable credential contains the following claims:

**Issuer:** The actor enterprise identity DID that creates a claim and has the authority to assign / delegate the TN. This will be the Enterprise Identity DID of either the TNSP, TNR or Brand actor

Example of the “issuer” claim: -

"issuer": "did:example:<**TNSP1**>",

**Subject:** The TN identity DID and information about the TN calling purpose as agreed with for this TN use by the Holder. The following claims are declared in the TN authorization verifiable credential: -

The “id” , which DID of the TN authorization that this verifiable credential supports.

The "TN Authorization": verifiable subject claims are included in the TN authorization Verifiable credential:

“TN” The Telephone Number authorized to be used to the **Holder** identity

“Type” The type of TN authorization by the Issuer, being Allocation, Assignment or Delegation.

“Purpose” Optionally the verifiable credential can include a call purpose representing the use of the TN authorization. The call purpose is used for verification of the calling party ‘crn’ presented in a modified “rcd” PASSporT covered in clause 7.

Example of the “credentialSubject” claim TN authorization including call purpose: -

"credentialSubject": {

"id": "did:example:<**TNIDENTITY**>",

"TN Authorization": {

"TN": "**12002001234**",

"Type": "**Delegation**",

“Purpose": “**Loan Application**"

**Holder:** The Actor enterprise Identity DID that is authorized to use the TN as a result of this assignment or delegation. This will be the Enterprise Identity DID of either the TNR, Brand or BPO actor

Example of the “holder” claim: -

"holder": "did:example:<**BRAND1**>",

**TN Decentralized Identifier**

For each unique TN identity being authenticated by the EIDL network, a unique TN identity Decentralized identifier (DID) exists to represent that unique TN identity. All TN authorization for the TN identity, is contained in this TN Identity DID. It has the chain of custody for a TN and the history of is authorized assignment and delegation.

All TN Identity DIDs for a TN identity are created on the EIDLN by TNSPs. When a TN is being assigned by a TNSP, the TNSP’s action depends on whether there already exists a TN Identity created by the TNSP for the TN. If no such TN Identity exists, the TNSP will create a TN Identity DID for the corresponding TN assigning it to the 'Issuer' DID, i.e., TNR or Brand. If such a TN Identity exists, the TNSP will update the TN Identity DID 'Issuer' DID to include the new assignment.

When removing assignment of a TN to an actor, the TNSP will update the TN Identity DID 'Issuer' DID to indicate the DID of the actor the TN is no longer assigned When a TN is no longer assigned by a TNSP to any actors, the TN Identity DID is deleted from the EIDL.

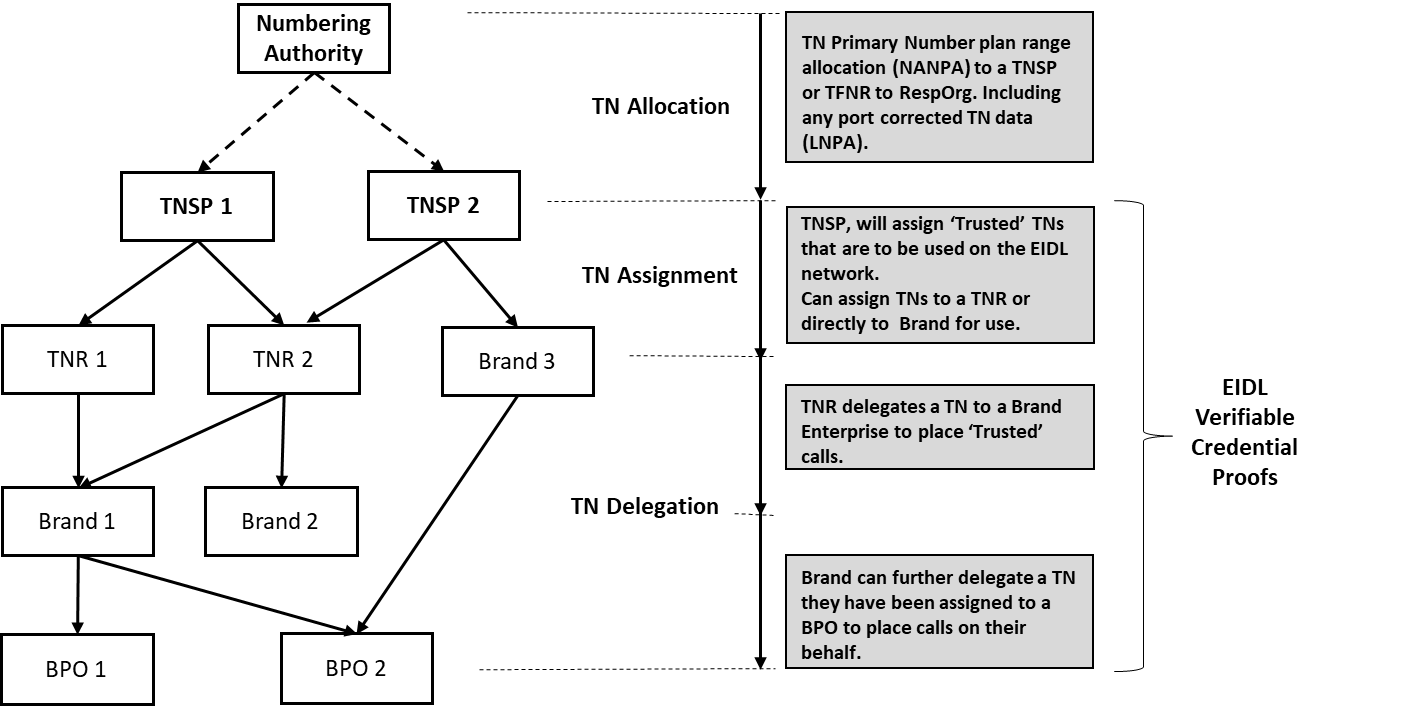
Note: A given TN may be assigned by multiple TNSPs with each TNSP creating its own TN Identity DID for the TN.

The TN Identity DID, holds the public key details of the authorized actors Enterprise Identity DID~~,~~ used when signing calls.

The following clauses show how the TN Identity DID is created at each stage of TN authorization.

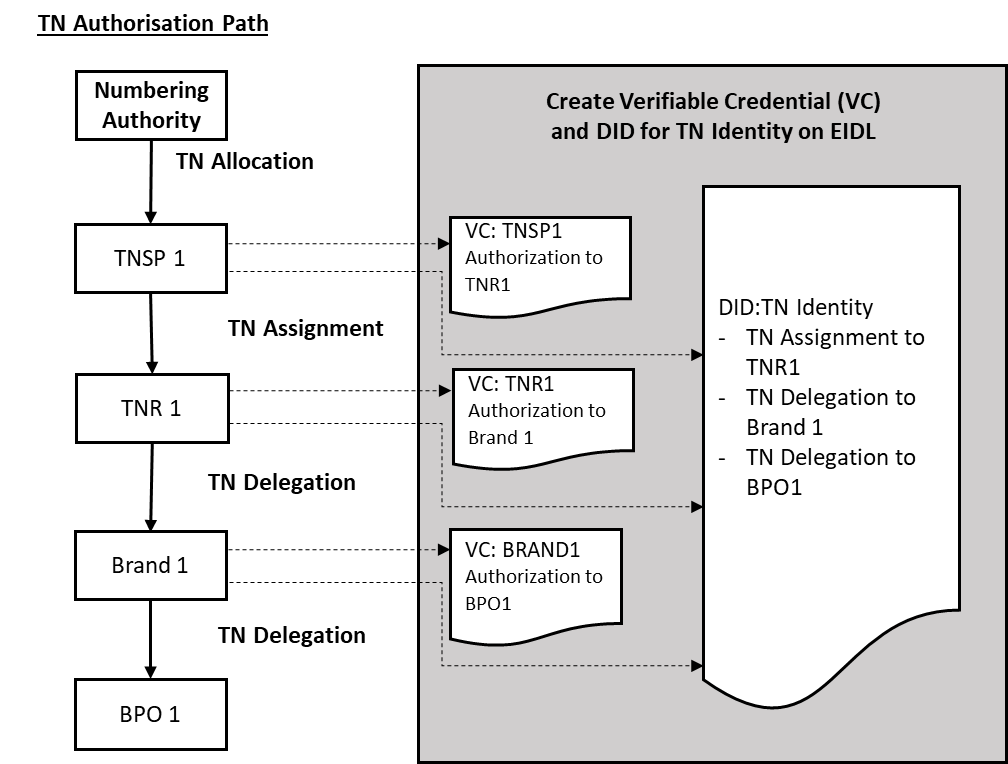
The hierarchy for which TNs are allocated from the top-level level Numbering Authority to a TNSP, assigned to a TNR, which can be delegated to a Brand Enterprise and then further delegated to a BPO, is outlined in figure 6.1. An allocation, assignment, or delegation can only be made to an actor with vetted status 'Authorized'.

Each TN authorization assignment or delegation from the TNSP to a TNR, Brand or BPO is contained within individual TN Verifiable Credential Claim and recorded to the EIDL network. This will prove that the authority for use of a given TN is assigned, or delegated to the recipient organization identity ‘Holder’**,** which is signed by the authorizing organization identity ‘Issuer’ who is assigning or delegating it to them.



***Figure 6.1 TN Authorization Hierarchy***

When an authorized actor makes a TN authorization, they will generate a corresponding verifiable credential proof of the authorized use. The first TN allocation of a TN identity will generate a TN identity DID used to verify and authenticate the TN’s use. As the TN is further assigned and delegated by actors, the DID for the TN identity is appended with the authorized verifiable credential claim. The DID for a TN identity represents the current chain of authorization for the given TN.



***Figure 6.2 TN Authorization Verifiable Credential on EIDL***

Should the TN authorization be canceled or revoked, the verifiable credential status can be updated with the reason code to indicate that the TN is no longer authorized for the actor. The TN Identity DID is updated accordingly to show the chain of authorization that currently exists for the TN. If this is from the top-level actor, i.e., number porting from one TNSP to another TNSP, the chain of custody to the new TNSP will require the creation of verifiable credential claims to support the new chain of authorization from the recipient TNSP to the assignee and delegee(s). Each authorizing actor in the chain will need to sign these verifiable claims to support the new chain of authorization. See clauses below on TN cancellation and revocation.

## Assignment of a TN by TNSP to TNR or BRAND

Assignment of a TN by a TNSP to a TNR or BRAND actor is implemented with the following steps. The following example is for a TN assignment from a TNSP to a TNR

1, A TN assignment request is made by the TNR to the TNSP as an out of band process, signing the request with their Enterprise Identity DID private key, so that the TNSP can verify the request using the Enterprise Identity DID public key

2, The TNSP will verify that the TNR requesting the TN assignment is currently of the vetting status '‘authorized/vetted’. This is done by checking the TNR Enterprise Identity DID vetting authorization Verifiable Credential claim, before allocating the TN to the TNR.

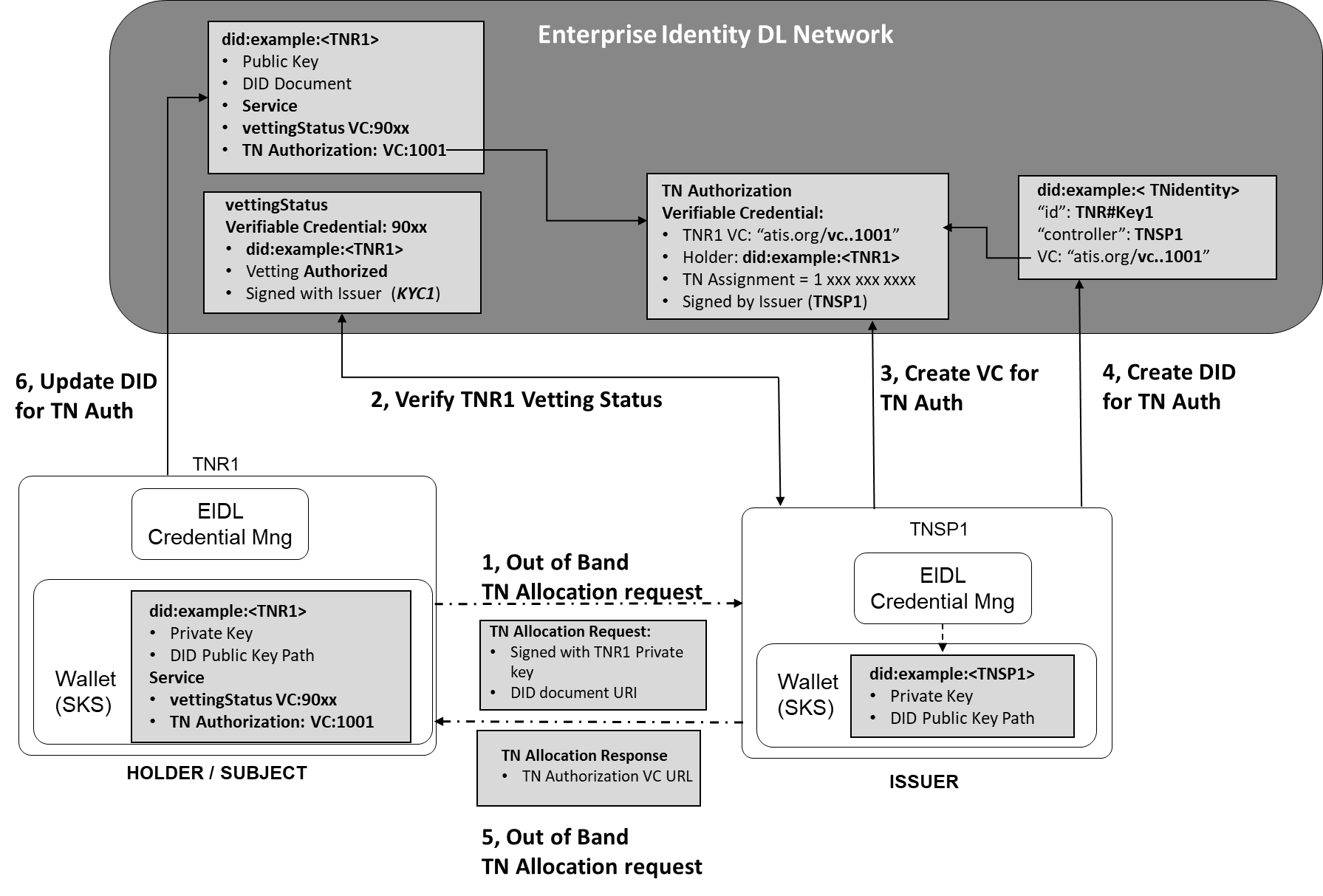
3, If the TNR is status '‘Authorized/Vetted', the TNSP will create a Verifiable Credential claim for the specific TN assignment to the TNR, signed by the TNSP, recorded on the EIDL network as proof.

4, The TNSP will then create a TN Identity DID with a reference to the specific TN Assignment Verifiable Credential to indicate that the TNR is now a holder of this TN and can sign calls with the corresponding Enterprise Identity DID private key.

If the TNR Status is not '‘Authorized/Vetted', then the TN cannot be assigned to the TNR.

5, The TNSP will respond to the TNR that the TN has been authorized to them, with the URL of the Verifiable Credential that supports this.

6, The TNR will update their Enterprise Identity DID services to include the TN Authorization Verifiable credential URL on the EIDL.



***Figure 6.4 TNSP to TNR TN Assignment – Verifiable Credential***

**TNSP TN Assignment Verifiable Credential Claim**

Example of a W3C verifiable credential claim representing the TN assignment by a TNSP to a TNR.

{

"@context": [

"https://www.w3.org/2018/credentials/v1",

"https://access.atis.org/apps/group\_public/dlt/v1/"

],

"id": **"https://access.atis.org/credentials/1001"**,

"type": ["VerifiableCredential", "TN Authorization"],

"holder": "did:example:<**TNR1**>",

"issuer": "did:example:<**TNSP1**>",

"issuanceDate": "2020-01-01T19:73:24Z",

"credentialSubject": {

"id": "did:example:<**TNIDENTITY**>",

"TN Authorization": {

"TN": "**12002001234**",

"Type": "**Assignment**",

}

},

"credentialStatus": {

"id": "https:// access.atis.org /status/24,

"type": "CredentialStatusList2017"

},

"proof": { signed by TNSP1 }

}

**Adding the TN Assignment to the TN Identity DID**

Example of a W3C DID Document used to prove the chain of custody for the TN allocation to the TNR.

{

"@context": [

"https://www.w3.org/2018/credentials/v1",

"https://access.atis.org/apps/group\_public/dlt/v1/"

],

"id": "did:example:<**TNIDENTITY**>",

“authentication" :[{

“id”:”did:example:<**TNR1#Key1**>”,

“controller":" did:example:<**TNSP1**>",

“level":" **TopRestricted**",

"publicKey": ""LMv6g…………." }],

"Service" :[{

“id”:”did:example:<**TNR1#Key1;tnAuthorization**>”,

“type": ""**TN Authorization**",

"tn":"**12002001234**",

“serviceEndpoint":**"https://access.atis.org/credentials/1001"**

}],

"proof": {...}

}

**Adding the TN Assignment Verifiable Credential to the Enterprise Identity DID**

Example of a W3C DID Document used to verify and authenticate the TN allocation to the TNR DID.

"@context": ["https://www.w3.org/ns/did/v1", "https://w3id.org/security/v1"],

"id": "did:example:<**TNR1>**",

"publicKey": [{

"id": "did:example:<TNR1>#key-1",

"type": "Ed25519VerificationKey2018",

"controller": "did:example:<**TNR1**>",

"publicKeyJwk": {

“kty":"EC",

"crv":"P-256",

"x":"f83OJ3D2xF1Bg8vub9tLe1gHMzV76e8Tus9uPHvRVEU",

"y":"x\_FEzRu9m36HLN\_tue659LNpXW6pCyStikYjKIWI5a0"

}]

"service": [{

"id":"did:example:<TNR1>;vettingstatus",

"type": "VerifiableCredentialService",

"serviceEndpoint": "https://access.atis.org/credentials/90xx”

}, {

"id":"did:example:**<TNIDENTITY>;tnAuthorization**",

"type": "VerifiableCredentialService",

"serviceEndpoint": "**https://access.atis.org/credentials/1001”**

}]

}

The same process as above is used for a TN assignment from a TNSP to a BRAND directly, where the BRAND Enterprise Identity DID is assigned for the holder within the VC claim, and the BRAND Enterprise Identity DID public key assigned to the TN identity DID.

## Delegation of a TN by TNR to Brand / Brand to BPO

Delegation of a TN by a TNR to a BRAND actor or a BRAND to a BPO actor is implemented with the following steps. The following example is from a TNR delegation of a TN to a BRAND.

1, A TN assignment request is made by the BRAND to the TNR as an out of band process, signing the request with their Enterprise Identity DID private key, so that the TNR can verify the request using the Enterprise Identity DID public key

***Note:*** The TNR can only assign a TN that is under their authorized control from a TNSP.

2, The TNR will verify that the BRAND requesting the TN assignment is currently of the vetting status '‘authorized/vetted’. This is done by checking the BRAND Enterprise Identity DID vetting authorization Verifiable Credential claim, before allocating the TN to the BRAND.

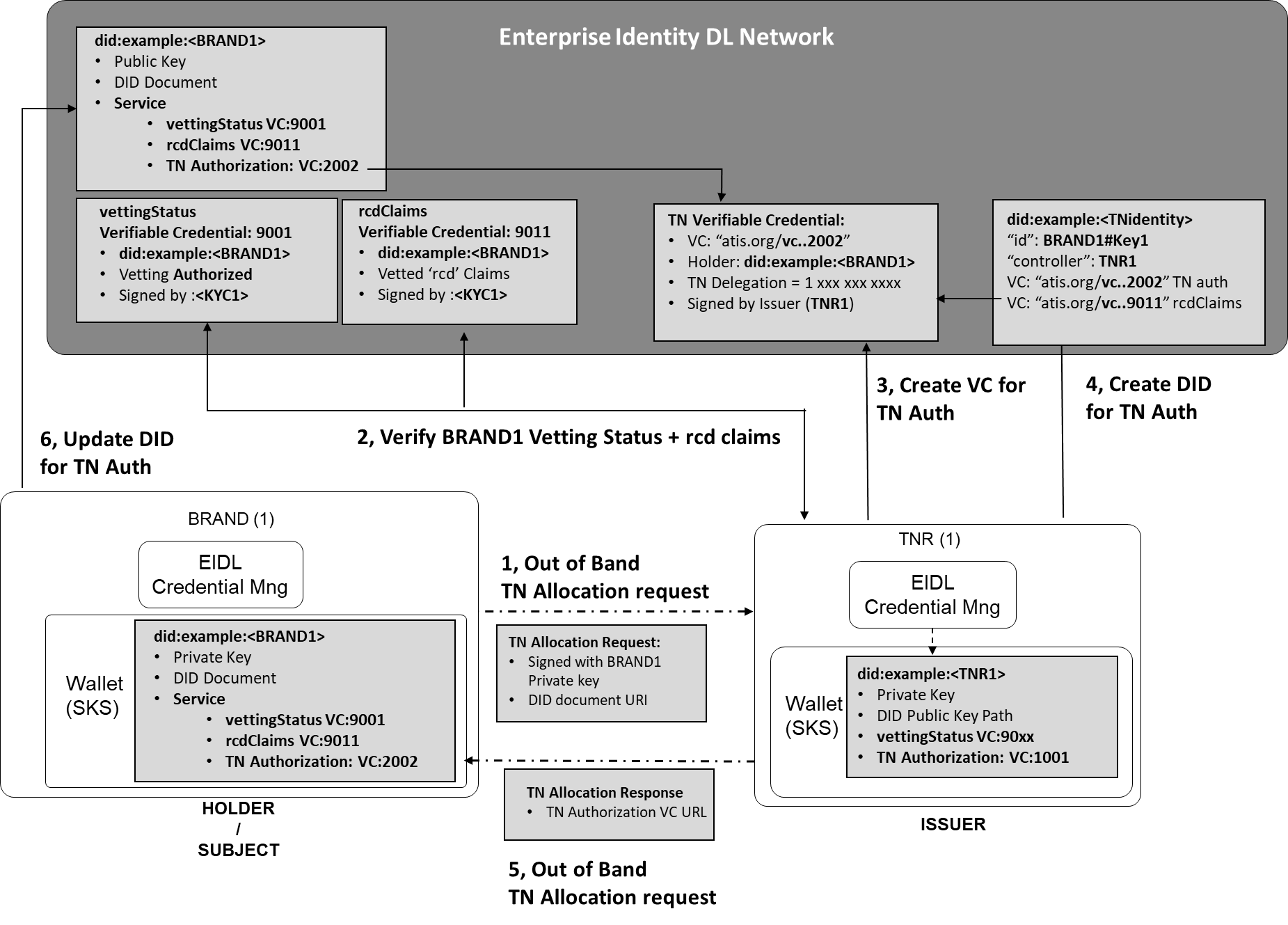
3, If the BRAND is status '‘Authorized/Vetted', the TNR will create a Verifiable Credential claim for the specific TN delegation to the BRAND, signed by the TNR, recorded on the EIDL network as proof.

4, The TNR will create a TN Identity DID update with a B reference to the specific TN Authorization Verifiable Credential which indicates that the BRAND is now a holder of this TN and can sign calls with their corresponding Enterprise Identity private key.

If the BRAND Status is not '‘Authorized/Vetted', then the TN cannot be assigned to the BRAND.

5, The TNR will respond to the BRAND that the TN has been authorized to them, with the URL of the Verifiable Credential that supports this.

6, The BRAND will update their Enterprise Identity DID services to include the TN Authorization Verifiable credential URL on the EIDL.



***Figure 6.5 TNR to BRAND TN Delegation – Verifiable Credential***

**TNR TN Delegation to a Brand Actor Verifiable Credential Claim**

Example of a W3C verifiable credential claim representing the TN delegation by a TNR to a BRAND enterprise. In the claim by the TNR to the BRAND, the claims may include the brand identity and intended reason for the call. It is possible to include within the verifiable claim other constraints about the use of the TN. In the example below the TN is allocated for the intended purpose of making calls for *‘Loan Application’*. The “Purpose” claims can be verified against the “rcd” PASSporT claim ‘crn’.

{

"@context": [

"https://www.w3.org/2018/credentials/v1",

<https://access.atis.org/apps/group_public/dlt/v1/>],

"id": **"https://access.atis.org/credentials/2002"**,

"type": ["VerifiableCredential", "TN Authorization"],

"holder": "did:example:<**BRAND1**>",

"issuer": "did:example:<**TNR1**>",

"issuanceDate": "2020-01-01T19:73:24Z",

"credentialSubject": {

"id": "did:example:<**TNIDENTITY**>",

"TN Authorization": {

"TN": "**12002001234**",

"Type": "**Delegation**",

“Purpose": “**Loan Application**"}},

"credentialStatus": {

"id": "https:// access.atis.org /status/24",

"type": "CredentialStatusList2017"},

"proof": { signed by TNR1 }

}

**Adding the TN Delegation from the TNR to the BRAND in the TN Identity DID**

Example of a W3C DID Document used to prove the chain of custody for the TN allocation to the BRAND.

{

"@context": [

"https://www.w3.org/2018/credentials/v1",

"https://access.atis.org/apps/group\_public/dlt/v1/"

],

"id": "did:example:<**TNIDENTITY**>",

“authentication" :[{

“id”:”did:example:<**TNR1#Key1**>”,

“controller":" did:example:<**TNSP1**>",

“level":" **TopRestricted**",

"publicKey": "LMv6g…………."

},{

“id”:”did:example:<**BRAND1#Key1**>”,

“controller”:”did:example:<**TNR1**>”,

“level":"**TopRestricted**",

"publicKey": "LMv6g…………."

}],

"Service" :[{

“id”:”did:example:<**TNR1#Key1;tnAuthoirization** >”,

“type": ""**TN Authorization**",

"tn":"**12002001234**",

“serviceEndpoint":**" https://access.atis.org/credentials/1001"**

},{

“id”:”did:example:<**BRAND1#Key1;tnAuthoirization**>”,

“type”: "**TN Authorization**",

"tn":"**12002001234**",

“serviceEndpoint”:**”https://access.atis.org/credentials/2002”**

}],

"proof": {...}

}

**Adding the TN Assignment Verifiable Credential to the Enterprise Identity DID**

Example of a W3C DID Document used to verify and authenticate the TN allocation to the BRAND DID.

"@context": ["https://www.w3.org/ns/did/v1", "https://w3id.org/security/v1"],

"id": "did:example:<**BRAND1>**",

"publicKey": [{

"id": "did:example:<BRAND1>#key-1",

"type": "Ed25519VerificationKey2018",

"controller": "did:example:<**BRAND1**>",

"publicKeyJwk": {

“kty":"EC",

"crv":"P-256",

"x":"f83OJ3D2xF1Bg8vub9tLe1gHMzV76e8Tus9uPHvRVEU",

"y":"x\_FEzRu9m36HLN\_tue659LNpXW6pCyStikYjKIWI5a0"

}]

"service": [{

"id":"did:example:<BRAND1>;vettingstatus",

"type": "VerifiableCredentialService",

"serviceEndpoint": "https://access.atis.org/credentials/9001”

}, {

"id":"did:example:**<BRAND1>;rcdclaims**",

"type": "VerifiableCredentialService",

"serviceEndpoint": "**https://access.atis.org/credentials/9011”**

}, {

"id":"did:example:**<TNIDENTITY>;tnAuthorization**",

"type": "VerifiableCredentialService",

"serviceEndpoint": "**https://access.atis.org/credentials/2002”**

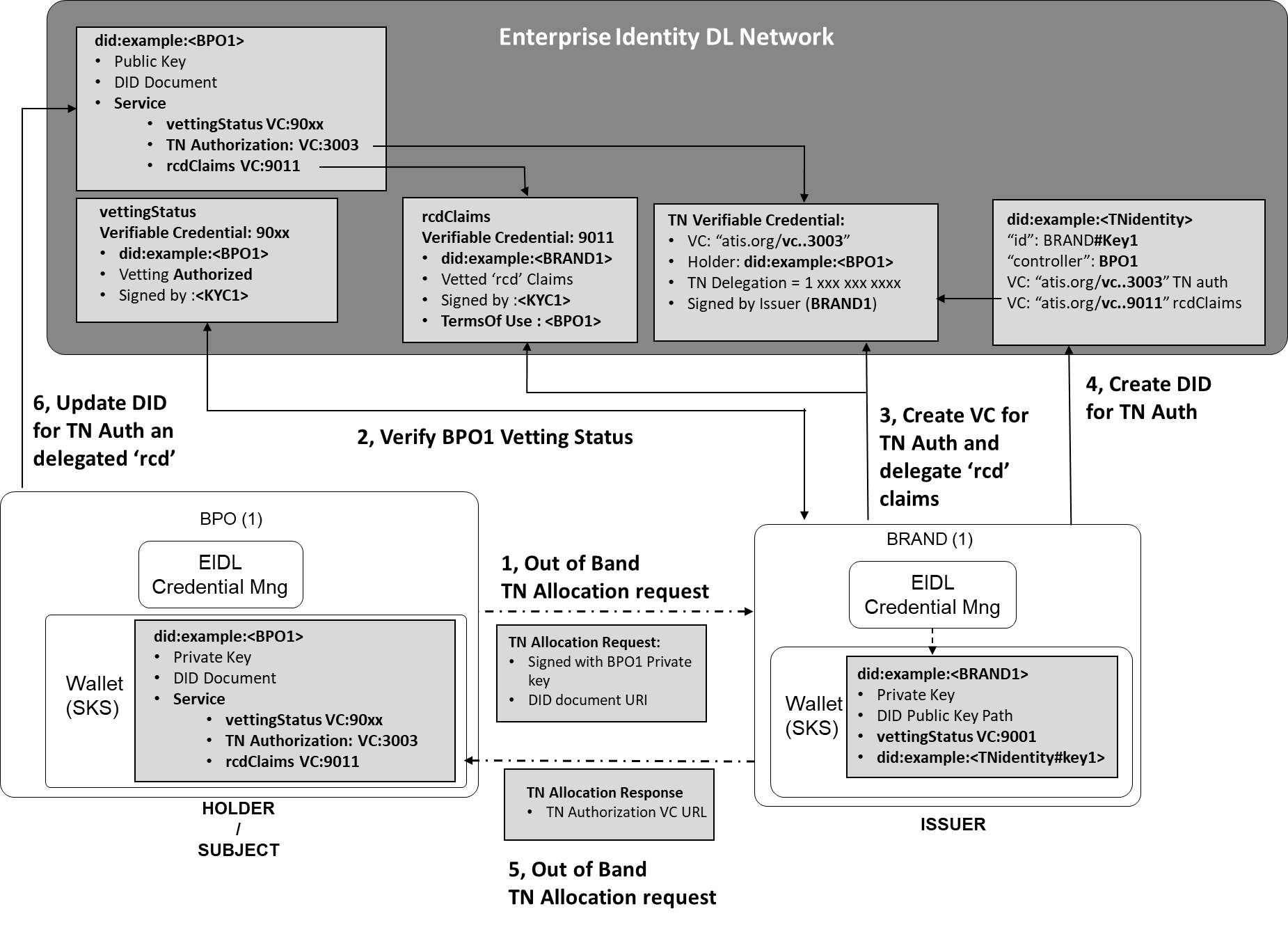
}]

}

### Delegation of a TN by Brand to BPO

The same process as described in figure 6.5 above is used for a TN delegation from a BRAND to a BPO directly, where the BPO Enterprise Identity DID is assigned for the ‘holder’ within the VC claim, and the BPO Enterprise Identity DID public key assigned to the TN identity DID.

Specifically when a BRAND delegates a TN authorization of use to a BPO , where the call is being placed by the BPO on behalf of the BRAND and the BPO will send enhanced call data with the signed call , to indicate that the call is from the BRAND. The Verifiable Credential for the TN delegation by the BRAND to the BPO contains a ‘termsOfUse’ reference to allow the BRAND rcdClaim information to be delegated to the BPO. This allows the rcdClamins for the BRAND to be authenticated with the calls are placed by the BPO and signed by the BPO for this TN.



***Figure 6.6 BRAND to BPO TN Delegation***

**TNR TN Delegation to a Brand Actor Verifiable Credential Claim**

Example of a W3C verifiable credential claim representing the TN delegation by a BRAND to a BPO. In the claim by the BRAND to the BPO, the claims may include the brand identity and intended reason for the call. It is possible to include within the verifiable claim other constraints about the use of the TN. In the example below the TN is still allocated by the BRAND for the intended purpose of the BPO making calls for *‘Loan Application’* on behalf of the BRAND. The “Purpose” claims can be verified against the “rcd” PASSporT claim ‘crn’

{

"@context": [

"https://www.w3.org/2018/credentials/v1",

<https://access.atis.org/apps/group_public/dlt/v1/> ],

"id": **"https://access.atis.org/credentials/3003"**,

"type": ["VerifiableCredential", "TN Authorization"],

"holder": "did:example:<**BPO1**>",

"issuer": "did:example:<**BRAND1**>",

"issuanceDate": "2020-01-01T19:73:24Z",

"credentialSubject": {

"id": "did:example:<**TNIDENTITY**>",

"TN Authorization": {

"TN": "**12002001234**",

"Type": "**Delegation**",

“Purpose": “**Loan Application**"

}},

"credentialStatus": {

"id": "https:// access.atis.org /status/24",

"type": "CredentialStatusList2017"},

"proof": { signed by BRAND1 }

}

**Adding the TN Delegation to the TN Identity DID**

Example of a W3C DID Document used to verify and authenticate the TN allocation to the BRAND.

{

"@context": [

"https://www.w3.org/2018/credentials/v1",

"https://access.atis.org/apps/group\_public/dlt/v1/"

],

"id": "did:example:<**TNIDENTITY**>",

“authentication" :[{

“id”:”did:example:<**TNSP#Key1**>”,

“controller":" did:example:<**TNR1**>",

“level":" **TopRestricted**",

"publicKey": "LMv6g…………."

},{

“id”:”did:example:<**TN`R#Key1**>”,

“controller”:”did:example:<**BRAND1**>”,

“level":"**TopRestricted**",

"publicKey": "LMv6g…………."

},{

“id”:”did:example:<**BRAND1Y#Key1**>”,

“controller”:”did:example:<**BPO1**>”,

“level":"**TopRestricted**",

"publicKey

}],

"Service" :[{

“id”:”did:example:< **TNR1#key1;tnAuthoirization** >”,

“type": ""**TN Authorization**",

"tn":"**12002001234**",

“serviceEndpoint":**" https://access.atis.org/credentials/1001"**

},{

“id”:”did:example:< **BRAND1#key1;tnAuthoirization** >”,

“type”: "**TN Authorization**",

"tn":"**12002001234**",

“serviceEndpoint”:[**https://access.atis.org/credentials/2002**](https://access.atis.org/credentials/2002)

}, {

“id”:”did:example:< **BPO1#key1;tnAuthoirization** >”,

“type”: "**TN Authorization**",

"tn":"**12002001234**",

“serviceEndpoint”:**”https://access.atis.org/credentials/3003”**

}],

"proof": {...}

}

**Adding the delegated use of the BRAND ‘rcd’ Verifiable Credential information to the BPO.**

The BRAND can add a ‘termsOfUse’ statement in their ‘vettedrcd’ Verifiable Credential allows the ‘rcd’ information that has been vetted for the BRAND to be delegated to the BPO. The terms of use will allow the ‘assigner ‘=BRAND to the ‘assignee’ = BPO the use of the ‘rcd’ information strictly for the TN authorization as indicated in the ’action’.

Example of a W3C Verifiable Credential used to assign delegated terms of use for the BRANDS ‘rcd’ information to the BPO.

"@context": [

"https://www.w3.org/2018/credentials/v1",

"https://access.atis.org/apps/group\_public/dlt/v1/"

],

"id": **"https://access.atis.org/credentials/9011"**,

"type": ["VerifiableCredential", "vettedcrd"],

"holder": "did:example:<**BRAND1**>",

"issuer": "did:example:<**KYC1**>",

"issuanceDate": "2020-01-01T19:73:24Z",

"credentialSubject": {

"id": "did:example:<**BRAND1>;rcdclaims**>",

“organizationType”: “**Brand**”,

“organizationName”: “**Big Bank**”,

“organizationLogo”: "**https://vetter.org/bigbankLogo.jpg**",

“organizationJcard**”: "https://vetter.org/bigbank.json",**

“callingDataDigest”: “**A9……”**

},

"proof": { signed by KYC vetter }

}

}],

**"termsOfUse": [{**

"type": "IssuerPolicy",

"id": **"https://access.atis.org/credentials/9033"**,

"prohibition": [{

"assigner": "did:example:<**BPO1**>",

"assignee": "did:example:**<BRAND1>”;**

"target": "**http://example.edu/credentials/9011**", ( VC of ‘rcd’ Claims )

"action": [**https://access.atis.org/credentials/3003**](https://access.atis.org/credentials/3003)( VC of TN Delegation**)**

},

"proof": [ ... ]

}

**Adding the TN Assignment and delegated ‘rcd’ Verifiable Credentials to the BPO Enterprise Identity DID**

Example of a W3C DID Document used to verify and authenticate the TN allocation to the BRAND DID.

"@context": ["https://www.w3.org/ns/did/v1", "https://w3id.org/security/v1"],

"id": "did:example:<**BPO1>**",

"publicKey": [{

"id": "did:example:<BRAND1>#key-1",

"type": "Ed25519VerificationKey2018",

"controller": "did:example:<**BPO1**>",

"publicKeyJwk": {

“kty":"EC",

"crv":"P-256",

"x":"f83OJ3D2xF1Bg8vub9tLe1gHMzV76e8Tus9uPHvRVEU",

"y":"x\_FEzRu9m36HLN\_tue659LNpXW6pCyStikYjKIWI5a0"

}]

"service": [{

"id":"did:example:<BPO1>;vettingstatus",

"type": "VerifiableCredentialService",

"serviceEndpoint": "https://access.atis.org/credentials/90xx”

}, {

"id":"did:example:**<TNIDENTITY>;tnAuthorization**",

"type": "VerifiableCredentialService",

"serviceEndpoint": [**https://access.atis.org/credentials/3003**](https://access.atis.org/credentials/3003)

}, {

"id":"did:example:**<BRAND1>;rcdclaims**", ( delegated by BRAND for TN )

"type": "VerifiableCredentialService",

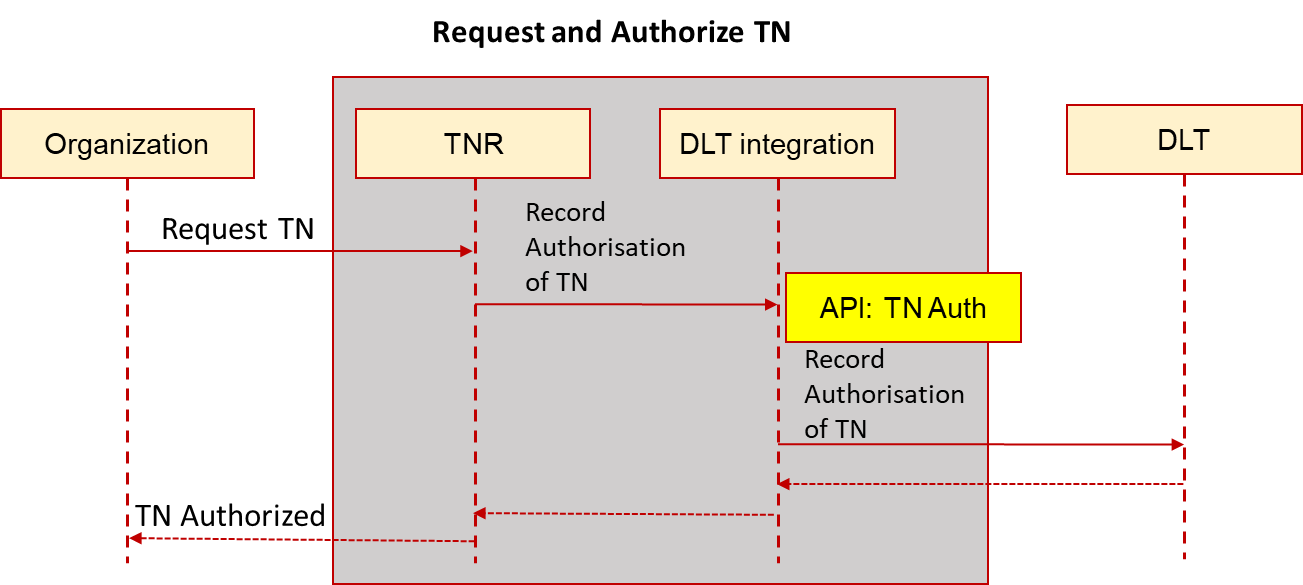
"serviceEndpoint": "**https://access.atis.org/credentials/9011”**

}]

}

### TN Authorization API

Using the Enterprise Identity API definition described in Annex A of this document, an organization will request the TN authorization from an assignee organization that will generate the verifiable credential claim to support the TN allocation / assignment / delegation on the EIDL.



***Figure 6.7 API Flow for TN Authorization***

## TN Cancel and Revocation - Credential Claim Status

When a TN allocation, assignment, or delegation credential needs to be canceled or revoked. A VC issuer can update the 'credentialStatus' property for the discovery of information about a verifiable credential. The ‘credentialStatus’ property for the discovery of information about a verifiable credential can be updated by a VC issuer.

The following example of the credential claim status file is for a TN being ported from a TNSP1 and a canceled number from being used by TNR1.

{

"id": "https://access.atis.org/status/24,

""description": ""Status of TN Allocation credentials."

"verifiableCredential": [{

"claim": {

"id": "**https://access.atis.org/credentials/1001**>",

"currentStatus": "**Revoked**",

"statusReason": "**Number Port**"

},

"issuer": " did:example:<**TNSP1**>",

"issued": "2020-02-04T14:27:42Z",

"proof": { ... }

}, {

"claim": {

"id": "**https://access.atis.org/credentials/2002**>",

"currentStatus": "**Revoked**",

"statusReason": "**Canceled**"

},

"issuer": " did:example:<**TNR1**>",

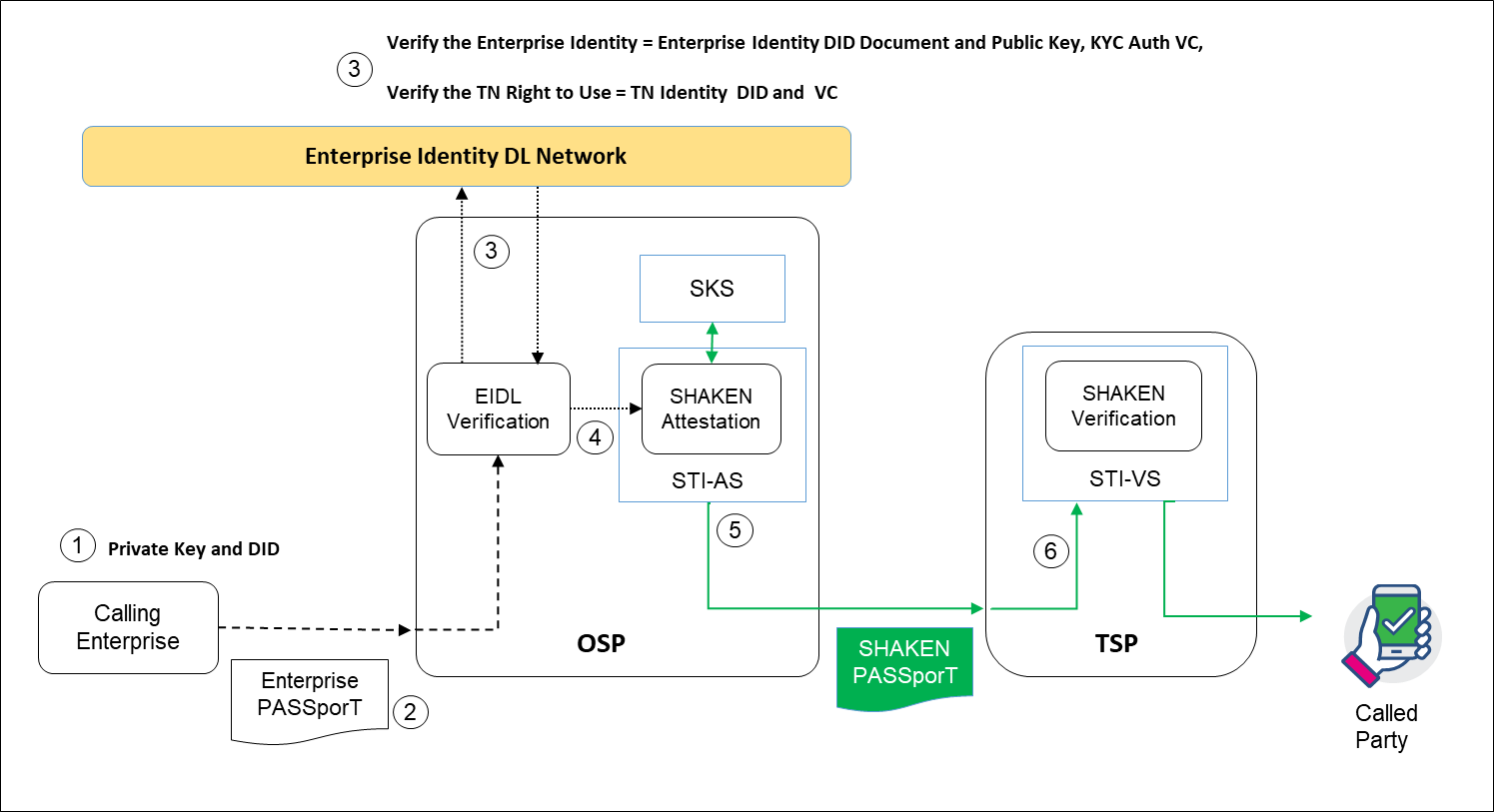
"issued": "2020-02-04T14:27:42Z",

"proof": { ... }

}]

# OSP Attestation of a call using EIDL Network

This clause describes how the EIDL network integrates with the STIR/SHAKEN Architecture to enable an OSP to use the Enterprise Identity to perform verification and authentication of the calling organization identity and proof that they have the authority to place calls with the calling TN.



***Figure 7.1 Enterprise Identity from EIDL Network to provide OSP A Attestation***

The call scenario in figure 7.1 illustrates where a KYC vetted Enterprise originates a call to an OSP, including an ‘PASSporT token in a SIP Identity header. In this scenario, the OSP is able to use the information included in the ‘Enterprise’ PASSporT token to qualify the caller identity and authorization to place this call from the calling TN.

1. The calling Enterprise will use the Private Key associated to their KYC vetted Enterprise Identity DID to generate the ‘Enterprise’ PASSporT for the call being placed. The ‘Enterprise’ PASSporT header will include the URI of the Enterprise Identity DID on the EIDL, and a signed claim within the ‘Enterprise’ PASSporT payload of the TN Identity DID that indicates the authorization for the calling Enterprise Identity to use the TN it is placing the call with.
2. The ‘Enterprise’ PASSporT token is included in the SIP Invite sent to the OSP.
3. The OSP EIDL verification service will verify the ‘Enterprise’ PASSporT token from the EIDL Network to authenticate the calling entity Enterprise identity and check the authorized use of the calling TN.
   1. Using the URI of the Enterprise Identity DID from the ‘Enterprise’ PASSporT header, retrieve the Public Key from the EIDL to verify the token.
   2. Check the Enterprise Identity has been KYC vetted from the vetting verifiable credential, that the current status is ‘authorized’
   3. Check that the Enterprise Identity DID, contains a verifiable credential authoring the use of the TN by this DID, for the TN indicated in the SIP Calling number and the ‘orig’ claim in the PASSporT payload.
4. Based on the verification of the ‘Enterprise’ PASSporT for the signed call.
   1. If the 'Enterprise’ PASSporT verification passes, the OSP authentication service may, based on local policy, interpret this verification result as establishing that the entity populating the PASSporT has a known authenticated identity and an association with the calling TN. Armed with this attestation criteria information, the OSP shall perform the SHAKEN authentication procedures defined in ATIS-1000074 and may assert an attestation level of Full or "A" attestation. The OSP shall sign the "shaken" PASSporT with SHAKEN certificate credentials tied to its SPC.
   2. If the 'Enterprise’ PASSporT verification fails, the OSP authentication service should ignore this as input to determine the attestation level of a generated "shaken" PASSporT and follow the standard procedures of ATIS-1000074 for determining attestation level as described in [ATIS-1000074] and based on local policy.
5. The TSP will verify the “shaken” PASSporT according to the SHAKEN Specification [ATIS-1000074].

## Enterprise PASSporT Encoding

The ‘Enterprise PASSporT’ token will be encoded using either a modified base PASSporT, as shown below, or by using a modified “rcd” PASSporT with the same modifications. The proposed modification to the base PASSporT used the reference type for the JWT authentication credentials ("jku" versus "x5u") to provide the URI of the Enterprise identity DID on the EIDL.,

### Use of a modified base PASSporT encoding for Enterprise Identity

This is an example of a modified base PASSporT encoded for Enterprise Identity and TN authorization.

Protected Header

{

"typ":"passport",

"alg":"ES256",

**“jku":”did:example:098765432abcdefghi”**

}

Payload

{

"dest":{"tn":"12125551213 "},

"iat":1443208345,

"orig":"tn":"12155551212"},

}

Description of the header fields and payload claims: -

The “jku" indicates the URI of the DID of the of the caller identity. Use "jku" URI format is defined in RFC 7515 sec 4.1.2.

***Note:*** it is suggested RFC 8225 be updated to include the option to 4.3 to indicate that “jku” Header Parameter to reference the Enterprise Identity DID URI for the public key used to digitally sign the JWS of the PASSporT.

The 'orig' claim and 'dest' claim shall be of type URI or 'tn'.

The "iat" should be set to the date and time of issuance

### Use of a modified “rcd” PASSporT encoding for Enterprise Identity

The ‘rcd’ PASSporT is defined in ATIS-XXX Signature-based Handling of Asserted information using toKENs (SHAKEN): Calling Name and Rich Call Data Handling Procedures. The proposed modification to the ‘rcd’ PASSport used the reference type for the JWT authentication credentials ("jku" versus "x5u") to provide the URI of the Enterprise identity DID on the EIDL.

This is an example of an "rcd" extension PASSporT with modifications to resolve the Enterprise Identity and TN authorization, with “rcd” claims for enhanced calling data and call reason:

Protected Header

{

"alg":"ES256",

"typ":"passport",

“ppt”:”rcd”,

**“jku":”did:example:098765432abcdefghi”**

}

Payload

{

"dest":{“tn”:"12155551213"}

"iat":1443208345,

"orig":{“tn”:"12155551212"},

"rcd":{"nam":"Big Bank","jcd":["vcard",[["logo",{},"uri",

"https://vetter.org/bigbankLogo.jpg"]]]}}

"rcdi":<computed per draft-ietf-stir-passport-rcd>

"crn":"Loan Application"

}

Description of the header fields and payload claims:-

The “jku” indicates the URI of the DID of the of the caller identity. Use "jku" URI format is defined in RFC 7515 sec 4.1.2

***Note:*** it is suggested RFC 8225 be updated to include the option to 4.3 to indicate that “jku” Header Parameter to reference the Enterprise Identity DID URI for the public key used to digitally sign the JWS of the PASSporT.

The 'orig' claim and 'dest' claim shall be of type URI or 'tn'.

The ""iat"" should be set to the date and time of issuance.

***Note:*** [draft-ietf-stir-passport-rcd] defines three rcd PASSporT claims; the "rcd", “crn", and "rcdi" claims. There are two main key values possible as part of the "rcd" claim used for the Enterprise PASSporT. They are; (1) "nam" which is a minimally required key value as part of the "rcd" claim value JSON object and (2) the “crn” claim which contains a call reason phrase that describes the intent of the call. It is optional but recommended for enhancing usefulness to call recipients.

The "rcd" claim containing the JSON object of the enhanced calling data. The “rcd” claim can contain the following key values:

* A "nam" key with a value that identifies the display name of the originating entity. If the originating entity does not have a display name, the "nam" key value shall be the empty string.
* A "jcd" key value for an "rcd" claim should be constructed with the value being equal to a jCard string. Note that additional objects are optional but may be ignored or disregarded by the receiving entity depending on the rendering capabilities of the device and/or network local policy.
* A "jcl" key value for an "rcd" claim should be constructed with the value being equal to an HTTPS URL of a file hosted on an HTTPS server containing a jCard string. Note that additional objects are optional but may be ignored or disregarded by the receiving entity depending on the rendering capabilities of the device and/or network local policy.

The “rcdi” claim protects the contents of resources referenced by the "rcd" claim from being inadvertently or maliciously modified to unauthorized values.

The "crn" claim to convey the reason for the call. The contents of the "rcd" claim have no bearing on the inclusion or value of the “crn" claim.

## ‘Enterprise’ PASSporT Verification Procedure

### OSP Verification using a modified base PASSporT

When the Brand/BPO originates a SIP call to an OSP (directly or indirectly), they encode the Enterprise PASSporT using the modified base PASSporT format included in the SIP identity header as defined in clause 7 of this document. The OSP decision to process the Enterprise PASSporT received in originating INVITE requests is based entirely on local policy; i.e., the OSP can apply a policy to perform these functions always, selectively based on some criteria, or never.

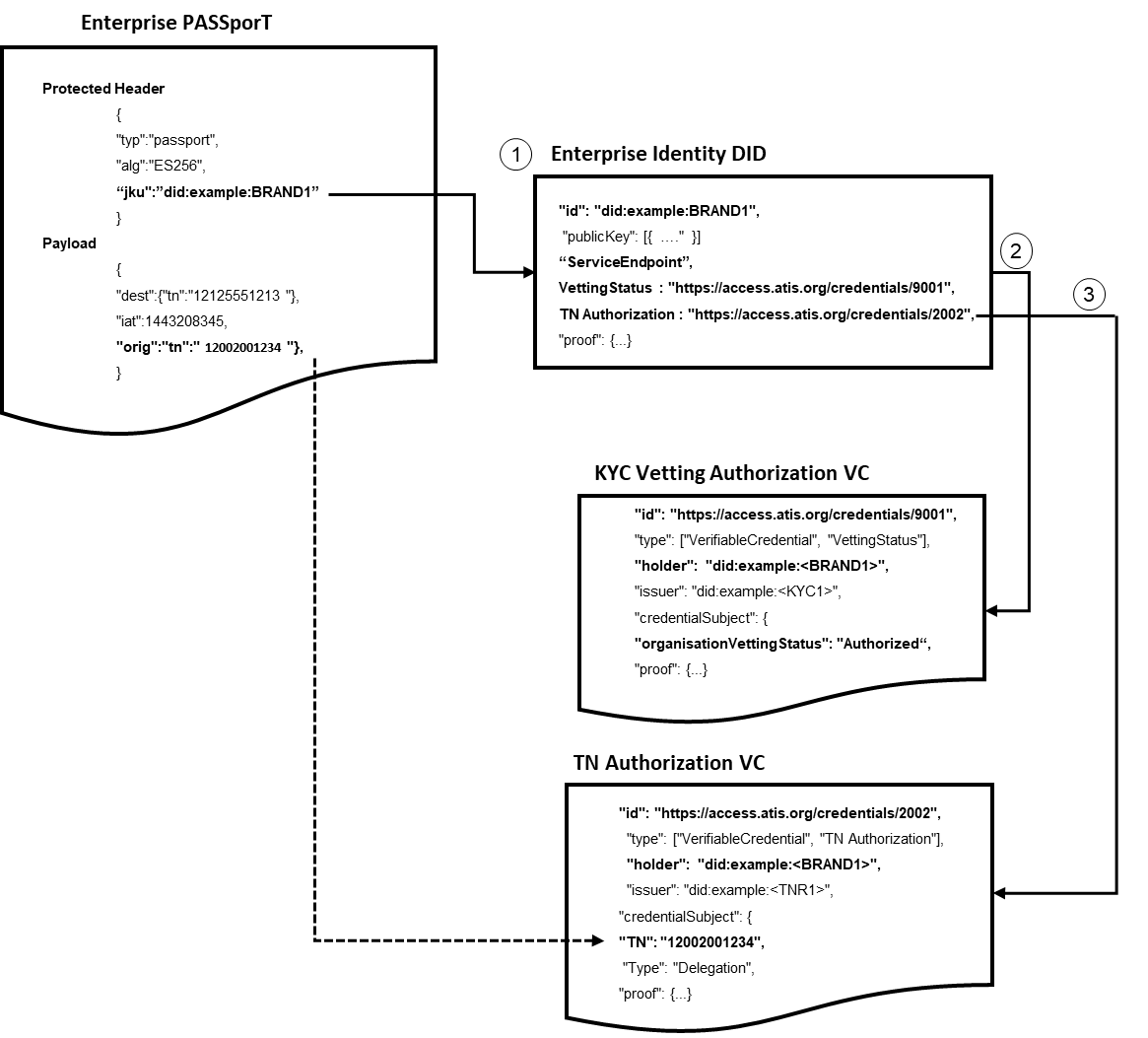
If local policy dictates that the OSP accepts the received Enterprise PASSporT, then it shall verify the Enterprise PASSporT from the EIDL as shown in figure 7.2.

1, From the “jku” claim in the header, retrieve the Enterprise Identity DID Public Key verify the PASSporT token.

2, Verify the Enterprise Identity DID has vetting status “Authorized” from the vetting Status verifiable credential.

3,.

Verify that the Enterprise Identity DID has the authorized right to use this TN from the TN Authorization Verifiable Credential.



***Figure 7.2 OSP verification of an originating caller using a modified base PASSporT***

### OSP Verification using a modified “rcd” PASSporT

When the Brand/BPO originates a SIP call to an OSP (directly or indirectly), they encode the Enterprise PASSporT using the modified “rcd” PASSporT format included in the SIP identity header as defined in clause 7 of this document. The OSP decision to process the Enterprise PASSporT received in originating INVITE requests is based entirely on local policy; i.e., the OSP can apply a policy to perform these functions always, selectively based on some criteria, or never.

If local policy dictates that the OSP accepts the received Enterprise PASSporT, then it shall verify the Enterprise PASSporT from the EIDL as shown in figure 7.3.

1, From the “jku” claim in the header, retrieve the Enterprise Identity DID Public Key verify the PASSporT token.

2, Verify the Enterprise Identity DID has a vetting status “Authorized” from the vetting Status verifiable credential.

3,.

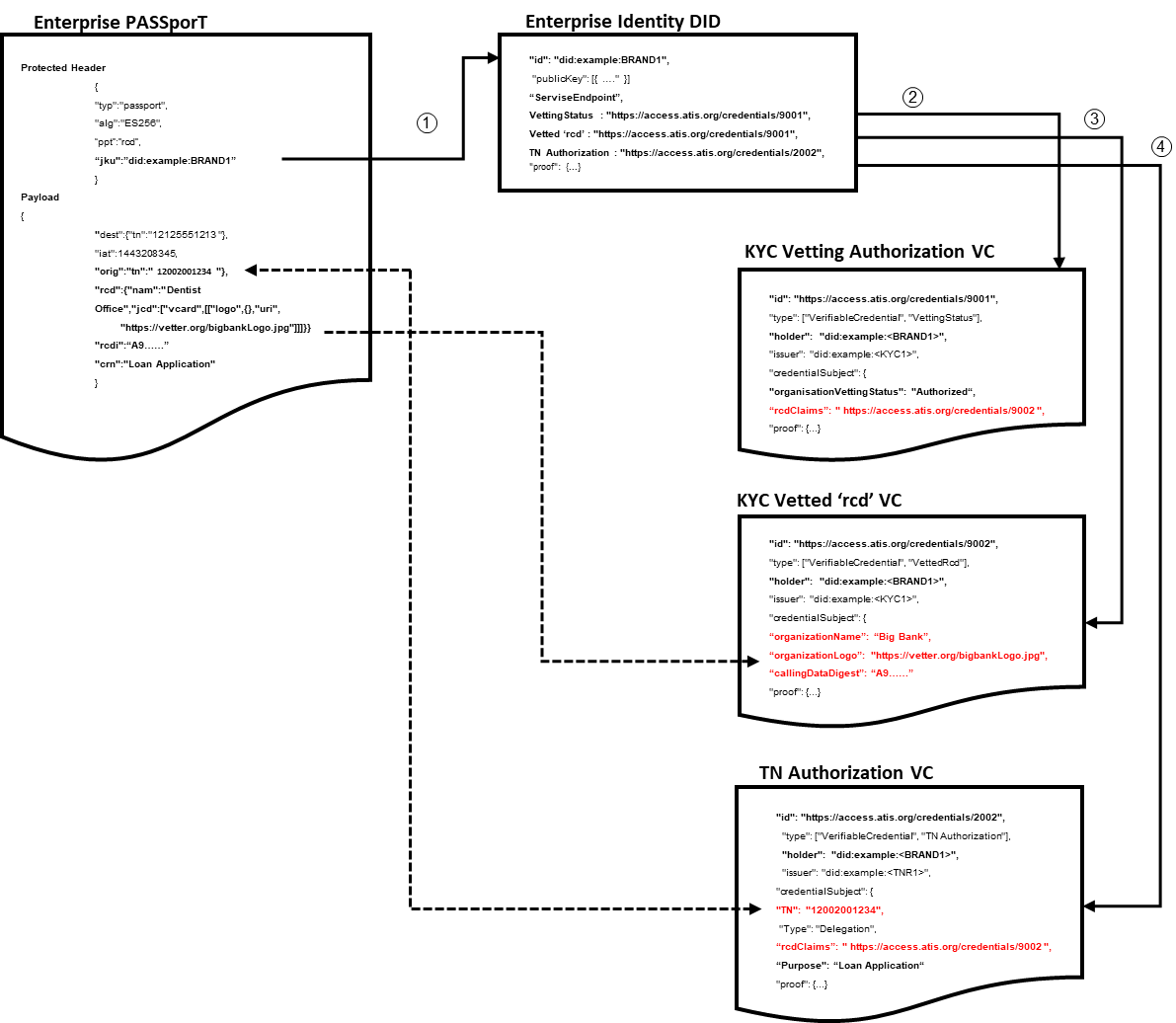
Verify that the Enterprise Identity DID has the authorized right to use this TN from the TN Authorization Verifiable Credential. .

4, Where the Enterprise PASSporT is using a modified “rcd” PASSporT format. The Enterprise PASSporT may include “rcd” claims that can be used to verify the caller name and purpose of the call against the corresponds TN identity VC claim used in point 4 of this claim.

The OSP can verify and compare the presented ‘rcd’ claims in the modified ‘rcd’ PASSporT with the EIDL values from the ’rcd’ Verifiable Credential claims: -

* The rcdClaimVC claim ‘organizationName’ = the "nam" value for “rcd”
* The rcdClaim VC claim “organizationLogo”: = the ‘jcl’ value for ”rcd”
* The rcdClaimVC claim “organizationJcard” = the ‘jcd’ value for “rcd”
* The rcdClaim VC claim “callingDataDigest” = the ‘rcdi’ value for “rcd”
* The TN identity VC claim ‘Purpose’ = the ‘crn" value for “rcd”

Based on local policy the OSP will add these "rcd" claims to a "shaken" PASSporT. the EIDL verification will convey to the SHAKEN authentication service to populate the base "shaken" claims as specified in [ATIS-1000074]. The shaken authentication service shall add "rcd" claims to a "shaken" PASSporT only if the criteria for "A" attestation are met.



***Figure 7.3 OSP verification of an originating caller using a modified “rcd” PASSporT***

# Traceback using Enterprise Identity

The United States Congress approved TRACED Act of 2019, directs the FCC to facilitate industry’s effort to “trace back” the origin of unlawful robocalls by establishing a consortium that would coordinate these private-led traceback efforts. These efforts are intended to uncover the true origin of unlawful robocalls, the source of which often is masked due to spoofed caller ID information.

Traceback is the process of determining the origin of a call, typically by starting with the receiving party and terminating ~~voice~~ service provider (TSP) and tracing backwards through the path of the intermediate providers and, ultimately, to the originating voice service provider and the origin of the call. Traceback can be used to find the source of robocalls and, thus, the entities responsible for those calls.

Two key principals as it relates to traceback that can use the Enterprise Identity DLT Network are:-

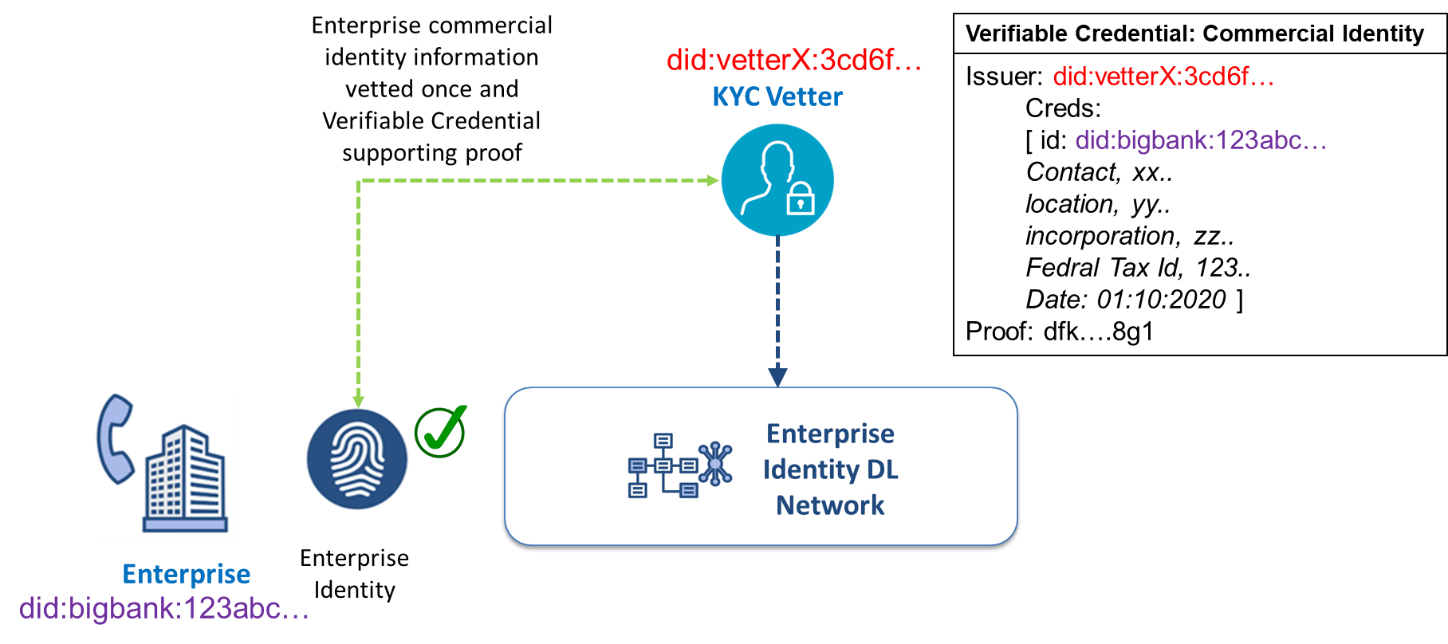
* **Principle #5**. **Confirm the Identity of Commercial Customers.** Confirm the identity of new commercial VoIP customers by collecting information such as physical business location, contact person(s), state or country of incorporation, federal tax ID, and the nature of the customer’s business.
* **Principle #6. Require Traceback Cooperation in Contracts.** For all new and renegotiated contracts governing the transport of voice calls, use best efforts to require cooperation in traceback investigations by identifying the upstream provider from which the suspected illegal robocall entered its network or by identifying its own customer if the call originated in its network.

## Principle #5. Confirm the Identity of Commercial Customers.

A business entity can have their commercial identity data verified and contained within a verifiable credential claim from the KYC vetting process that they can use with other businesses on the EIDLN.

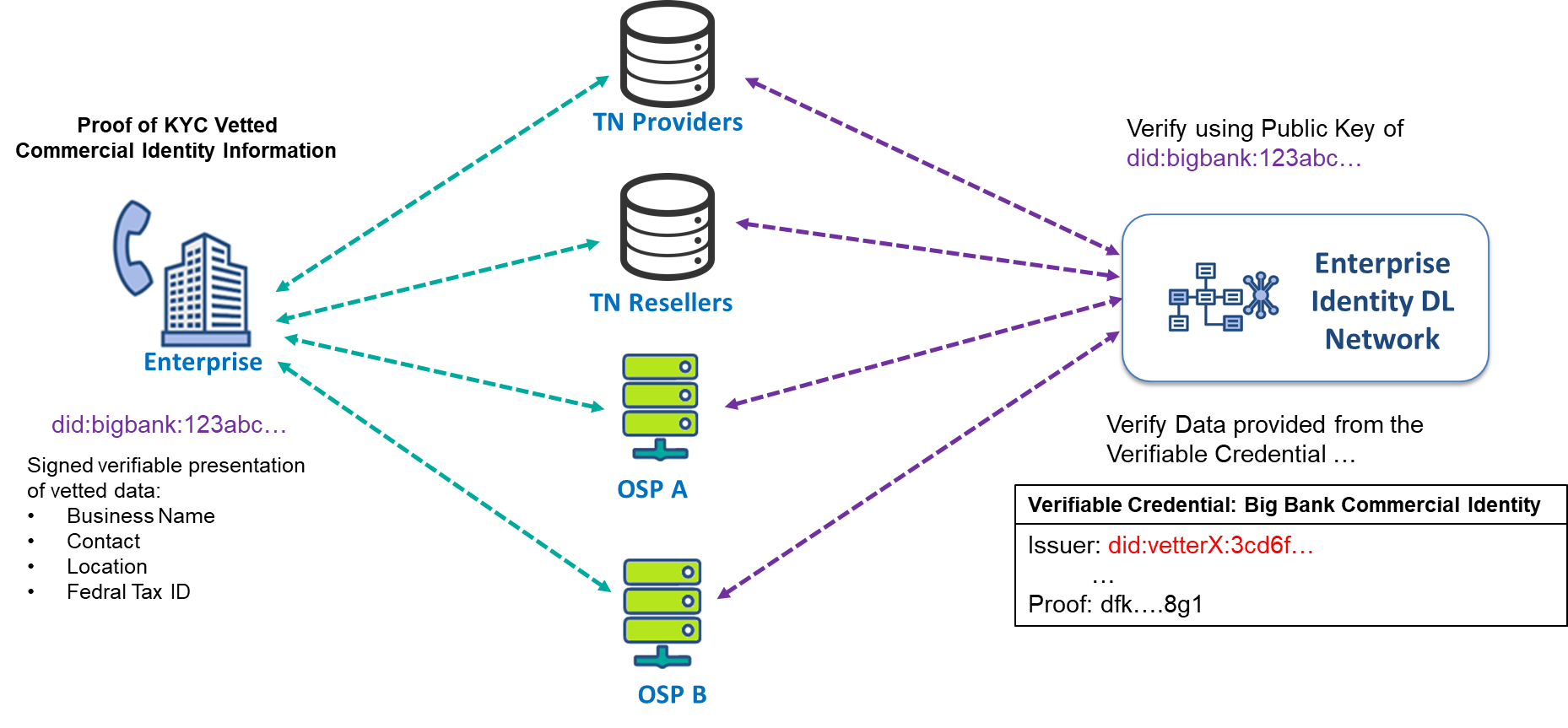
The KYC vetting process of an ‘Enterprise Identity’ will authenticate a business entity’s identity on the EIDLN for a TNR, Brand Enterprise or BPO Enterprise. As part of the KYC compliance process, the business entity will provide commercial identity information to the KYC vetter to confirm that the business representing the ‘Enterprise Identity’ is a legitimate business entity and whom they claim to be.

The KYC Vetter compliance process will perform vetting of the enterprise representing the ‘Enterprise Identity’ checking key commercial identity information for example: - Business Trading Name, Business Brand Name, Contact Name, Contact details, Location, Incorporation, and Federal Tax ID. Upon successful vetting of the enterprise’s commercial identity info, the KYC vetter can create a signed Verifiable credential claim containing the key commercial identity information verified.

******

***Figure 8.1 KYC Vetter creation of Commercial Identity Credential Claim***

This Verifiable Credential claim containing the commercial identity information, can now be used by the enterprise to share and prove that the ~~he~~ commercial identity information is vetted, that can be used by other actors on the EIDLN. I.e., a TNSP, TN reseller, or an OSP that is hosting the TN for the enterprise for the first time and required commercial information to satisfy their own KYC compliance process.

******

***Figure 8.2 Using the VC Commercial Identity to prove with multiple actors***

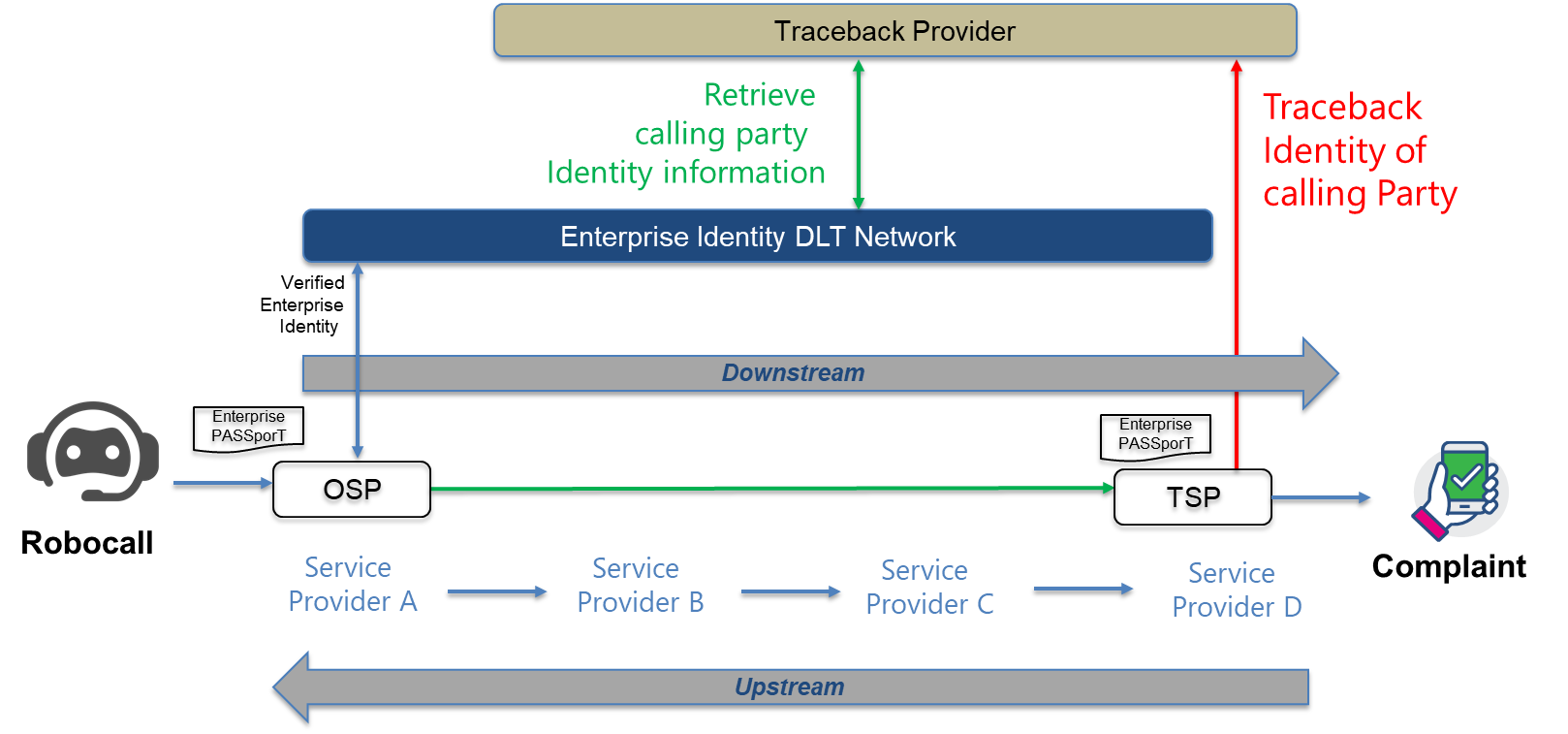
When an enterprise is requested to provide commercial data to another EIDL actor, they can present verifiable data that is contained within the verifiable credential claim that is proofed by the KYC vetter. The EIDL actor can then verify that the data provided for this enterprise has been vetted and can be trusted.

## Principle #6. Require Traceback Cooperation in Contracts

Traceback is a cooperative effort by telecommunications providers that starts with a terminating service provider (TSP) possessing evidence of suspicious traffic. A Traceback Provider will trace and identify the source of illegal robocalls, in order to provide for the robust protection of voice networks and users of such services from fraudulent, abusive, and/or unlawful robocalls.

When a complainant has reported to the TSP about a suspicious robocall, which has been verified using the EIDLN, the TSP can provide the enterprise identity of the calling party as provided within the Enterprise PASSporT for that call, to the Traceback Provider.

The Traceback Provider can directly identify the calling party by retrieving the commercial identity data for the enterprise identity from the EIDLN, without the need to request ~~on~~ each upstream service providers information to identify the calling party.



***Figure 8.2 Traceback of robocaller identity using EIDLN***

# ANNEX A: Example of Open API for EIDL

This annex provides an example of the RESTful API definitions used in the ATIS Enterprise Identity DLT PoC, used to integrate organizations with the Enterprise Identity DLT Network solution. By using the *OpenAPI Specification (OAS Version 3.0.3* for defining the API, it is possible to integrate with the DLT using the following programming applications, Curl, Java, Android, Obj-C, JavaScript, C#, PHP, Perl or Python. For the purpose of the API described in this Annex the examples provided are using Curl.

***Note:*** The Enterprise Identity DL network OpenAPI definition format, is available download~~able~~ from ATIS (Location TBD)

## Organization Accounts

These APIs are used for organization account integration with the DLT.

### createAccount

Create an account for an organization includes a TNSP, a TNR, a Brand Enterprise, a BPO, or a KYC Vetter.

This will generate an account identity and a Key Pair. These will be stored in a secure fashion by the organization client DLT Integration API. The DID will be returned in the response payload and will be used to identify the organization going forward in both the integration API and in the DLT

/accounts

**Usage and SDK Samples**

curl -X POST <http://localhost/v1/accounts>

**Parameters**

Body parameters

|  |  |
| --- | --- |
| **Name** | **Description** |
| body \* | {  An account to represent a given organization  Required: name,type  did: *string*  The DID/UUID of the account. This is assigned during the creation (POST) process and should be left blank for POSTs.  name: *string*  type: *integer (int32)*  Type of organization. 0 - KYC Vetter, 1 - TNR, 2 - Brand, 3 – BPO, 4 – TNSP, 5 – CSP, 6 - TA  publicKey: *string*  The DID document URI for the current public key for this account  createdTS: *integer (int64)*  Time that this account was created in UTC milliseconds from EPOCH  updatedTS: *integer (int64)*  Time that this account was last updated in UTC milliseconds from EPOCH  } |

**Responses**

**Status: 201 - Expected response to a valid request**

*Schema*

{

*An account to represent a given organization*

did: *string*

DID of the account. This is the UUID assigned during the creation (POST) process. This is used going forward to identify the organization.

name: *string*

type: integer (int32)

Type of organization. 0 - KYC Vetter, 1 - TNR, 2 - Brand, 3 – BPO, 4 – TNSP, 5 – CSP, 6- TA

publicKey: *string*

The DID document URI reference on the DLT of the current public key for this account

createdTS:  *integer (int64)*

Time that this account was created in UTC milliseconds from EPOCH

updatedTS: *integer (int64)*

Time that this account was last updated in UTC milliseconds from EPOCH

}

**Status: default - unexpected error**

Schema

 {

Required: code,message

code: integer (int32)

message: string

}

### getAccount

Retrieve account information from the Enterprise Identity DL Network, based on the DID/UUID for the account.

/accounts/{orgIdParam}

**Usage and SDK Samples**

curl -X GET "http://localhost/v1/accounts/{orgIdParam}"

**Parameters**

Path parameters

|  |  |
| --- | --- |
| **Name** | **Description** |
| orgIdParam\* | String  *The DID/UUID of the organization*  Required |

**Responses**

**Status: 200 - Expected response to a valid request**

* [Schema](file:///H:\Innov8\ATIS\DLT%20PoC\API\API%201.0.2-SNAPSHOT.html#responses-getAccount-200-schema)

 {

*The account information to represent a given organization DID/UUID*

did: *string*

DID of the account. This is the UUID assigned during the creation (POST) process. This is used going forward to identify the organization.

name: *string*

type: integer (int32)

Type of organization. 0 - KYC Vetter, 1 - TNR, 2 - Brand, 3 – BPO, 4 – TNSP,

5 – CSP, 6 - TA

publicKey: *string*

The DID document URI reference on the DLT of the current public key for this account

createdTS:  *integer (int64)*

Time that this account was created in UTC milliseconds from EPOCH

updatedTS: *integer (int64)*

Time that this account was last updated in UTC milliseconds from EPOCH

**Status: default - unexpected error**

Schema

 {

Required: code,message

code: integer (int32)

message: string

}

## TNAuthorization

Allows an authorized organization to assign or delegate a TN to another organization on the Enterprise Identity DL Network.

***Note:*** the term ALLOCATION in the API call refers to TN, assignment of a TN from a TNSP to a TNR or TNSP to a BRAND, and Delegation of a TN from a TNR to a Brand or Brand or BPO.

/tnallocation

**Usage and SDK Samples**

curl -X POST "http://localhost/v1/tnauthorization"

**Parameters**

Body parameters

|  |  |
| --- | --- |
| **Name** | **Description** |
| body | *{*  *tnAuthorization: [*  *{*  *Required: TNauthorizationByOrganisationId,ToOrganisationId,validEndDate,validStartDate*  tnId: *string*  *Identifier of the telephone number.*  issuerId: *string*  *DID of the Organization that is allocating the TN.*  subjectId: *string*  *DID of the Organization that the TN is being authorized to.*  intendedUseOfAuthorization: *string*  *Text field that allows you to specify the intended purpose the TNs in the authorization will be used for.*  tnIdType*: integer (int32)*  *Type of telephone number identifier. (0-E.164, 1-UUID, 2-DID, 3-Verifiable Credential). This field will allow us some flexibility as we grow the solution to allow different types of identifiers for TNs.*  *authorizationStartDate:  string (date-time)*  *Time that this TN Authorization becomes valid. Must be a future date.*  authorizationEndDate*:  string (date-time)*  *Time that this TN Authorization will be revoked. Must be a future date. If this is not specified, it is assumed there is no end date*  *}*  *]*  *}* |
|  |  |

**Responses**

**Status: 201 - Return the newly created TNAuthorization**

* [Schema](file:///H:\Innov8\ATIS\DLT%20PoC\API\API%201.0.2-SNAPSHOT.html#responses-tnAllocation-201-schema)

 {

tnAuthorizationResponse:[

{

verifiableCredential:

  {

@context:[

*Sets the context, which establishes the special terms we will use like issuer and alumniOf*

id:  string

*Identifier for the credential*

type: string

*Credential types, which declare what data to expect in the credential*

issuer: string

*Entity DID that issued the credential*

issuanceDate:  string (date-time)

*Time that this verifiable credential was issued*

}

proof:

 {

*Digital proof that makes the credential tamper-evident.*

type:   string

*The cryptographic signature suite that was used to generate the signature*

created: string (date-time)

*Time that this signature was created.*

proofPurpose:  string

*Purpose of this proof*

verificationMethod:  string

*The identifier of the public key that can verify the signature*

jws:  string

*The digital signature value*

}

**all of:**

 {

Required: authorizedByOrganisationId,authorizedToOrganisationId,validEndDate,validStartDate

tnId: *string*

*Identifier of the telephone number.*

issuerId: *string*

*DID of the Organization that is allocating the TN. T*

subjectId: *string*

*DID of the Organization that the TN is being allocated to.*

intendedUseOfAuthorization: *string*

*Text field that allows you to specify the intended purpose the TNs in the authorization will be used for.*

tnIdType*: integer (int32)*

*Type of telephone number identifier. (0-E.164, 1-UUID, 2-DID, 3-Verifiable Credential). This field will allow us some flexibility as we grow the solution to allow different types of identifiers* for TNs.

authorizationStartDate:  string (date-time)

Time that this TN Authorization becomes a valid allocation. Must be a future date*.*

authorizationEndDate*:  string (date-time)*

*Time that this TN* Authorization *will be revoked. Must be a future date. If this is not specified, it is assumed there is no end date*

*}*

]

}

**Status: default - unexpected error**

Schema

 {

Required: code,message

code: integer (int32)

message: string

}

## Vetter

### createVettedOrganisationOnDLT

Allows a KYC vetter to update the vetted organization authorization status to the DLT

/vetterAUthorization/{orgIdParam}/vetter/{vetterIdParam}

**Usage and SDK Samples**

curl -X POST "http://localhost/v1/vetterAUthorization/{orgIdParam}/vetter/{vetterIdParam}"

**Parameters**

Path parameters

|  |  |
| --- | --- |
| **Name** | **Description** |
| orgIdParam\* | String  *The DID/UUID of the subject organization vetted status change*  Required |
| vetterIdParam\* | String  *The DID/UUID of the vetter*  Required |

Body parameters

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | |
| body \* | }  status:  integer (int32)  *Status of the vetting process. 0 - In Progress, 1 - Complete*  result:  integer (int32)  *Result of the vetting process. 0 - Authorized, 1 - Not Authorized, 2 - Revoked*  vettedTS:  string (date-time)  *Time that this authorization became effective*  version: integer (int32)  *In the case that an authorization is updated on the ledger, we will update the version. This will help to identify and order related authorizations.*  } |

**Responses**

**Status: 201 - Null response**

Schema

 {

Required: organization ,status ,vetter

organization:  {

*The DID/UUID of the subject organization vetted status change*

did:   string

DID *of the organization being vetted. This is assigned during the creation (POST) process. This is used going forward to identify the organization.*

name: string

type:  integer (int32)

*Type of organization. 1 - TNR, 2 - Brand, 3 – BPO*

createdTS:  string (date-time)

*Time that this account was created*

updatedTS:  string (date-time)

*Time that this account was last updated*

}

vetter:  {

*An account to represent a given organization*

did:  string

*DID of the TA / KYC vetter account.*

name: string

type:  integer (int32)

*Type of organization. 0 - KYC Vetter, 6 - TA*

createdTS:  string (date-time)

*Time that this account was created*

updatedTS: string (date-time)

*Time that this account was last updated*

}

status:  integer (int32)

*Status of the vetting process. 0 - In Progress, 1 - Complete*

result:  integer (int32)

*Result of the vetting process. 0 - Authorized, 1 - Not Authorized, 2 - Revoked*

vettedTS:  string (date-time)

*Time that this authorization became effective*

version: integer (int32)

*In the case that an authorization is updated on the ledger, we will update the version. This will help to identify and order related authorizations.*

}

**Status: default - unexpected error**

Schema

 {

Required: code,message

code: integer (int32)

message: string

}

### getVettedOrganisationAuthorization

Retrieve the vetted authorization status and history for a given organization

/vetterAUthorization/{orgIdParam}/vetter/{vetterIdParam}

**Usage and SDK Samples**

curl -X GET "http://localhost/v1/vetterAUthorization/{orgIdParam}/history"

**Parameters**

Path parameters

|  |  |
| --- | --- |
| **Name** | **Description** |
| body\* | Id: String  *The DID/UUID of the subject organization vetted status change*  Required  status: integer(int32)  *Type of status response . 0 - Current, 1 - History* |
|  |  |

**Responses**

**Status: 201 - Null response**

Schema

 {

Required: organization ,status ,vetter

organization:  {

*The DID/UUID of the subject organization vetted status change*

did:   string

*DID of the account. This is assigned during the creation (POST) process. This is used going forward to identify the organization.*

name: string

type:  integer (int32)

*Type of organization. 1 - TNR, 2 - Brand, 3 - BPO*

createdTS:  string (date-time)

*Time that this account was created*

updatedTS:  string (date-time)

*Time that this account was last updated*

}

**all of:[ NOTE: :List authorization history if Status = 1(History )**

vetter:  {

*An account to represent a given organization*

did:  string

*DID* *of the TA/ KYC vetter account.*

name: string

type:  integer (int32)

*Type of organization. 0 - KYC Vetter, 6 - TA*

createdTS:  string (date-time)

*Time that this account was created*

updatedTS: string (date-time)

*Time that this account was last updated*

}

status:  integer (int32)

*Status of the vetting process. 0 - In Progress, 1 - Complete*

result:  integer (int32)

*Result of the vetting process. 0 - Authorized, 1 - Not Authorized, 2 - Revoked*

vettedTS:  string (date-time)

*Time that this authorization became effective*

version: integer (int32)

*In the case that an authorization is updated on the ledger, we will update the version. This will help to identify and order related authorizations.*

]

}

**Status: default - unexpected error**

Schema

 {

Required: code,message

code: integer (int32)

message: string

}

### OSP API goes here (or anywhere it this Annex)

1. Available from ATIS at : <<https://www.atis.org/>> [↑](#footnote-ref-2)
2. Available from IETF at : <https://www.ietf.org/> [↑](#footnote-ref-3)
3. Available from W3C at : <<https://www.w3.org/>> [↑](#footnote-ref-4)