Electronic Skill Credential Standard

Technical Specification - Version 1.0RC

(**RELEASE CANDIDATE FOR PUBLIC REVIEW**)

January 2019

Table of Contents

Introduction and Purpose

The jobs and skilling ecosystem is a complex arena with a number of frictions preventing an optimal supply and demand equilibrium, resulting in not only underemployment but also significant wage gaps. These frictions include low trust, information asymmetry, low comparability and portability, and supply gaps amongst others.

A core issue in the jobs and skills arena is the difficulty an individual faces in showcasing their skills and exchanging credentials in a trusted manner. This document proposes a set of electronic standards for machine-readable data to represent various credentials in the skilling ecosystem across industry verticals. An electronic standard for data representing credentials allows certificates to be issued in digital, machine-readable formats. Machine-readable formats in-turn make it possible for an individual to freely transfer credentials in a trusted manner and apply for jobs remotely.

Currently, skills certificates are issued in paper form, meaning they are neither verifiable digitally nor machine readable. In contrast, digital credentials are freely portable and easily verifiable at scale and speed by employers and job matching platforms, whilst continuing to allow print and other visual forms for human consumption.

This document presents an open specification which can be incorporated into the information management systems of skill training providers, apprenticeships, employers, testing agencies, or others in the ecosystem. It is targeted towards technology and implementation teams within organisations who issue certificates.

In this document the terms *credential* and *credentials* are used to mean a *qualification or achievement of a person or entity used to indicate their suitability for something*. A certificate or other form of attestation is typically issued to award such a qualification.

Purpose

The purpose of the electronic credential specification is to provide certificate recipients with a means for participating in a digital economy where:

- They can **digitally transfer skill credentials** as per training outcomes using trusted and verifiable means
- Skilled individuals can be **automatically matched** to jobs across regions
- Participating organisations can issue certificates using their **preferred vocabularies for academic standards** without inhibiting exchange of information on skills supply and employment opportunity

Principles

Verifiability: Authenticity of a credential should be digitally verifiable by any application to which it is presented. This verification need not require the physical presence of the credential holder.

Portability: The credentials should be digitally portable across systems participating in the ecosystem. Physical certificates are unrestricted -- the recipient can present the certificate to any party of her choice. The same property should be preserved. This includes easy digital storage in the control of the recipient of the credentials and easy consented transfer & sharing by the recipient for various purposes.

Permanence: The credentials should continue to exist and be valid beyond the lifetime of the institution where it was issued. That is, if an organisation which has issued a credential subsequently ceases to exist, the issued credentials remain verifiable and portable across the ecosystem.

Self-Describing: The credential model should be self-describing in a manner that the consumer of the credential does not require private sources of information to validate or understand it. In practice, this means that any declaration of skill level or reference to an awarding institution links to a publicly accessible and unencumbered source of further information. This makes the credential truly portable across the ecosystem, since anyone to whom it is presented can make sense of the information contained within and have enough context to compare the accreditation with another certificate.

Current State

Issuers

There are a variety of issuers of certificates within the skill ecosystem. An individual trainee may complete a training course designed to provide a basket of skills across the spectrum of soft skills, life skills and trade skills. In some cases, the trainee may receive a single certificate which covers the entire program. In other scenarios, the trainee may receive multiple certificates from different trainers or assessors. Both methods present challenges as we shall see.

Issuing a single certificate for a basket of skills is problematic in scenarios where the certificate does not include details of the contents of the basket. It makes it difficult for employers to get a complete picture of the employee's capabilities. To ascertain the ability of an employee, employers must either conduct their own evaluations or have experience with the competencies of trainees graduating from a particular program. Both alternatives increases the cost and complexity of employer processes while lowering the value of the certificate issued.

Conversely, issuing multiple certificates can be challenging for employers when this practice is not consistently followed amongst issuers. Without intimate familiarity with the certification practices of the issuer, employers will not know which certificates to expect from a trainee. This lowers the acceptability of a certificate reducing it to the set of employers who are familiar with the issuer.

Assessors

The role of the assessor is, in some instances, separate from the role of the certificate issuer. Assessors may be private bodies which specialise in a specific skill (such as a Driving School) or may be regulatory bodies (such as a Sector Skills Council) which certify many skills within a specific industry.

Standards

In the absence of certification standards maintained across issuing bodies, employers must be familiar with the issuer's reputation in order to compare potential employees. Migrating certificates to an industry-led set of standards will allow portability across the ecosystem. *Ensuring that the standards are represented in digital, machine-readable formats will allow certificate management, candidate evaluation and mobilisation at scale*.

There are a network of standards available within the skills domain. The National Skills Qualification Framework (NSQF) outlines a taxonomy of industry skills and competency levels which are broken down into Qualification Packs (QPs) comprising a collection of National Occupational Standards (NOS). Utilising this framework of standards for issuing certificates across issuing institutions can dramatically increase the value of certificates for trainees and employers. However, *it is imperative that a mechanism is established for these standards to evolve led by industry demand for new skills* in existing domains as well as for entirely new domains.

Proposed Specifications

Conceptual Model

A certificate captures a relationship between the following entities

- Issuer
- Recipient
- Domain
- Standard(s) accomplished
- Assessor
- Evaluation
- Time

The **issuer** certifies that the **recipient** has **accomplished** specified **standard(s)** in a **domain** based on an **evaluation** conducted by the **assessor** at a **particular time**.

Note that a single certificate may be issued for the recipient's accomplishment of multiple standards.

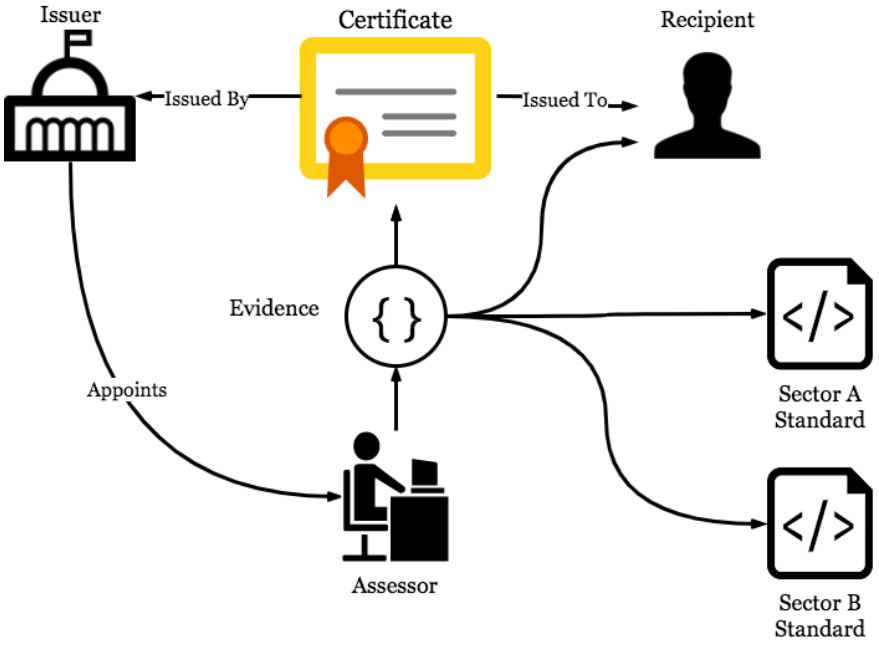


Fig 1 The actors in the certificate ecosystem

Data Modelling Principles

1. All entities in the certificate issuance transaction must be represented using machine-readable objects
2. Each object in the data-model is a representation of an entity
3. Assign strong identifiers[1] to all entities where some other party might reasonably want say something about an entity or do something with the representation of the entity. The strong identifiers for the entities should never be altered.

Visual Representation

A visual representation of the credential (rendering) can be generated using data from the JSON-LD representation and subsequently embedded into the JSON-LD object. Applications issuing credentials can apply business-specific rules for rendering certificates. In some contexts, it may be preferable to create multiple renderings -- the granularity of the details presented can be tuned according to the needs outlined by the use case.

Applications are also open to using format of choice when embedding the rendered certificate. Formats such as PDF documents or images are well-suited for long-term survivability and compatibility. Others such as HTML are easier to generate. One must note that HTML standards evolve over time. To ensure that embedded HTML will render well twenty or thirty years in the future, a very minimal and high-compatibility subset of HTML should be used.

[1] An identifier is any alphanumeric string which has a denotational property of representing the identity of an entity

Entity Identity

An important part of issuing credentials is to identify the entities involved in the process. To issue the credential a few different types of entities participate: institutions, individuals, documents and standards. Since a credential is a permanent document by nature which may be used and remain valid over a long period of time, two broad principles apply for all identifiers:

- **Singularity**: An identifier should resolve to a single entity, for example a person or an institution are single entities.
- **Permanence**: An identifier should not be reassigned to a different entity at a future point.

For example, a phone number is not a good means of identifying a person for the purpose of issuing a credential. A credential must be valid for many years, it may be presented for validation 10, 20 or even 50 years later. A person's phone number may change in the intervening time-period and the number given up by that person may be re-allocated to another different person. Hence an ID, such as a phone number, which does not provide the guarantee of resolving to the same person for the duration of use of the credential is not well-suited for identifying the recipient.

Conversely, a document such as a passport, a ration card or a voter identity card (which will eventually expire) is a better candidate for identifying a person since the passport number or voter id number is not reissued and will always continue to represent the same person over time.

Structural Model

The structural model of a credential aligns with the OpenBadges v2.0 schema and adds extensions where necessary.

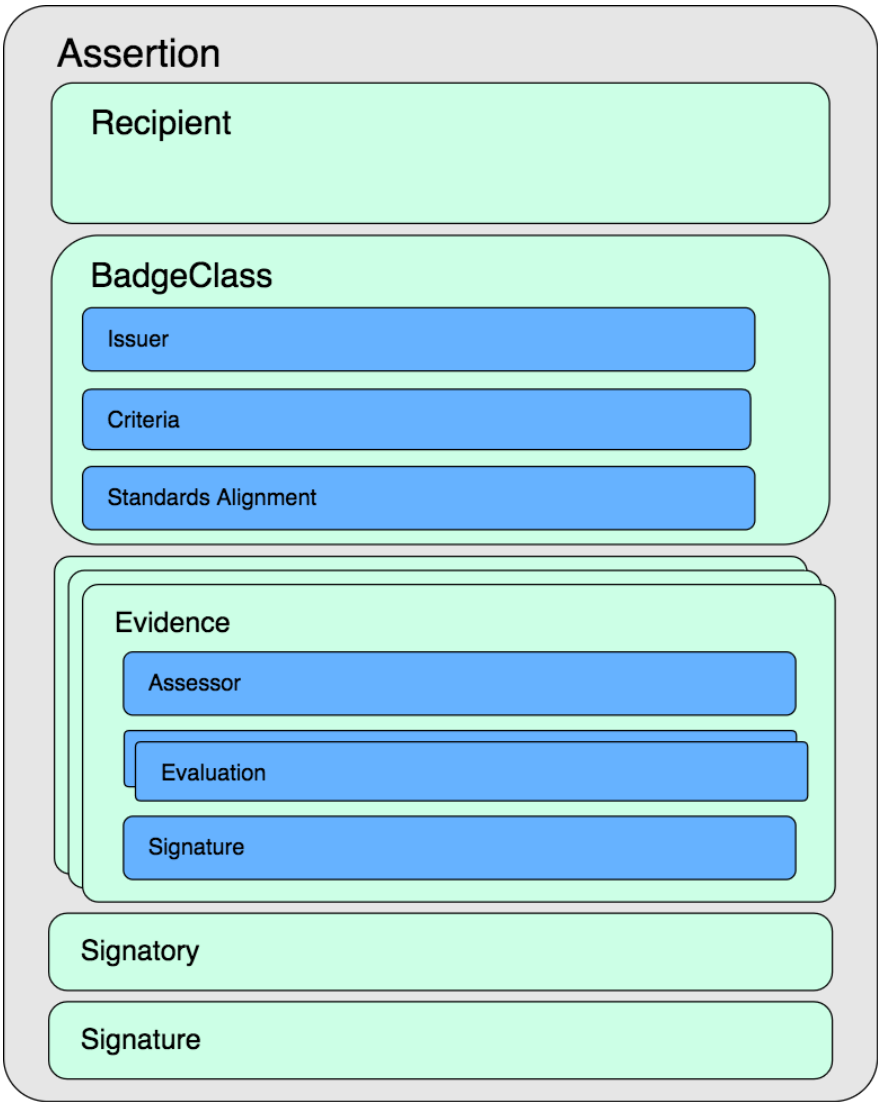


Fig 2: The Credential Structural Model

Assertion

An assertion is a statement of fact. The credentials issued are statements of fact about recipient's accomplishments. Hence, the root object of the credentail is called an assertion. The assertion object is a container object for sub-objects representing the issuer, recipient and the recipient's credentials being certified. The assertion object described in this specification extends the OpenBadges v2 **`Assertion`** object with a few additional properties via the **`CertificateExtension`** class.

Recipient

The recipient of the certificate may be an individual or an organisation. The recipient of the certificate is identified by an **`IdentityObject`**.

BadgeClass

The **`BadgeClass`** describes a category of credentials that is issued. The badge class contains details of the issuer of the credential, the skill domain & standard achieved. For

instance, one category of of credentials may be a School Leaving Certificate issued by a school board, another could be a Bachelor's Degree certificate issued by a University.

Issuer

The issuer of the credential will be an organisation or institution which is a competent authority to certify recipients. Note that the issuer is different from the signatory (i.e. the individual(s) who applies his/her signature to the credential).

Standard

Standard is a skill definition or a competency defined by a certified standards body for the domain. Any given standard must be uniquely identifiable and will also commonly be part of a framework which defines relationships between standards in a domain. **`BadgeClass`** descriptions link to defined standards via **`AlignmentObjects`**.

Evidence

The credential may contain one more list items of evidence which have been assessed in support of the credential. Each item of evidence may contain an assessment performed by the assessor. The evidence may link to standards criteria and a level of mastery which the recipient displays for the given criterion.

Assessor

The assessor evaluates a trainee's competencies. The assessor is an organisation or institution which has been certified as a competent authority to assess skills for a given domain standard(s).

Data Model

Standards

The credentials data model uses the following standards for defining a credential:

- Alignment with OpenBadges v2 for defining accomplishments. The OpenBadges schemas are extended in some areas for specific use cases.
- RDF 1.1 (Resource Description Framework) is used as the means for expressing the data contained in a credential.
- The preferred RDF serialisation format is JSON-LD, however, consumers of a credential are free to use alternative serialisation formats such as RDFs or a triple expression language such as Turtle.
- The credentials model uses (and in some places extends) the vocabulary of Classes and Properties described by schema.org and the WebPayments specification.
- Additionally the credentials model defines some new classes of objects.
- Each object class used in the model must be defined in terms of an RDF schema. Under the JSON-LD serialisation format, the schema needs to be published and made available at a specific web URL for validation and consumption of credentials (a hypothetical schema URL is used in the example below).

Identifiers

Based on the above principles for identifying entities using strong identifiers the following apply when assigning identifiers for objects.

- Use URIs wherever possible for compatibility with web protocols
- Since we represent data in JSON-LD, the preferred URI for an entity is a dereferenceable URL which points to the location where the entity's JSON-LD representation can be found
- If a URL is unavailable, use a URN. The id must be issued within a well-known namespace with an established namespace identifier (NID)[2]
    - URN: `urn:{namespace-id}:{private-id}`
- If an established NID is unavailable to create a URN, the TAG URI scheme[3] can be used when there is a web domain name for the issuer of the ID.
    - TAG: `tag:{issuer-domain},{domain-register-date}:{id-type}:{id}`
- Where URNs and TAGs cannot be derived, strong globally unique identifiers should be created via locally unique identifiers using **`IdentityObject`** where the **`@type`** is a namespace describing the type of identifier and the identity is the local ID (see an example in the certificate below).
    - **`@type`** must contain a well-known id for the issuer of the local identifier

Identifier Namespaces

Below, we propose some informal namespaces for commonly used identifiers types. These namespaces can be used as the NID for URNs. The format used for NIDs is **`in.<domain>.<id-issuer>.<id-type>`**, where domains are .gov, .edu, .com & .org. This format can be used to construct new informal namespaces for NIDs.

Non-unique namespaces cannot be used as NIDs for URNs but may be utilised as the **`@type`** property for a component of a composite identity represented by an **`IdentityObject.`**

Identifier Type	Namespace	Example
PAN	`in.gov.itd.pan`	`urn:in.gov.itd.pan:ZZZZZ00000`
GSTN	`in.gov.gstn.gstn`	`urn:in.gov.gstn.gstn:Z00000000000001`
Driver License (KA)	`in.gov.ka-dot.dl`	`urn:in.gov.ka-dot.dl:ZAAAAAAAAAAAB`
Driver License (MH)	`in.gov.mh-dot.dl`	`urn:in.gov.mh-dot.dl:ZOOOOOOOOOOAB`
Aadhaar	`in.gov.uidai.aadhaar`	`urn:in.gov.uidai.aadhaar:11111111111`
Voter Identity	`in.gov.eci.voterid`	`urn:in.gov.eci.voterid:X11111111X`
Passport	`in.gov.mea.psprt`	`urn:in.gov.mea.psprt:XX99999999`

[2] See the section on Identifier Namespaces
[3] The 'tag' URI scheme, https://tools.ietf.org/html/rfc4151

Roll Number	`in.<dom>.<iss>.rollno`	`urn:in.gov.msde-dgt.rollno:999999999`
Name	`name`	`Ram Singh`
Date of Birth	`dob`	`Date of birth in YYYY-MM-DD format`
Photo	`photo`	`<base64 encoded>`

Table 1: Informal namespaces for commonly used identifiers

URN Identifiers

URNs are URIs which follow the Universal Resource Name (URN) scheme. A URN consists of three parts separated by colons (:)

urn : **{nid}** : **{nss}**

namespace id — namespace specific string

The namespace id (nid) defines the type of identifier. The consumer of a URN can make processing decisions based on the nid. The consumer could use the nid to determine how to resolve the namespace specific string to an entity. For instance, the isbn namespace is registered for ISBN book codes.

The namespace specific string (nss) is the value of the ID within the nid. Continuing the above example of books, the nss would be the ISBN of a specific book such as 8120351312.

Together, the combination of **`urn:isbn:8120351312`** forms a URN which uniquely identifies a book.

Institution Ids

Institutions are identified in two scenarios. The first of these is an institution is an issuer of credentials or is an assessor of a subject (recipient). In this scenario, the institution should be identified using a URL which points to the institution's profile (in JSON-LD format). The profile must contain the properties issuer/assessor properties defined in the model below. Additionally, the profile may also contain more information about the institution (e.g. rankings, institutional standards, authorisations etc) using the appropriate RDF schemas. The profile may also contain links to other tools where such information about the institution can be retrieved.

Second, as the recipient of credentials, institutions should also be identified using a URL which points to the institution's profile. The profile data may contain embedded objects detailing publicly available and well-known identifier types such as a GST number or a permanent account number (PAN). The would be represented using the RDF **`sameAs`**[4] property whose value is an **`IdentityObject`**. When using such identifiers the identity can be represented as a URN.

[4] https://www.w3.org/TR/owl-ref/#sameAs-def

Individual Ids

Individuals are identified in two scenarios. First as recipients of credentials and second, as signatories to a credential. In both scenarios, the individuals should be identified by an **`IdentityObject`** which contains one or more identifying attributes. If the individual can be identified by a HTTP URL, the **`IdentityObject`** should be of **`@type: "url"`**. In case of an identity which is comprised of multiple parts, an **`IdentityObject`** of **`@type: "composite"`** should be used containing components which are in turn **`IdentityObjects`** of a variety of types.

Object Ids

Machine-readable objects such as **`Assertions`**, **`BadgeClasses`** & **`Evidence`** should be identified for the purpose of validation and comparison.

Assertions & Evidence

Ideally, Assertions and Evidence should be identified via a UUID encoded as a URN.
Example: **`urn:uuid:ec58b28e-a6ab-49c2-a24d-ebefa02476cd`**

If use of a UUID is not feasible, for instance if documents are numbered serially, the TAG URI scheme should be used where a document id is prefixed with a namespace consisting of the issuer's domain, its registration date[5] and the document type.
Example: **`tag:examplecompany.com,2010-09:marksheet:9871624`**

BadgeClass

BadgeClasses should ideally be identified via a HTTP URL which returns a JSON-LD representation of the class. If the BadgeClass is defined for a single ad-hoc usage and is not published at a URL, a UUID encoded as a URN is recommended.

Object Model

Objects and the properties as defined by this credentials specification are detailed here.

Assertion

The base of the model is an assertion. The assertion class is defined as part of the **`OpenBadges`** v2.0 specification. Some additional properties are added via **`CertificateExtension.`**

- **`Id`** which uniquely identifies the assertion. If assertions are verified by signing, a UUID should be used from the **`urn:uuid`** namespace. If hosted verification is used, the Id should be a HTTP URL to a JSON-LD document containing the certificate data.
- **`Type`** An array containing the strings Assertion, Extension and **`CertificateExtension`**
- **`IssuedOn`** when the credential was issued
- **`Recipient(s)`** person(s) or organisation(s) who received the credential.

[5] The registration date should be the date the domain was first registered by an entity in an unbroken spell to the present date.

    - If the certificate contains signed evidence, the subject of the evidence can be assigned a relative IRI as an @id and referenced as the recipient of the certificate.
        - **`"recipient": {"@id": "#/evidence/0/subject"}`**
    - If there are multiple recipients, the element should be a list of entities
- **`Badge`** an embedded BadgeClass object which describes the type of certificate being awarded or a HTTP URL which contains a machine-readable representation of a BadgeClass
- optional **`Image`** which is a HTTP URL to a *baked*[6] PNG or SVG image
- **`Evidence`** containing details of competencies assessed
- optional **`Expires`** containing the date the certificate ends validity
- **`Verification`** containing the string **`LinkedDataSignatures`** for signed certificates
- **`Narrative`** as text or markdown text which can be used to describe and connect multiple pieces of evidence

CertificateExtension

The Assertion type from OpenBadges is extended by the **`CertificateExtension`** which adds the following field definitions.

- optional **`IssuedThrough`** service or program through which the credential is issued (eg: PMKVY)[7]
- optional list of **`Signatory(s)`** who signed the certificate
- **`PrintUri`** a HTTP URL which points to a printable version of the certificate (for instance the digilocker URL where the certificate HTML is stored).
    - The printable version of the certificate can be embedded into the machine-readable data using data URIs. The encoding scheme is: **`data:[mime-type][;base64],<base64-encoded-data>`**
        - A base64 encoded PDF document could be represented as a data URI: **`data:application/pdf;base64,<base64 encoded binary PDF>`**
        - A base64 encoded image document can be represented as a data URI: **`<base64 encoded binary JPG>`**
    - If there are multiple printable documents to be embedded, a list of URIs can be added.
- optional **`ValidFrom`** containing the date the certificate begins validity[8]
- **`Signature`** added by the issuer using its private key for signed verification

CompositeIdentity Object

OpenBadges v2 uses IdentityObjects to represent the recipient of a certificate. We extend it to represent a composite identity which is composed of a sub component **`IdentityObjects`**.

---

[6] See the OpenBadges baking specification: https://www.imsglobal.org/sites/default/files/Badges/OBv2p0Final/baking/index.html

[7] https://schema.org/issuedThrough

[8] https://schema.org/validFrom

To facilitate composite identities, the specification introduces:

- a new **`IdentityObject`** **`@type`** identifier called **`composite`**
- a new property **`annotation`** which can be used to qualify the identity type
- a new property **`components`** for composite identities. This property will contain sub-fields of the identity

The fields of a composite **`IdentityObject`** will be

- **`Type`** will be Composite
- **`Hashed`** will be false
- **`Identity`** will be omitted
- **`Components`** will be an array of other **`IdentityObjects`**

Additionally, we define the following **`IdentityObject`** **`type`** identifiers to extend the set of profile identifier properties defined by OpenBadges.

- **`composite`**: the identity is expressed as a composite identity containing **`components`**
- **`urn`**: the identity is expressed as a Uniform Resource Name (urn) URI
- **`tag`**: the identity is expressed as a Tag URI
- **`name`**: the identity is expressed as a name, often used in conjunction with **`annotation`** to represent a father's or a spouse's name. May be part of a composite identity
- **`photo`**: the identity is expressed as a photo, either a HTTP URL for a image or a data URI containing the mime type and image data in base64 encoding. May be part of a composite identity
- **`dob`**: the identity is expressed as the date of birth in YYYY-MM-DD format. May be part of a composite identity
- **`gender`**: the identity is expressed as the gender of the individual. May be part of a composite identity.

BadgeClass

- **`Id`** which uniquely identifies the **`BadgeClass`**
    - If the credential represents a larger category which may be issued multiple times, it must be a HTTP URL where the JSON-LD description of the **`BadgeClass`** is available.
    - If the certificate is short-lived and unlikely to be issued repeatedly, a **`urn:uuid`** class identifier can be used
- **`Type`** will be the string BadgeClass
- **`Name`** The name of the credential represented by this **`BadgeClass`**
- optional **`Description`** A short text description of the credential
- optional **`Version`** describing the version of the **`BadgeClass`**, which can be updated as the credentials evolve over time.
- optional **`Image`** an image representing the credential, may be a HTTP URL where the image can be found or a data URI including mime-type

- optional **`Criteria`** an embedded object describing the criteria for achieving the credential or a HTTP URL of a JSON-LD object describing the criteria
- **`Issuer`** profile of the person or organisation entity which issued the credential. It is recommended that the complete JSON-LD object representing the issuer is embedded into the BadgeClass. The set of fields to uniquely identify the issuer and verify the issuer's signature on the credential must be embedded into the certificate. If the complete issuer profile is not embedded into the BadgeClass, then the **`@id`** of the embedded object must be a HTTP URL from which a more complete JSON-LD object can be retrieved.
- optional **`Alignment`** which maps the **`BadgeClass`** to one or more academic standards.
- optional **`Related`** an object or list of objects relating this **`BadgeClass`** to other **`BadgeClasses`**.
    - If the **`BadgeClass`** is part of a series, for instance if a new class is created for each batch or semester, this may be used to link to previous instances of the same class.

Profile: Issuer/Assessor/Trainer

- **`Identifier or ID`** to identify the issuer. Should be a HTTP URL where a JSON-LD object describing the issuer can be retrieved.
- **`Type`** will be the array **`Profile`** along with **`Extension`** and **`SignatoryExtension`** if the profile is for an entity which will be providing digital signatures.
- **`Name`** the name by which the issuer is known
- optional **`Email`** containing the address where this issuer can be reached
- optional **`URL`** pointing to a HTTP accessible profile page or homepage of the issuer
- optional **`Description`** short text describing the issuing person or organisation
- **`PublicKey`** containing the public key of the Issuer/Assessor/Trainer which must be embedded inside the assertion if the credential is signed.
- optional **`RevocationList`** containing a HTTP URL of the list of signed badges issued by this issuer which have been revoked

Alignment Object

The OpenBadges v2 **`AlignmentObject`** is used to link a **`BadgeClass`** or an item of **`Evidence`** to an academic standard.

- **`targetName`** which identifies the name of target standard
- **`targetURL`** a HTTP endpoint which uniquely identifies the standard where a description of the standard can be found. Ideally should be a JSON-LD object describing the standard.
- **`targetFramework`** a string describing the name of the standards framework
- **`targetCode`** a string containing a code within target framework for the aligned standard

Evidence

The **`Evidence`** class from OpenBadges v2 spec is used to describe evidence in support of a credential.

- **`Id`** which uniquely identifies the evidence from the **`urn:uuid`** namespace or a HTTP URL of a webpage which presents the evidence
- **`Type`** containing the string Evidence. Additional types Extension and **`AssessedEvidence`** will be added when using **`AssessedEvidence`**
- optional textual **`Narrative`** which describes the evidence
- optional text **`Name`** of the evidence
- optional longer text **`Description`** of the evidence
- optional text **`Genre`** which describes the type of evidence, such as Certificate, Painting, Artefact, Medal, Video, Image. May be omitted in favour of the **`assessment`** property in the context of **`AssessedEvidence`**
- optional text **`Audience`** detailing the audience for which the evidence is presented
- optional **`Alignment`** containing details of alignment to educational standards.

TrainingEvidence

The **`TrainingEvidence`** class is used to add training specific properties to an item of **`Evidence.`**

- **`Type`** should an array containing **`Evidence, Extension`**, **`TrainingEvidence`**
- **`Subject(s)`** the subject(s) of the evidence
- **`TrainedBy(s)`** HTTP URL identifier to a JSON-LD object or an embedded profile of the person or organisation entity(s) which provided training to the recipient of the credential. The **`@id`** of the profile should either be a URI which uniquely identifies the training entity or should be a HTTP URL from which a more complete JSON-LD object can be retrieved
- optional **`Duration`** containing **`startDate`** and **`endDate`** specifying the duration of the course
- optional **`Session`** containing a string which describes which session of the course the recipient completed

AssessedEvidence

The **`AssessedEvidence`** class is an extension to the evidence class which adds additional fields for assessors and signatures.

- **`Type`** should an array containing **`Evidence, Extension`**, **`AssessedEvidence`**
- **`Subject(s)`** the subject(s) of the evidence
- **`Assessment`** conducted which elicited the evidence
- **`AssessedBy`** HTTP URL identifier to a JSON-LD object or an embedded profile of the individual(s) or organisation(s) who assessed the competency. The **`@id`** of the profile should either be a URI which uniquely identifies the training entity or should be a HTTP URL from which a more complete JSON-LD object can be retrieved
- **`AssessedOn`** the date when the assessment was conducted

- optional **`Signature`** of the assessor(s)

If assessors are not ready and able to submit individually signed evidence, the assessor's signature on the **`AssessedEvidence`** may be omitted. It is then the responsibility of the issuer to ensure the authenticity of the evidence submitted by the assessor through other available channels. Over time, migrating the assessment ecosystem to standardise on individually signed items of evidence will increase the overall level of trust in the certification process and will increase the utility of certificates in the employment process.

Assessment

`Assessments` can vary based on the nature of the assessment carried out. The essential fields of an assessment are

- **`Type`** of the assessment will be marks, percentage, grade, rank or some other means of assessing the subject
- **`Value`** of the assessment based on the type indicated.

Specific types of assessments may add additional properties to the Assessment object. Each new property must be published via JSON-LD documents which describe the schema and means of validation for the property.

To illustrate, a *schema for a marks-based assessment* is described:

- **`MinValue`** The minimum score which could have been achieved
- **`MaxValue`** The maximum score which could have been achieved
- **`PassValue`** The passing score for the assessment

Signatory Extension

`SignatoryExtension` extends the OpenBadges v2 **`IdentityObject`** to add the following properties.

- optional **`Designation`** of the signatory
- optional **`Image`** containing a HTTP URL or a data URI for an image associated with the signatory
- optional **`PublicKey`** object containing the public key of the signatory as defined by the LinkedDataSignatures specification[9]. Required if the signatory will affix their digital signature to the credential.

Additional Properties

Since the objects are modeled using RDF principles, additional properties may be added to the objects without affecting consumers. All consumers of credential data must be able to accept objects containing additional properties beyond the ones described as part of the credentials standard.

[9] https://w3c-dvcg.github.io/ld-signatures/#linked-data-signature-overview

RDF properties define a set of classes which are in the domain of a property[10]. When adding a new property to an object, the appropriate domain class must be include in the **`@type`** array.

Examples

Ecosystem

The examples below are based on a proposed ecosystem of issuers and assessors whose electronic profiles are available for consumption and validation. The ecosystem comprises:

1. A registry of certificate issuers maintained by the regulator. When an issuer commences issuing certificates, the issuer's profile is stored in the registry. This registry of profiles is accessed via HTTPS protocol and is protected by a SSL certificate under the control of the regulator. Hence all information about issuers which is retrieved from the registry is trusted.
    a. In the examples below, this registry is assumed to be at the address: **`https://certs.example.gov`**
2. Issuers who publish their public keys and certificate metadata at HTTPS endpoints. The public keys of the issuers MUST be referenced from their profile in the registry.
    a. In the examples below, we assume an example ITI which maintains its keys and certificate metadata at the addresses: **`https://example.iti.org/keys`** and **`https://example.iti.org/certs`** respectively
3. Assessors in the ecosystem publish their own profiles, which are accessed via HTTPS endpoints.
    a. In the examples below, we assume an assessor which maintains its profile at **`https://example.assessor.org/about.json`**
4. Assessors also publish public keys and records of assessments conducted at HTTPS endpoints. The public keys of the assessors must be referenced from their published profile.
    a. In the examples below, we assume the assessor publishes keys at **`https://example.assessor.org/keys/...`**

Ad-Hoc Certificate

A simple example of an assertion which certifies that a person has participated in an event.

[10] https://www.w3.org/TR/rdf-schema/#ch_properties

Fig 3: Ad-hoc Participation Certificate

Attributes

1. The `@id` of the assertion is a URL where this credential can be downloaded for verification.
2. Issued to a recipient identified by a hashed email.
3. Participation badge class is identified by a urn URI and embedded in the assertion.
4. The badge image is embedded into the certificate as a data URI representing a base64-encoded PNG image.
5. Issuer is identified by an `@id` which is a tag URI which uses the issuer's domain name.
6. The assertion employs hosted verification and is thus not signed.

```
{
  "@context": {
    "ob": "https://w3id.org/openbadges/v2",
    "oc": "https://certs.example.gov/opencerts/v1",
  },
  "@id": "https://www.example.com/certs/2018/9900001234.json",
  "@type": "ob:Assertion",
  "recipient": {                                <<<<<        2. Recipient identity
    "@type": "email",
    "hashed": "true",
    "identity": "sha256$bdeffdadbd28657adcead3825fdb23875dab8e928ad8d68f6",
    "salt": "bluewater",
    "name": "<Name of the recipient>",
  },
  "badge": {                                    <<<<<        3. BadgeClass inline
    "@id": "urn:uuid:ec58b28e-a6ab-49c2-a24d-ebefa02476cd",
    "@type": "ob:BadgeClass",
    "name": "Certificate of Participation",
    "description": "Content Marketing Course",
    "issuer": {
```

```
      "@type": ["ob:Profile", "oc:IssuerProfile"],
      "@id": "tag:example.com,2009-11-28:#company.json"5. Issuer identity
      "name": "Example Training Corp",
      "image": "https://www.example.com/images/logo.png",
      "email": "certificates@example.com",
    },
  },
  "issuedOn": "2018-08-11T09:27:30.613UTC",
  "narrative": "Issued for participating in Content Marketing Course in
association with Partner Marketing Solutions",
  "verification": {                        <<<<<        6. Hosted verification
    "@type": "hosted"
  }
  "oc:signatory": [{                  <<<<<                Official signatory
    "@type": ["oc:composite", "ob:Extension", "oc:SignatoryExtension"],
    "oc:components": [{
      "@type": "oc:name",
      "annotation": "FATHER",
      "identity": "<signatory's father's name>"
    }, {
      "@type": "oc:photo",
      "identity": "<base64 jpg image>"
    }],
    "name": "<Name of signatory>",
    "image": "https://example.com/p/ceo/sign-image.jpg",
    "oc:designation": "CEO, Example Training Corp",
  }, {                                <<<<<                Official signatory
    "@type": ["oc:urn", "ob:Extension", "oc:SignatoryExtension"],
    "name": "<Name of signatory>",
    "image": "https://example2.com/edb/l:dir/mkt/sign-image.jpg",
    "identity": "urn:in.gov.eci.voter:<Voter #>",
    "oc:designation": "Director, Partner Marketing Solutions",
}
```

Table 2: JSON-LD representation of an ad-hoc certificate

Course Completion

This example details the data added to a certificate for completion of a course which aligns to more than one academic standard. The BadgeClass contains details of the alignments to the target standards. The evidence for the certificate is represented in the form of the grade received for the course. While this example includes evidence aligned to one QP of the NSQF framework, the issuer is free to include multiple such items which detail the recipient's performance across multiple QPs if such granularity is so desired.

Fig 4: Sample Course Completion Certificate

Attributes

1. In the illustration, the recipient is identified by name, father's name and the last digits of an Aadhaar number.
2. The course is aligned to a NSQF qualification which in turn comprises of multiple NOS standards
3. The certificate has an issuer and an assessor institution
4. Though the illustration does not contain this, the evidence below encodes the grade received for the course
5. The signatory of the certificate is the CEO of the Automotive Skills Development Council

```
{
  "@context": {
    "sec": "https://w3id.org/security#",
    "ob": "https://w3id.org/openbadges/v2",
    "oc": "https://ncvet.gov.in/opencerts/v1",
  },
  "@id": "urn:uuid:1c0af19b-df85-42f3-9441-8a390b6c1589",
  "@type": ["ob:Assertion", "ob:Extension", "oc:CertificateExtension"],
  "recipient": {
    "$ref": "#/evidence/0/subject",  <<<<<     1. Reference to evidence subject
  },
  "badge": {
    "@id": "https://example.pasdc.org/certs/courses/ASCL3",
    "@type": "ob:BadgeClass",
    "name": "Automotive Service Technician Course",
    "description": " ... ",
    "image": "<base64-encoded-png-data>",
    "criteria": {
      "@type": "ob:Criteria",
      "narrative": "Successfully cleared assessment for role of Automotive
```

```
Service Technician",
    },
    "issuer": {                          <<<<<        3a. Issuer identified by URL
      "@type": ["ob:Profile", "ob:Extension", "ob:SignatoryExtension"],
      "@id": "https://certs.example.gov/o/pasdc/0781ABCDEAC191",
      "name": "Partner & Associate Skills Development Corporation",
      "publicKey": {
        "@id": "https://example.pasdc.org/keys/1",
        "@type": "sec:Key",
        "owner": "https://certs.example.gov/o/pasdc/0781ABCDEAC191",
        "publicKeyPem": "-----BEGIN PUBLIC KEY-----\n... ... ... ...
...\n-----END PUBLIC KEY-----\n",
      }
    },
    "alignment": [{                      <<<<<        2. Alignment to standards
      "targetName": "Automotive Service Technician - Level 3",
      "targetURL": "https://www.nqr.gov.in/ASC/Q1401",
      "targetDescription": " ... ",
      "targetFramework": "NSQF",
      "targetCode": "ASC/Q1401"
    }, {
      "targetName": " Assist in vehicle service and maintenance",
      "targetURL": "https://www.nqr.gov.in/ASC/N1401",
      "targetDescription": " ... ",
      "targetFramework": "NOS",
      "targetCode": "ASC/N1401"
    }, {
      "targetName": "Plan and organise work to meet expected outcomes",
      "targetURL": "https://www.nqr.gov.in/ASC/N0001",
      "targetDescription": " ... ",
      "targetFramework": "NOS",
      "targetCode": "ASC/N0001"
    }, {
      ...
    }]
  },
  "issuedOn": "2018-10-29T10:21:43.087UTC",
  "image": "https://example.pasdc.org/certs/1c0af19b8a390b6c1589.png",
  "training": {
    "@type": ["ob:Extension", "oc:TrainingExtension"],
    "trainer": {
      "@type": "ob:Profile",
      "@id": "https://traininer.example.edu/about.json",
      "name": "Partner Training Institute",
    }
    "duration": {
      "startDate": "2018-07-25",
      "endDate": "2018-10-22",
    },
    "session": "2018 Aug-Oct Batch #3",
  },
  "evidence": [{
    "@type": ["ob:Evidence", "ob:Extension", "oc:AssessedEvidence"],
    "@id": "urn:uuid:02644c88-d2b7-41ef-a78c-6adf7fbdb268",
    "subject": {
      "@type": "composite",              <<<<<  1. Identified by composite identity
      "components": [{
```

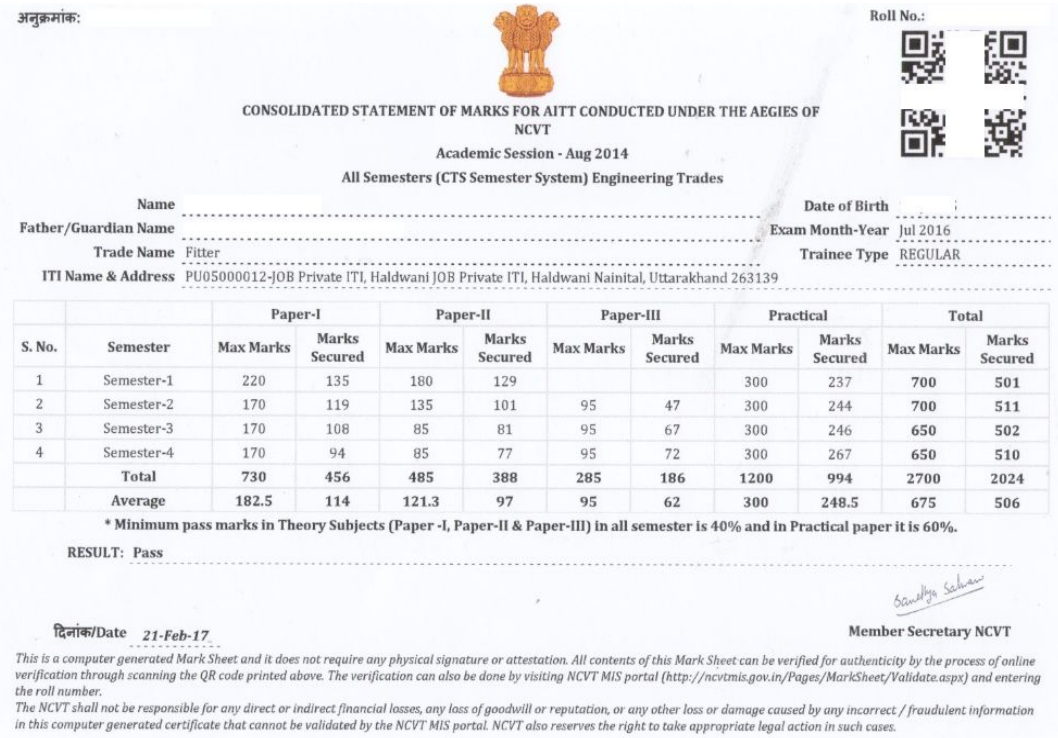
```
        "@type": "name",
        "annotation": "FATHER",
        "identity": "<Name of father>"
      }, {
        "@type": "oc:urn",
        "annotation": "MASKED",
        "identity": "urn:in.gov.uidai.aadhaar:XXXX-XXXXXXXX-2437"
      }],
      "name": "<Name of recipient>",
    },
    "name": "Course Grade",
    "assessment": {                  <<<<<        4. Representing the grade
      "@type": "oc:grade",           <<<<<
      "value": "B+",
      "maxValue": "A+",
      "related": {
        "@type": ["oc:equivalent", "oc:percentage"],
        "value": "72",
      }
    },
    "assessedBy": {                  <<<<<        3b. Identity of the assessor
      "@type": "ob:Profile",
      "@id": "https://certs.example.gov/o/acmeauto/0781ABCDEAC191",
      "name": "Acme Automotive Assessments Institute",
      "publicKey": {
        "@id": "https://example.assessor.org/keys/1",
        "@type": "sec:Key",
        "owner": "https://certs.example.gov/o/acmeauto/0781ABCDEAC191",
        "publicKeyPem": "-----BEGIN PUBLIC KEY-----\n... ... ... ...
...\n-----END PUBLIC KEY-----\n",
      }
    },
    "assessedOn": "2018-10-19T6:30:00Z",
    "signature": {
      "@type": "sec:LinkedDataSignature2015",
      "creator": "https://example.assessor.org/keys/1",
      "created": "2018-10-23T20:21:34Z",
      "signatureValue": "LTQzNTVVmMm3ODM3QzNmtlYzIZD34GIyZGk="
    }
  }],
  "verification": {
    "@type": "sec:LinkedDataSignatures"
  },
  "oc:signatory": {                  <<<<<        5. Official signatory
    "@type": ["oc:urn", "ob:Extension", "oc:SignatoryExtension"],
    "identity": "urn:in.gov.ka-dot.dl:<KA DL #>",
    "name": "<Name of signatory>",
    "image": "https://example.asdc.org/p/ceo/sign-image.jpg",
    "oc:designation": "CEO, Automotive Skills Development Council",
  },
  "oc:printUri": "<jpg-data>",
  "signature": [{
    "@type": "sec:LinkedDataSignature2015",
    "creator": "https://example.pasdc.org/keys/1",
    "created": "2018-10-23T10:21:40.817UTC",
    "signatureValue": "OGQzNGVkMzVmMmQ3ODIyYzI4ZDY3NjI4NTIyZTk="
  }],
```

```
}
```

Table 3: JSON-LD representation of a course completion certificate

Graduation Marks Certificate

In this example we consider a graduation marks certificate which describes a student's performance in multiple subjects as part of a larger achievement.



अनुक्रमांक: Roll No.:

CONSOLIDATED STATEMENT OF MARKS FOR AITT CONDUCTED UNDER THE AEGIES OF NCVT

Academic Session - Aug 2014

All Semesters (CTS Semester System) Engineering Trades

Name ..... Date of Birth .....

Father/Guardian Name ..... Exam Month-Year Jul 2016

Trade Name Fitter Trainee Type REGULAR

ITI Name & Address PU05000012-JOB Private ITI, Haldwani JOB Private ITI, Haldwani Nainital, Uttarakhand 263139

		Paper-I		Paper-II		Paper-III		Practical		Total	
S. No.	Semester	Max Marks	Marks Secured	Max Marks	Marks Secured	Max Marks	Marks Secured	Max Marks	Marks Secured	Max Marks	Marks Secured
1	Semester-1	220	135	180	129			300	237	700	501
2	Semester-2	170	119	135	101	95	47	300	244	700	511
3	Semester-3	170	108	85	81	95	67	300	246	650	502
4	Semester-4	170	94	85	77	95	72	300	267	650	510
	Total	730	456	485	388	285	186	1200	994	2700	2024
	Average	182.5	114	121.3	97	95	62	300	248.5	675	506

* Minimum pass marks in Theory Subjects (Paper -I, Paper-II & Paper-III) in all semester is 40% and in Practical paper it is 60%.

RESULT: Pass

दिनांक/Date 21-Feb-17

Member Secretary NCVT

This is a computer generated Mark Sheet and it does not require any physical signature or attestation. All contents of this Mark Sheet can be verified for authenticity by the process of online verification through scanning the QR code printed above. The verification can also be done by visiting NCVT MIS portal (http://ncvtmis.gov.in/Pages/MarkSheet/Validate.aspx) and entering the roll number.

The NCVT shall not be responsible for any direct or indirect financial losses, any loss of goodwill or reputation, or any other loss or damage caused by any incorrect / fraudulent information in this computer generated certificate that cannot be validated by the NCVT MIS portal. NCVT also reserves the right to take appropriate legal action in such cases.

Fig 5: Sample Consolidated Marksheet

For brevity, previously described objects (**`BadgeClass`** and **`Assessor`**) have been represented via their URIs, and repetitive elements have been truncated. For maximum portability and permanence, the assertion should have all these entity URIs dereferenced and their JSON-LD representations embedded into the credential.

Attributes

1. Multiple items of evidence per subject are linked into a single certificate
2. Evidence is described using **`alignmentObject`** which link to specific items in the curriculum.
3. PrintURI embeds a printable document for the assertion into the JSON-LD object.

```
{
  "@context": {
    "sec": "https://w3id.org/security/v1",
    "ob": "https://w3id.org/openbadges/v2",
    "oc": "https://ncvet.gov.in/opencerts/v1",
  },
  "@id": "urn:uuid:1c0af19b-df85-42f3-9441-8a390b6c1589",
  "@type": ["ob:Assertion", "ob:Extension", "oc:CertificateExtension"],
  "recipient": {
    "@type": "composite",
```

```
    "components": [{
      "@type": "name",
      "annotation": "FATHER",
      "identity": "<Name of father>",
    }, {
      "@type": "oc:dob",
      "identity": "<DOB of recipient>",
    }, {
      "@type": "oc:urn",
      "identity": "urn:in.gov.msde.dgt-rollno:<Roll # of the recipient>",
    }],
    "name": "<Name of recipient>",
  },
  "badge": "https://example.examboard.org/certs/csma.json",
  "issuedOn": "2017-02-21T10:21:43.087UTC",
  "image": "https://example.examboard.org/certs/1c0af19b8a390b6c1589.png",
  "narrative": "Passed",
  "evidence": [{                          <<<<<          1. Multiple evidence
    "@type": ["ob:Evidence", "ob:Extension", "oc:AssessedEvidence"],
    "@id": "urn:uuid:02644c88-d2b7-41ef-a78c-6adf7fbdb268",
    "subject": {
      ... same as recipient above ...
    },
    "name": "Semester 1 Mathematics",
    "assessment": {
      "@type": "marks",
      "value": "135",
      "maxValue": "220",
      "minValue": "0",
      "passValue": "88",
    },
    "assessedBy": "https://example.examboard.org/assessor.json",
    "assessedOn": "2015-12-22T6:30:00Z",
    "alignment": {                        <<<<<          2. Evidence descriptors
      "targetName": "Semester 1 Mathematics",
      "targetURL": "https://example.examboard.org/fitter/maths/s1",
      "targetDescription": " ... ",
      "targetFramework": "Examboard Semester Curriculum",
      "targetCode": "MATHS/S1"
    },
    "signature": {
      ... optional signature of the assessor ...
    }
  }, ..., {
    "@type": ["ob:Evidence", "ob:Extension", "oc:AssessedEvidence"],
    "@id": "urn:uuid:4ccc8f7b-30a4-466f-a50a-aef08378ae65",
    "subject": {
      ...
    },
    "name": "Semester 4 Practical",
    "assessment": {
      "@type": "marks",
      "value": "267",
      "maxValue": "300",
      "minValue": "0",
      "passValue": "180",
    },
    "assessedBy": "https://example.schoolboard.org/assessor.json",
```

```
    "assessedOn": "2017-02-19T06:30:00Z",
    "alignment": {
      "targetName": "Semester 4 Practical",
      "targetURL": "https://example.examboard.org/fitter/practical/s4",
      "targetDescription": " ... ",
      "targetFramework": "Examboard Semester Curriculum",
      "targetCode": "PRAC/S4"
    },
  }, {
    "@type": ["ob:Evidence", "ob:Extension", "oc:AssessedEvidence"],
    "@id": "urn:uuid:4ccc8f7b-30a4-466f-a50a-aef08378ae65",
    "subject": {
      ...
    },
    "name": "Semester 4 Total",
    "assessment": {
      "@type": "marks",
      "value": "510",
      "maxValue": "650",
      "minValue": "0"
    },
    "assessedBy": "https://example.schoolboard.org/assessor.json",
    "assessedOn": "2017-02-19T06:30:00Z",
    "alignment": {
      "targetName": "Semester 4 Total",
      "targetURL": "https://example.examboard.org/fitter/total/s4",
      "targetDescription": " ... ",
      "targetFramework": "Examboard Semester Curriculum",
      "targetCode": "TOT/S4"
    },
  }],
  "verification": {
    "@type": "sec:LinkedDataSignatures"
  },
  "oc:signatory": {                    <<<<<          4. Official signatory
    "@type": ["oc:urn", "ob:Extension", "oc:SignatoryExtension"],
    "name": "<Name of signatory>",
    "image": "https://example.ncvet.org/p/secretary-sign-image.jpg",
    "identity": "urn:in-dl-dl:<DL DL #>",
    "oc:designation": "Member Secretary, NCVT",
  },
  "oc:printUri": "data:application/pdf;base64,<pdf-data>",          <<<<<
                                        3. Printable representation
  "signature": [{
    ... signature of the issuer ...
  }],
}
```

Table 4: JSON-LD representation of a graduation marks certificate

University Degree

A university degree certificate is awarded after completion of the requirements for a degree. Such a certificate may reference other credentials as evidence. When a certificate references another certificate, the reference is encoded as a JSON reference object. The certificate below references each semester in the curriculum as evidence towards completing a degree. Other degree certificates may choose alternate evidence which is

accumulated towards a degree. A certificate may freely mix **`AssessedEvidence`** and **`Evidence`** which references other certificates in its **`evidence`** array.

Attributes

1. Evidence links to other certificates which are referenced by **`urn:uuid`** URIs. The referenced certificates are not embedded into the document. However, **`AlignmentObjects`** provide consumers with information about the content of the referenced certificate.
2. The **`genre`** field of the Evidence is the string certificate.

```
{
  "@context": {
    "sec": "https://w3id.org/security/v1",
    "ob": "https://w3id.org/openbadges/v2",
    "oc": "https://ncvet.gov.in/opencerts/v1",
  },
  "@id": "urn:uuid:1c0af19b-df85-42f3-9441-8a390b6c1589",
  "@type": ["ob:Assertion", "ob:Extension", "oc:CertificateExtension"],
  "recipient": {
    "@type": "oc:composite",
    "components": [{
      "@type": "oc:urn",
      "identity": "urn:in.gov.eci.voterid:<Recipient voter id>",
    }, {
      "@type": "oc:photo",
      "identity": "<... base64 encoded image ...>",
    }],
    "name": "<Name of recipient>",
  },
  "badge": "https://example.university.org/certs/degree/bvoc.json",
  "issuedOn": "2019-05-21T10:21:43.087UTC",
  "image": "https://example.university.org/certs/1c0af19b8a390b6c1589.png",
  "narrative": "<Recipient name> has successfully completed 6 (six) semesters
of B. Voc curriculum",
  "evidence": [{                    <<<<<     1. Evidence links to certificates
    "@type": "ob:Evidence",
    "@id": {"$ref": "urn:uuid:8db760b1-1348-4edd-8dfe-2c29799de4b2"},
    "name": "Semester 1",
    "genre": "certificate",         <<<<<          2. Genre is a certificate
    "alignment": {
      "targetName": "Semester 1",
      "targetURL": "https://example.university.org/certs/bvoc/semesters/1",
      "targetDescription": " ... ",
      "targetFramework": "B. Voc Semster Curriculum",
      "targetCode": "BVOC/S/1"
    },
  }, ... {
    "@type": "ob:Evidence",
    "@id": {"ref": "urn:uuid:0b3b0c30-c4f0-4075-b663-970cf9768cf0",
    "name": "Semester 6 ",
    "genre": "certificate",
    "alignment": {
      "targetName": "Semester 6",
      "targetURL": "https://example.university.org/certs/bvoc/semsesters/6",
      "targetDescription": " ... ",
```

```
      "targetFramework": "B. Voc Semester Curriculum",
      "targetCode": "BVOC/S/6"
    },
  }],
  "verification": {
    "@type": "sec:LinkedDataSignatures"
  },
  "oc:signatory": {
    "@type": ["oc:urn", "ob:Extension", "oc:SignatoryExtension"],
    "identity": "urn:in-dl-dl:<Delhi DL #>",
    "name": "<Name of signatory>",
    "oc:image": "https://example.ncvet.org/p/secretary-sign-image.jpg",
    "oc:designation": "Member Secretary, NCVT",
  },
  "oc:printUri": "<png-data>",
  "signature": [{
    ... signature of the issuer ...
  }],
}
```

Table 5: JSON-LD representation of a University degree certificates

Recognition of Performance

Building on top of the examples, we inspect a certificate of appreciation which recognises the standard of the recipient's performance. In this example the recipient of the certificate is a training institute.

Fig 6: Sample Certificate of Merit/Rank/Appreciation

Attributes

1. The credential's document identifier is a tag URI where the ID namespace combines the domain of the issuer with the registration date and adds a id-type (dgt.certificate).

2. Since performance may be awarded systematically, a hosted badge class is created and identified by its URL (the class is also embedded into the assertion).
3. Issuer is identified by a HTTP URL, along with essential properties of **`name`** and **`publicKey`**. The URL should return a JSON-LD object which can contain additional properties of the issuer (see description of the Issuer Profile).
4. A signatory to the certificate is added.
5. The certificate is signed with the issuer's private key.

```
{
  "@context": {
    "sec": "https://w3id.org/security/v1",
    "ob": "https://w3id.org/openbadges/v2",
    "oc": "https://certs.example.gov/opencerts/v1",
  },
  "@id": "tag:msde.gov.in,2015-02-27:dgt.certificate/1800122349",
  "@type": ["ob:Assertion", "ob:Extension", "oc:CertificateExtension"],
  "recipient": {
    "$ref": "#/evidence/subject"
  },
  "badge": {                              <<<<<       1. Hosted BadgeClass, URL ID
    "@id": "https://dgt.example.gov.in/certs/iti/grading/appreciate",
    "@type": "ob:BadgeClass",
    "name": "Certificate of Appreciation in National Level ITI Grading",
    "description": " ... ",
    "image": "<base64-encoded-png-data>",
    "criteria": {
      "@type": "ob:Criteria",
      "narrative": "For exhibiting outstanding performance"
    },
    "issuer": {                           <<<<<     2. Issuer with URL ID, publicKey
      "@type": "ob:Profile",
      "@id": "https://certs.example.gov/o/dgt/HJ5327VB1247G",
      "name": "Ministry of Skill Development and Entrepreneurship, Directorate
General of Training",
      "publicKey": {
        "@id": "https://dgt.example.gov.in/keys/issuer.json",
        "@type": "sec:Key",
        "owner": "https://certs.example.gov/o/dgt/HJ5327VB1247G",
        "publicKeyPem": "-----BEGIN PUBLIC KEY-----\n... ... ... ...
...\n-----END PUBLIC KEY-----\n",
      }
    },
  },
  "issuedOn": "2018-09-05T10:21:43.087UTC",
  "validFrom": "2018-09-01",
  "expires": "2020-06",
  "evidence": {                           <<<<<       Unsigned AssessedEvidence
    "@type": ["ob:Evidence", "ob:Extension", "oc:AssessedEvidence"],
    "@id": "urn:uuid:02644c88-d2b7-41ef-a78c-6adf7fbdb268",
    "subject": {
      "@type": "oc:urn",
      "identity": "urn:in.gov.gstn.id:Z000000000000001",
      "name": "Government Industrial Training Institute, Salboni",
    },
```

```
    "description": "Rank in National ITI Grading",
    "assessment": {
      "@type": "rank",
      "value": "8",
      "maxValue": "1"
    },
    "assessedBy": "https://dgt.example.gov.in/iti-assessor.json",
    "assessedOn": "2018-08-19T6:30:00Z",
  },
  "verification": {
    "@type": "sec:LinkedDataSignatures"
  },
  "signatory": {                        <<<<<          3. Official signatory
    "@type": ["oc:urn", "ob:Extension", "oc:SignatoryExtension"],
    "name": "<Name of signatory>",
    "identity": "urn:in.gov.msde.dgt-employee-id:GITI2D37A483ADJ452",
    "designation": "Director General (Training)",
  },
  "signature": [{                       <<<<<          4. Issuer's signatures
    "@type": "sec:LinkedDataSignature2015",
    "creator": "https://dgt.example.gov.in/keys/issuer.json",
    "created": "2018-10-23T20:21:34Z",
    "signatureValue": "OGQzNGVkMzVmMmQ3ODIyYzI4ZDY3NjI4NTIyZTk="
  }],
}
```

Table 6: JSON-LD representation of a certificate of merit

Issuing Credentials

Credentials are issued by creating an assertion using the items of data described below. They reference the structure of the assertions in the examples described above

1. **Assertion ID**: Each assertion must be given a unique id. In the example above, the unique id is a web URL where JSON-LD representation of the certificate can be retrieved. Where URLs cannot be generated to store representations of the certificate, a URI in the **`urn:uuid`** namespace can be generated.
2. **Issuer ID**: In the first example given above, the issuer's identity (**`badge.issuer.identifier`**) is represented using a tag URI. This shows an example of adding an identifier to a resource object where a HTTP URI is not available. The remaining examples identify the BadgeClass's issuer with an **`@id`** field which is a HTTP URI to a JSON-LD object containing the issuer's profile.
3. **Issuer Public Key**: the issuer's public key from the cryptographic key pair (**`badge.issuer.publicKey`**) is embedded into the certificate for verifiability.
4. **Recipient ID**: in the first example given above, the recipient is identified by a hash of the recipients email address (**`recipient.identity`**). The recipient may be identified by other strong identifiers. Where the information in the identifier is sensitive, the identifier can be salted and hashed for security
5. **Signed Evidence**: The issuer of the certificate collects evidence from the assessors of the trainee. The evidence may optionally be signed. The issuer must ensure that

   a. The subject of the evaluation conducted is the same person to whom the credentials are being issued. The issuer can do this by comparing the id of the evidence's subject (`evidence.subject.identity`) with the id of the recipient (`recipient.identity`).
   b. If the evidence is signed by the assessors, each evidence's signature can be verified by using the assessor's publicKey embedded in the evidence (`evidence.assessedBy.publicKey`).
6. **Signature**: Finally, the issuer uses their own private key to generate a signature for the assertion data and inserts that signature to the assertion. In the event that there is a signatory to the certificate who is also applying a digital signature, the unsigned assertion is signed using the signatory's private key as well. Both signatures are then included in the list of signatures for the certificate.

Signing Procedure for Assertions and Evidence

Signing a certificate is a process by which a one-way digest of the assertion object is computed and is then cryptographically signed (encryption) using the issuer's private key. The signature suite used will specify the digest and the cryptographic functions which are to be applied (suggestion is to use LinkedDataSignature2015[11]).

The private key used for signing must be maintained in a secure repository and should be transferred only via secure channels. During application usage the keys should be maintained in the secure area such as an HSM. The data to be signed using the private key should be sent for signing and the signed value returned.

Usage

Delivery and Storage of Credentials

Delivery mechanisms for credential documents can vary based on the context where they are issued. A certificate for an online course may be issued immediately in the browser, while an offline course with written exams may have an alternate method to deliver certificates. Furthermore, these mechanisms are dynamic and change over time. New modes of delivery may be developed and present methods may become obsolete. Similar considerations apply to the means of storing the certificate; the recipient may choose from multiple options available for securely storing the digital document.

To empower recipients with a choice of delivery and storage mechanisms and to ensure compatibility with future methods, delivery and storage is independent of the data in the credential. Thus we may use any means to identify the recipient of the credential and independently choose the way it is delivered and stored. Below we consider some well-established delivery and storage options as well as a few emerging technology options such as Blockcerts and uPort.

[11] https://w3c-dvcg.github.io/lds-rsa2018/

Email

One mode of delivery may be email. If the recipient's email address is known, and the recipient elects for email delivery, the certificate can be sent to the recipient's email address as an attachment. The recipient is then free to store using any solution available such as a cloud drive, offline storage etc.

Web

An alternate mode of delivery can be via a private URL which allows the recipient to download the JSON-LD credential. The recipient is then free to store using any solution available such as a cloud drive, thumb drive, offline storage etc. The private URL can be sent to the recipient via an SMS message to a mobile device, an email or any other communication channel available to the issuer and the recipient.

DigiLocker and other National Repositories

Certificates can be stored in the cloud using services such as DigiLocker or the National Academic Depository. The machine-readable format has a printable representation embedded in it which can be used by client applications to render previews of the credential.

Blockchain

Blockchain applications may also be used to deliver the certificate to the recipient. The issuer can use a blockchain certificate publishing protocol such as Blockcerts[12] which is also compatible with OpenBadges to store certificates on a blockchain as part of transaction metadata. Recipients can then retrieve the certificate from the blockchain and store as per their choice.

Wallet

Application developers can create mobile wallets for storing credentials. The JSON-LD document containing the certificate can be imported into any number of such applications to manage accomplishments and credentials. The embedded printable representation may be used by wallet applications to render previews of the credential.

uPort

uPort is a toolkit for building distributed applications on top of the Ethereum blockchain. uPort applications can issue credentials to a user which are then attached to the user's profile. The certificate spec described above can be linked into a uPort message which is transported a JSON Web Token (JWT). The JWT contains signed-data as a **`claim`** which can be the assertion JSON object.

Verifying Authenticity of a Certificate

A certificate is authenticated along three dimensions.

First, verify that the certificate has been issued by the issuer.

[12] https://www.blockcerts.org

1. The certificate contains one or more signatures in the signature field. The signature is encrypted using the issuer's private key.
2. The signature is decrypted using the issuer's public key which returns the digest of the message.
    a. The issuer's public key details are available in **`badge.issuedBy.publicKey`**
3. The digest of the message is computed and compared with the decrypted digest.
4. If the two digests match, the signature is verified.

Second, verify that the certificate is optionally digitally signed by the signatory

1. The certificate may also contain the signatory's digital signature in the signature field. The signature is encrypted using the signatory's private key.
2. The signature can be verified as above using **`assertion.signatory.publicKey`**

Finally, if the certificate contains signed evidence, verify the assessor's signature in each item of evidence in certificate.

1. The credential contains one or more items of evidence which contain signatures in the **`evidence.signature`** field encrypted using the assessor's private key.
2. The signature can be verified as above using the assessor's public key available from **`evidence.assessedBy.publicKey`**
3. The subject of the evidence must be the same as the recipient of the certificate.

Note that when authenticating the physical certificate, downloading the machine readable version from the URL contained in it's QR code is a first step. If the URL resolution fails, it cannot be assumed that the certificate is invalid, the URL may be unavailable for many reasons. In this scenario, the authenticity of the physical certificate must be verified via other means.

Portability of Certificates

The accomplishments in the certificate identify standards, levels and criteria which are evaluated by an assessor. This allows the consumer of the certificate to compare competencies against desired skills using the same standard, level and criteria as a reference.

Permanence

To ensure that the authenticity of a certificate is verifiable without needing external data, the certificate payload carries the public keys required to verify signatures.

Key Management

Public keys for the signing authorities could be cloud-hosted by each signing body. For instance, each ITI could maintain its own public keys in the cloud where they can be accessed by anyone trying to verify a certificate issued by the ITI. However, if signing keys are cloud-hosted and the cloud location is embedded inside the certificate then any change in the location of the key will invalidate certificates. Issuers may or may not be able to maintain a permanent location for their keys metadata. This could be worked around by

either employing a key broker service which enables keys to be discovered after an issuer has changed its location or alternatively by a capable entity providing a secure repository of public keys for all issuers as an ecosystem service.

Summary

This document describes an electronic schema-based standard for describing credentials data in a machine-readable format (JSON-LD representation of RDFa) along with its printable human-friendly version to make credential exchange between digital agents open and reliable. Certificates issued according to this specification can be aligned to one or more educational standards. They are secured using digital signatures which employ asymmetric PKI and provide for severless certificate verification. This makes system more conducive for the diverse ecosystem in India where network connectivity may be absent. The serverless nature also eliminates the added costs of verification infrastructure which may be challenging to operate reliably at scale.

References

1. Architecture of the Web: https://www.w3.org/TR/webarch
2. OpenBadges v2: https://www.imsglobal.org/sites/default/files/Badges/OBv2p0Final/
3. OpenBadges examples: https://www.imsglobal.org/sites/default/files/Badges/OBv2p0Final/examples/
4. RDF: https://www.w3.org/RDF/
5. RDF Schema: https://www.w3.org/TR/rdf-schema
6. JSON-LD: https://www.w3.org/TR/json-ld/
7. Linked Data Platform: https://www.w3.org/TR/ldp/
8. RDFa: https://www.w3.org/TR/rdfa-primer/
9. Turtle: https://www.w3.org/TR/turtle/
10. Uniform Resource Names: https://tools.ietf.org/html/rfc8141
11. The Tag URI scheme: https://tools.ietf.org/html/rfc4151
12. Schema.org Full Vocabulary: https://schema.org/docs/full.html
13. Use of '@id' vs 'identifier': https://schema.org/docs/datamodel.html#identifierBg
14. Linked Data Signatures: https://w3c-dvcg.github.io/ld-signatures/
15. Digilocker: https://digilocker.gov.in/
16. National Academic Depository: http://nad.gov.in/
17. Blockcerts: https://www.blockcerts.org
18. uPort application development kit: https://www.uport.me/