Open Badges are Verifiable Credentials

A White Paper from Rebooting the Web of Trust VI

By Nate Otto (Concentric Sky) & Kim Hamilton Duffy (Learning Machine)

ABSTRACT

We identify use cases and requirements that connect threads of work happening in the Rebooting Web of Trust community around: educational achievement claims (particularly using the Open Badges vocabulary); use of decentralized identifiers (DIDs) within web services where educational claims circulate; and integrating blockchain-reliant verification layers. We illustrate each of these cases with a set of example documents and describe user stories for Open Badges ecosystem software in the roles of Issuer, Host/Backpack, Displayer, and Verifier that need to be implemented in order to enable the capabilities described.

INTRODUCTION

The Open Badges Specification is a vocabulary and set of protocols that describes credentials. The vocabulary can describe any achievement in terms of a common set of attributes and is most often used for educational or occupational credentials. At present in version 2.0, Open Badges defines two verification methods: `HostedBadge` (requiring resources hosted on HTTP in specific locations) and `SignedBadge` (using a JSON Web Signature, which references hosted Issuer Profile and CryptographicKey information).

The Blockcerts Open Badges Draft Extension introduced a verification method based on those used by Verifiable

PTBVENTURES

sovrin

Named Sponsors for the Rebooting the Web of Trust VI Design Workshop

Credentials for the specific use case of blockchain-anchored credentials. This paper expands that work and proposes a new option that can reside alongside existing Open Badges verification methods.

This paper also explores new capabilities that the Open Badges ecosystem could gain by adopting some of the emerging technologies and conventions from the Verifiable Credentials specification and its communities of implementers. It will inform the authors' development of working prototypes to demonstrate the viability and utility of these methods for publishing and circulating Open Badges.

MOTIVATION

In this section we provide the motivation for supporting Verifiable Credentials Specifications in Open Badges implementations. There are three key reasons for our work: these models are compatible in purpose; their data models are complementary; and there are benefits in alignment for the existing communities around both specifications.

Open Badges and Verifiable Credentials are compatible in purpose

Both Open Badges (OB) and Verifiable Credentials (VC) are capable of expressing a cryptographically verifiable statement about the subject, issuer, evidence, and status of a credential.

The Verifiable Credentials specification provides a lightweight structure for expressing a wide range of credentials — including driver's licenses or passports. A VC implementer chooses which schema/vocabulary to use, depending on the use case or domain.

The strength of Verifiable Credentials is its flexibility across a wide variety of use cases. At the same time, there is not yet general agreement on schema and vocabulary sets to use with Verifiable Credentials that will allow them to serve their varied purposes.

In comparison, Open Badges have been used in production deployments for nearly a decade. This has established fitness-of-purpose for real-world educational/occupational scenarios, and has led to a rich set of conventions and vocabularies. Successful deployments range from low to high stakes contexts, including informal recognition of a valuable contribution, completion of training, or completion of coursework or a university degrees.

Building on this success, the Open Badges community has driven related standards, such as stackable credentials leading to a larger goal ("Open Pathways", which could be used to represent the path from individual university courses to a degree).

Similar data models

There is significant cross-pollination in the development of Open Badges and Verifiable Credentials; accordingly, there is already some structural alignment: - Open Badge Endorsements employ the Verifiable Credentials structure. - Open Badges and Verifiable Credentials Issuer Profiles are aligned. - Open Badges and Verifiable Credentials both use JSON Linked Data (JSON-LD), allowing reuse among different contexts. They both allow the addition of terms beyond their own vocabulary, meaning components of each specification may be used in the other.

Blockcerts/Open Badge signing and verification processes use the same methods as Verifiable Credentials to establish the authenticity and integrity of claims. This is because the Blockcerts extension draft for Open Badges 2.0 uses the JSON-LD signatures/verification method (the same used by Verifiable Credentials) to anchor an Open Badge to a blockchain.

Benefits of continued alignment to both communities

Further alignment of Open Badges and Verifiable Credentials provides benefits to the communities and recipients: - Verifiable Credentials benefit from the rich expressiveness and vocabulary vetted by years of real-world Open Badges deployments and the ability to use these claims quickly across deployed services that already understand Open Badges. - Open Badges can participate in the upcoming Verifiable Credentials ecosystem, which benefits recipients by allowing their Open Badges to function as Verifiable Credentials (via alignment of structural and verification mechanisms). - Recipients may share their Open Badges just like any Verifiable Credential, taking advantage of potentially broad interoperability of these claims.

THE OPEN BADGES VOCABULARY

Understanding the alignment between these specifications is possible with a deeper examination of how Open Badges claims are described semantically. Open Badges defines three primary data classes, with one of each making up a valid badge awarded to a single recipient. The Open Badges Specification, currently in version 2.0, is published by the IMS Global Learning Consortium.

The *Issuer Profile* describes the issuer entity, whether that is an individual, an organization, or something else, like an autonomous actor.

The *BadgeClass* describes a particular achievement, the criteria that all recipients must pass in order to be awarded the badge. Each BadgeClass has an issuer, its creator.

The *Assertion* describes an instance of the achievement as it applies to a single recipient. It identifies which BadgeClass is awarded, and describes the recipient of the award using a single string-type identifier (such as an email address).

The Assertion does not currently identify its issuer directly, but only that the issuer of the BadgeClass may publish valid Assertions. The Open Badges validator(s) check to ensure the Assertion is either cryptographically signed with one of the Issuer's authorized keys or uses a hosted verification URI within the Issuer's allowed scope.

Members of the Open Badges community have long talked about use cases for Assertions with a different issuer than the issuer of their BadgeClass, but this feature has not yet been implemented in the specification.

THE VERIFIABLE CREDENTIALS SPECIFICATION

Members of the W3C and Rebooting Web of Trust community have been developing the idea of Verifiable Credentials (formerly Verifiable Claims) for several years, and it has now been picked up as official work of the W3C Verifiable Credentials Working Group. The current editor's draft of the VC Data Model describes that a *Credential* is a set of one or more *claims* about a *subject*. All of the claims made by traditional credentials such as driver's licenses may be made by a digital Verifiable Credential, and many novel claims may also be made. VCs become verifiable through *proofs*, such as cryptographic signatures.

IMPLEMENTATION, PRESENT AND FUTURE

The Open Badges Specification defines everything needed to create an ecosystem where credentials circulate. The core vocabulary allows issuers to describe their profiles and achievements (available and awarded) in JSON-LD. The verification protocols for hosted and signed badges describe how to establish that a particular Assertion is the valid expression of a badge award published by an Issuer Profile, while the Baking Specification describes a method by which Assertion data may be embedded within a PNG or SVG image file, which is portable across many file systems and publishable on websites.

There are dozens of implementers of Open Badges across the roles of Issuer, Host (commonly known as a "backpack"), and Displayer, including several platforms that are certified by IMS Global as compliant implementers of the 2.0 specification in these roles. Over 10 million credentials have been awarded as Open Badges.

Going forward from 2.0, community members are interested in increased flexibility to describe credentials, better information about how specific badges and issuers are situated in networks of trust, and a growing ecosystem of adopters with good complementary integration with other specifications. The natural focus of the Open Badges Specification is as *a vocabulary to describe defined achievements*. There are advantages to be gained in consistent implementation of a common vocabulary for achievement credentials.

Verifiable Credentials has a vibrant community of interested implementers, including several who are using draft versions of the specification in production systems. The Verifiable Credentials specification is a general purpose technology with a wide range of potential use cases. The specification's focus is not on a particular type of credential or specific set of terms and it does not have an inherent concept of a defined achievement like Open Badges does.

The strength for Verifiable Credentials as a specification is to enable innovation in different types of proofs, claims, and methods of identifying credential subjects. Examples of these efforts include the use of blockchain pointers in proofs and Decentralized Identifiers (DIDs) for identifying subjects.

There may be a clean fit between the focus of Open Badges as a controlled vocabulary for defined achievements and the innovation around proofs and subject identifiers occurring in the community of Verifiable Credentials implementers.

This paper illustrates how these technologies may move closer to full compatibility by making careful choices at the integration surface. Effectively, the Open Badges ecosystem can implement the Verifiable Credentials data model as an option alongside its existing hosted and signed delivery/verification methods with minimal changes to specification and software in the Issuer, Host, and Displayer roles.

Open Badges validation/verification software is most significantly affected, but the open source and shared nature of the official IMS Global Open Badges validator library means very few actual pieces of software must be updated in order to obtain wide-reaching availability of the techniques shown here across the Host and Displayer roles. Services in the Issuer Role have an easier and less critical adoption process, especially as long as existing verification methods remain in place as an option.

IMPLEMENTATION OPTIONS

We have identified two possible ways to connect the Open Badges and Verifiable Credentials specifications. We will circulate and compare the core options here and choose which to base our prototypes upon.

Option 1: A Verifiable Credential/Assertion claims that an entity holds an BadgeClass

The Verifiable Credentials Data model has moved to enable some "credential metadata" options that parallel attributes of Open Badges Assertions, including evidence. The core purpose of an Assertion is to describe the instance of an award, an issuer's claim that a particular recipient has met the criteria defined in a BadgeClass and to present the evidence related to that.

While Open Badges have a `badge` property that points to "the BadgeClass definition that is presently asserted" and a separate `recipient` property that serves to identify the recipient by some string-based identifier, such as an email or a DID, Verifiable Credentials have a `claim` that an issuer makes to describe attributes of a recipient. In the Verifiable Credentials context, the way to describe the Open Badges concept of "earning an instance of a BadgeClass definition" is to *claim* that "a recipient, identified by an IRI or by an attribute of a different type of string-type identifier, *holds* or has met the criteria of a BadgeClass definition".

Here is an example of a Verifiable Credential/Assertion that a recipient identified by a HTTP profile URI holds a certain BadgeClass. In this and future examples, `id` is an alias in the Open Badges and Verifiable Credentials context of the linked data subject identifier keyword, canonically `@id`. This requires the introduction of a new `holds` property in the Open Badges vocabulary in order to make the claim shown here "The entity identified as `https://example.com/profiles/bob` holds the BadgeClass 'Certificate of Accomplishment'".

```
{
    "@context": ["https://w3id.org/credentials/v1",
"https://w3id.org/openbadges/v2"],
    "id": "https://example.com/assertions/1001",
    "type": ["Credential", "Assertion"],
    "issuer": "https://example.com/profiles/alice",
    "issued": "2018-02-28T14:58:57.461422+00:00",
    "claim": {
        "id": "https://example.com/profiles/bob",
        "obi:holds": {
            "id": "https://example.com/badgeclasses/123",
            "type": "BadgeClass",
            "name": "Certificate of Accomplishment",
            "image": "data:image/png;base64,...",
            "description": "A badge describing great accomplishments",
            "criteria": {
                "narrative": "Perform tasks of valor and wit."
            },
            "issuer": {
                "type": "Profile",
                "id": "https://example.com/profiles/alice",
                "name": "Example Issuer",
                "url": "http://example.com",
                "email": "test@example.com"
            }
        }
    },
    "obi:evidence": {
        "id": "https://example.org/portfolios/25",
        "name": "Bob's Portfolio",
        "narrative": "Bob worked hard to develop a good portfolio",
        "genre": "ePortfolio"
    },
    "sec:proof": {
        "@graph": {
            "type": "sec:RsaSignature2018",
            "created": "2018-03-07T19:22:15Z",
            "creator": "https://example.com/profiles/alice/keys/1",
            "sec:jws":
```

```
"eyJhbGciOiJQUzI1NiIsImI2NCI6ZmFsc2UsImNyaXQiOlsiYjY0Il19..p9RDJqwzXyxfC69dMEz503_Z
Z_af1e_hV931dPlIdrofC6p2y_dcjjqIDysReJy6W_fnN_dZGVoXqFAg2OD_SmbDi5dNMOZILot-zJdDJCx
XWuwZtiCFlt29KfLmJs6me0bD5pU4RbknXDoyBhA8muMby8j1fUeBDo3Ienmzv5UlB3v0f0-w5l6-z_cswH
B_UXIlWw4EzcsmLvHzjB7TI76QLwq3KeVPSB3U9aM3o2Ejkq6Ygh5XxUGkXiZUQ5ungQ9Psy_VicjZyOc19
LoBPoiPxDHQodTrqCFNH2qCNhDc4lg2zE8S9KNlQhUUFatzkTN70s23fhWBMKz2a5DWgQ"
        }
    }
}
```

Values for the claim may vary in two meaningful cases:

- The issuer wishes to take advantage of the light anti-correlation mechanism enabled by Open Badges hashed recipient identifiers for a recipient who is identified by an `id` string.
- The issuer wishes to identify the recipient by a profile identifier property other than `id`, such as `email`. Use of hashed identifier may or may not be used.

In either case, the implementation option is to make an additional claim about the recipient. In addition to the above claim that "a recipient holds or has met the criteria of a BadgeClass definition", the issuer makes a claim that the recipient "holds an identifier of a particular type" using the Open Badges IdentityObject:

```
{
    ...
    "claim": {
        "holds": { "type": "BadgeClass" },
        "recipient": {
            "type": "email",
            "hashed": false,
            "identity": "testrecipient@example.com"
        }
    }
}
```

This example does not use any `id` at all for the credential's claim. This is nonstandard in the VC ecosystem if not disallowed. If an issuer desires to not omit `id` while still identifying the recipient via the IdentityObject, a one-time IRI may be generated, such as a randomly UUID, using the `urn:uuid:` scheme. An Open Badges verifier should assume that if a claim is found where `obi:holds` and `recipient` are claimed, the `recipient` IdentityObject takes precedence over the claim subject `id` or that both are valid ways to refer to the recipient.

Option 2: An Assertion is a Verifiable Credential Claim

The Verifiable Credentials Specification allows issuers to make a claim about a subject while remaining agnostic to the content of that claim. It is possible to use an Open Badges Assertion as the claim of a Verifiable Credential, signed with a Linked Data Signature. The ID for the key and its owner are modeled as an HTTP-resolvable document as in the examples on the jsonld-signatures library.

```
{
    "@context": "https://w3id.org/credentials/v1",
    "id": "https://some.university.edu/credentials/9732",
    "type": ["Credential", "OpenBadgeCredential"],
    "issuer": "https://example.com/i/alice",
    "issued": "2018-02-28T14:58:57.461422+00:00",
    "claim": [{
        "@context": "https://w3id.org/openbadges/v2",
        "id": "urn:uuid:437fc6ff-bb3c-4987-a4b7-be8661ff6f21",
        "type": "Assertion",
        "recipient": {
            "type": "email",
            "identity": "testrecipient@example.com",
            "hashed": false
        },
        "badge": {
            "type": "BadgeClass",
            "id": "urn:uuid:7aad3c57-3bfb-45ea-ae79-5a6023cc62e4",
            "name": "Certificate of Accomplishment",
            "image": "data:image/png;base64,...",
            "description": "A badge describing great accomplishments",
            "criteria": {
                "narrative": "Perform tasks of valor and wit."
            },
            "issuer": {
                "type": "Profile",
                "id": "https://example.com/profiles/alice",
                "name": "Example Issuer",
                "url": "http://example.com",
                "email": "test@example.com"
            }
        },
        "verification": {
            "type": "VerifiableClaim2018"
        }
    }],
    "sec:proof": {
        "@graph": {
            "type": "sec:RsaSignature2018",
            "created": "2018-03-07T19:22:15Z",
            "creator": "https://example.com/profiles/alice/keys/1",
            "sec:jws":
"eyJhbGciOiJQUzI1NiIsImI2NCI6ZmFsc2UsImNyaXQiOlsiYjY0Il19..p9RDJqwzXyxfC69dMEz503_Z
Z_af1e_hV931dPlIdrofC6p2y_dcjjqIDysReJy6W_fnN_dZGVoXqFAg2OD_SmbDi5dNMOZILot-zJdDJCx
XWuwZtiCFlt29KfLmJs6me0bD5pU4RbknXDoyBhA8muMby8j1fUeBDo3Ienmzv5UlB3v0f0-w5l6-z_cswH
B_UXIlWw4EzcsmLvHzjB7TI76QLwq3KeVPSB3U9aM3o2Ejkq6Ygh5XxUGkXiZUQ5ungQ9Psy_VicjZyOc19
LoBPoiPxDHQodTrqCFNH2qCNhDc4lg2zE8S9KNlQhUUFatzkTN70s23fhWBMKz2a5DWgQ"
        }
    }
}
```

The example above illustrates the following salient features:

1. The claim ID is the ID of the Assertion, not of the Recipient. Semantically, that means that the Verifiable Credential is effectively claiming "This Assertion exists with content..." instead of the Option 1 model of introducing a "holds" property making the claim about a recipient instead. For example, "The subject identified as `id` holds the Assertion with content".
2. The above approach is a simple option where we can preserve Open Badges' ability to "hash" the recipient identifier. When the recipient id has been hashed in this manner, the Verifiable Credential will likely not be usable in a wallet application intended for VCs. But because there will remain a strong market for Open Badges-specific tooling to serve recipients, this type of VC with a hashed subject identifier will likely remain useful as a light layer of personally identifiable information protection above a base VC.
3. There is some duplication in the data package around the issuer. Future relaxation of the Open Badges Specification may become possible when the Assertion issuer is defined elsewhere in the "envelope", but the relative cost of duplicating this data to achieve parity with the current Open Badges Specification is low and will minimize the work needed in Open Badges tools to implement this delivery method.

Comparing Options 1 and 2

Option 1 uses a recipient subject identifier in the same way as other claims, which is expected to increase compatibility with VC-native tooling. On the other hand, Option 2 requires significantly less work on the part of the Open Badges validator(s) to return a result graph that matches its existing format implemented based on Open Badges 2.0.

Option 2, in which the claim is roughly the same as the existing Assertion class, is a valid Verifiable Credential. But because the claim subject is the Assertion itself instead of the recipient, this is not assumed to be a very strong integration path that would allow Open Badges Verifiable Credentials to be circulated among general purpose Verifiable Credential "wallet" tools. These tools expect claims to be made about their users, not about entities that have relationships to their users.

AN ASSERTION AWARDED TO A RECIPIENT IDENTIFIED BY A DID

In the existing Open Badges ecosystem, recipients are identified by email address virtually all of the time. Open Badges 2.0 strengthened and clarified the ability to award to other types of identifiers, but Assertions far more often use email. An example implementation in an Assertion could identify a recipient like this:

```
"recipient": {
    "type": "email",
    "identity": "testrecipient@example.com",
    "hashed": false
}
```

The Open Badges Specification allows for several string-type attributes of an entity to be used as an Assertion's Recipient Profile Identifier Property.

Decentralized Identifiers (DIDs) are emerging as a new type of entity identifier that comes in an IRI string format. They have the ability to be deterministically "resolved" to a "DID Document". The resolved documents then describe the functionally useful aspects of its identity, such as the public keys and other authentication methods that can be used to connect a user in a web browser or a human sitting across the table from you to the DID identifier.

As Open Badges can be awarded to other string-type identifiers, like `@id` (aliased as `id` in the Open Badges & Verifiable Credentials contexts). We can identify a recipient in an Option 2-style Assertion like:

```
"recipient": {
    "identity": "did:example:recipient_did",
    "type": "id",
    "hashed": false
}
```

This is all that must be done to award an Assertion to a DID as recipient identifier. The verification that an assertion is awarded to the expected DID is already supported in the Open Badges Validator. The Specification can be clarified to add "id" to the list of identifier types "considered serviceable", but that will constitute a clarification rather than a change.

AN ASSERTION AWARDED BY AN ISSUER IDENTIFIED WITH A DID

The primary use case for a Decentralized Identifier (DID) string is to resolve it to a DID Document, which describes the entity; specifically, it has attributes like public keys that allow for the authentication of messages sent by and under the auspices of the entity that controls the DID.

Verifiable Credentials are one type of message that can be signed and delivered by a DID-identified entity, and Open Badges Verifiable Credentials are yet another type of Verifiable Credential that can be signed in the same way.

With this model in place, viewers of Open Badges (through Verifiers, Backpacks/Hosts and Displayers) can resolve a DID to a DID Document, access the authorized public keys associated with that DID, determine which authorized key was identified as the creator of a Credential signature, and verify the authenticity of that signature.

In the current Open Badges Specification, even when using the `SignedBadge` verification method, there are still at least two resources that must be hosted on HTTP(s) in order to be able to produce valid Open Badges Assertions: the Issuer Profile and the CryptographicKey (identified by the publicKey attribute of the Issuer Profile). Using a DID for the Issuer Profile `id` instead of an HTTP URI allows for open badge data to be entirely uncoupled from HTTP hosting dependencies.

Here is an example DID for an issuer: `did:example:issuer_did`. The part of the DID between the first and second `:` characters identifies the [resolver method](). The part after the second `:` is the [DID path](). If a service has implemented the algorithm specified by the particular resolver method used, it can consistently retrieve a DID document identified by the DID.

We now show an example DID document for the `did:example:issuer_did` issuer described above. An implementer of the `example` resolver method could have retrieved the DID document using the method specified by the method specified. For example, by fetching data from a specific blockchain and deriving the DID document from it.

```
{
  "@context": "https://w3id.org/did/v1",
  "id": "did:example:issuer_did",
  "publicKey": [{
    "id": "did:example:issuer_did#keys-1",
    "type": "RsaVerificationKey2018",
    "owner": "did:example:issuer_did",
    "publicKeyPem": "-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA2x3wWF6LY1pNxxTIUfPtHBGvPgcVvVDaoGpOjMk
9+QfcXBVzcEnIlYenI8WfvoNSNUbblmDyn5sqWg9KyQpcoJ2IAoemjTcb+Skw9PaS2KAYjCYq9pmnvGhmmX
Jjk1xuT3gevG8K9XGJ2MmuqdKQ4yzfPhD5kHdLxV9Y9VY2rgZJzSNL83Oz596tcFbA1QB0p8wj7xQpLYRu5
d1Mz+1qu1E8NM6HPgSBp54JHJF0yL3s39rGNbxmopwCq1Vw9E22ZnJpHQtc4nq4N3JksfVPeHhirC3eny0Y
S78Z6W7bGlVT+bf+T6r43Rq8kQ7N8hVLrGHc+NuHXm1JBIbpKwIDAQAB-----END PUBLIC
KEY-----\r\n"
  }],
  "authentication": [{
    "type": "RsaSignatureAuthentication2018",
    "publicKey": "did:example:issuer_did#keys-1"
  }],
  "service": [{
    "type": "ExampleService",
    "serviceEndpoint": "https://example.com/endpoint/8377464"
  }]
}
```

Within such a DID Document, one or more public keys can be identified. In the example above, a single RSA key is described. An Open Badges verifier that supports the required DID resolver method discovers keys that can be trusted to be under the control of the entity. The resolver then verifies that a particular Open Badges Verifiable Claim is signed by a key pair belonging to the entity. Here is an example of what such a signed Assertion looks like for this issuer:

```
{
    "@context": "https://w3id.org/credentials/v1",
    "id": "urn:uuid:01f0bb90-86ee-4469-9655-7ca6f4d591ae",
    "type": ["Credential", "OpenBadgeCredential"],
    "issuer": "did:example:issuer_did",
    "issued": "2018-02-28T14:58:57.461422+00:00",
    "claim": [{
        "@context": "https://w3id.org/openbadges/v2",
        "id": "urn:uuid:437fc6ff-bb3c-4987-a4b7-be8661ff6f21",
        "type": "Assertion",
        "issuedOn": "2018-02-25T00:00:00+00:00",
        "recipient": {
            "type": "id",
            "identity": "did:example:recipient_did",
            "hashed": false
        },
        "badge": {
            "type": "BadgeClass",
            "id": "urn:uuid:7aad3c57-3bfb-45ea-ae79-5a6023cc62e4",
            "name": "Certificate of Accomplishment",
            "image": "data:image/png;base64,...",
            "description": "Lorem ipsum dolor sit amet, mei docendi concludaturque
ad, cu nec partem graece. Est aperiam consetetur cu, expetenda moderatius
neglegentur ei nam, suas dolor laudem eam an.",
            "criteria": {
                "narrative": "Nibh iriure ei nam, modo ridens neglegentur mel eu.
At his cibo mucius."
            },
            "issuer": {
                "type": "Profile",
                "id": "did:example:issuer_did",
                "name": "Example Issuer",
                "url": "http://example.com",
                "email": "test@example.com"
            }
        },
        "verification": {
            "type": "VerifiableClaim2018"
        }
    }],
    "sec:proof": {
        "@graph": {
            "type": "sec:RsaSignature2018",
            "created": "2018-03-07T22:26:57Z",
            "creator": "did:example:issuer_did#keys-1",
            "sec:jws":
"eyJhbGciOiJQUzI1NiIsImI2NCI6ZmFsc2UsImNyaXQiOlsiYjY0Il19..GMPdCXn-qQcZHPoO6qHn6hBi
ysMNaZ7nSBx_e27LuDxJvRsCLbR1n7LGG7i8NVW1SVMwRjs8aJ3H2XXFphCZF_dGaueTsaehTzLQgh9n5im
PgrQFsAKsRAKTJ_zpVL8JpsbPcrXbb-fkAcD52oDuYJg1uVr3MOhe4BzibDUKaFg5-cXZ-Gs8KcXrh_Ddqt
d8CWw0zS3fRvI3SKbO6op6hNB1Jha4mfAn49Q0BRiSuCxbyPNy5MtX7FGoimvLhsluM7UAtPWHBi6iW8Nh5
7fk4uS5ZywHJSYS9-HPcvbDUGPHPHOnwq4qq7xc47yXveMmyo2VX4YSYe3LM-_9w1TnGg"
        }
    }
}
```

The above example shows a signed message where the claim is the Assertion (Option 2), but the same type of signature could be applied to Option 1 and any other Verifiable Credential.

The issuer `id` declared in the `claim.badge.issuer` is the DID that we expect the verifier will resolve to the `did:example:issuer_did` DID Document shown earlier in this section. When the verifier encounters this "did:example" IRI scheme, it will 1) resolve this DID to its corresponding DID Document, 2) verify the signature, and 3) verify that an authorized key created the signature.

There is some repetition of data in the example above. The issuer id is included at both `issuer` and `claim.badge.issuer`. While this data can be normalized to only appear at `issuer`, leaving it in supports consistency within the Open Badges Specification — there are no breaking changes needed to add this new verification option. In addition, there are attributes of the Issuer Profile that are not assumed to be verifiable by DID resolver methods, such as the name of the issuer, which is used by the Open Badges ecosystem.

A version of the claim using Option 1 looks very similar to non-DID-approach for the issuer, but the validator would be expected to resolve the issuer DID to the DID Document and to use that document to get the applicable signing keys for the issuer and confirm the Verifiable Credential was signed with one of those keys.

```
{
    "@context": ["https://w3id.org/credentials/v1",
"https://w3id.org/openbadges/v2"],
    "id": "https://example.com/assertions/1001",
    "type": ["Credential", "Assertion"],
    "issuer": "did:example:issuer_did",
    "issued": "2018-02-28T14:58:57.461422+00:00",
    "claim": {
        "id": "did:example:recipient_did",
        "obi:holds": {
            "id": "urn:uuid:7aad3c57-3bfb-45ea-ae79-5a6023cc62e4",
            "type": "BadgeClass",
            "name": "Certificate of Accomplishment",
            "image": "data:image/png;base64,...",
            "description": "A badge describing great accomplishments",
            "criteria": {
                "narrative": "Perform tasks of valor and wit."
            },
            "issuer": {
                "type": "Profile",
                "id": "did:example:issuer_did",
                "name": "Example Issuer",
                "url": "http://example.com",
                "email": "test@example.com"
            }
        }
    },
    "obi:evidence": {
        "id": "https://example.org/portfolios/25",
        "name": "Bob's Portfolio",
        "narrative": "Bob worked hard to develop a good portfolio",
        "genre": "ePortfolio"
    },
    "sec:proof": {
        "@graph": {
            "type": "sec:RsaSignature2018",
            "created": "2018-03-07T19:22:15Z",
            "creator": "did:example:issuer_did#keys-1",
            "sec:jws":
"eyJhbGciOiJQUzI1NiIsImI2NCI6ZmFsc2UsImNyaXQiOlsiYjY0Il19..p9RDJqwzXyxfC69dMEz503_Z
Z_af1e_hV931dPlIdrofC6p2y_dcjjqIDysReJy6W_fnN_dZGVoXqFAg2OD_SmbDi5dNMOZILot-zJdDJCx
XWuwZtiCFlt29KfLmJs6me0bD5pU4RbknXDoyBhA8muMby8j1fUeBDo3Ienmzv5UlB3v0f0-w5l6-z_cswH
B_UXIlWw4EzcsmLvHzjB7TI76QLwq3KeVPSB3U9aM3o2Ejkq6Ygh5XxUGkXiZUQ5ungQ9Psy_VicjZyOc19
LoBPoiPxDHQodTrqCFNH2qCNhDc4lg2zE8S9KNlQhUUFatzkTN70s23fhWBMKz2a5DWgQ"
        }
    }
}
```

An important capability enabled by using a DID instead of an HTTP URI as an Issuer `id` is that the issuer entity can authenticate into multiple applications that will then have equal ability to act on behalf of the issuer entity. There will be no implied primary/secondary distinction between these applications, because none of them will "own" the identifier the way DNS allows only one entity to authoritatively own an HTTP identifier.

The DID Document doesn't have any ability to describe in a verifiable sense the issuer's name, URL or email, but these properties are desirable to display within badging systems as part of the Open Badges Profile class, so they are embedded in the claim in these examples. Open Badges ecosystem tools need to determine for themselves when and why they trust these values to be correctly associated with an issuer ID. We can authenticate the DID, but not these properties directly. However, they may be the claims of other Verifiable Credentials where the Issuer is the subject. If a consumer trusts one or more of these claims, they could trust this data wherever it is presented associated with the issuer. This is likely a case where an issuer would want to reference Endorsements (the Open Badges Vocabulary term for "plain" Verifiable Credentials) they have received within the Issuer Profile that gets embedded in claims like the above example.

BLOCKCHAIN PROOF OF EXISTENCE WITH BLOCKCERTS

Blockchain-tethered issuance of Open Badges/Verifiable Credentials is enabled by the same JSON-LD signature and verification framework described here. The MerkleProof2017 LD signature suite allows verification of data anchored to a blockchain. This is the same signature suite used by the current version of Blockcerts, The above Option 1 example modified to use this MerkleProof2017 signature suite looks the following:

```
{
    "@context": ["https://w3id.org/credentials/v1",
                 "https://w3id.org/openbadges/v2",
                 "https://w3id.org/blockcerts/v2"],
    "id": "https://example.com/assertions/1001",
    "type": ["Credential", "Assertion"],
    "issuer": "did:example:issuer_did",
    "issued": "2018-02-28T14:58:57.461422+00:00",
    "claim": {
        "id": "did:example:recipient_did",
        "obi:holds": {
            "id": "urn:uuid:7aad3c57-3bfb-45ea-ae79-5a6023cc62e4",
            "type": "BadgeClass",
            "name": "Certificate of Accomplishment",
            "image": "data:image/png;base64,...",
            "description": "A badge describing great accomplishments",
            "criteria": {
                "narrative": "Perform tasks of valor and wit."
            },
            "issuer": {
                "type": "Profile",
                "id": "did:example:issuer_did",
                "name": "Example Issuer",
                "url": "http://example.com",
                "email": "test@example.com"
            }
        }
    },
    "obi:evidence": {
        "id": "https://example.org/portfolios/25",
        "name": "Bob's Portfolio",
        "narrative": "Bob worked hard to develop a good portfolio",
        "genre": "ePortfolio"
    },
    "sec:proof": {
            "type": "MerkleProof2017",
            "targetHash":
                "637ec732fa4b7b56f4c15a6a12680519a17a9e9eade09f5b424a48eb0e6f5ad0",
            "merkleRoot":
                "f029b45bb1a7b1f0b970f6de35344b73cccd16177b4c037acbc2541c7fc27078",
            "anchors": [ {
                "sourceId":

                "d75b7a5bdb3d5244b753e6b84e987267cfa4ffa7a532a2ed49ad3848be1d82f8",
                "type": "BTCOpReturn"
            }],
            "proof": [{
                "right":
                "11174e220fe74de907d1107e2a357e41434123f2948fc6b946fbfd7e3e3eecd1"
             }]
        }
    }
}
```

This demonstrates a special form of proof, applying to the transaction containing the data. This proof technique is known as a "Merkle proof" — a cryptographic data structure used to show that the present credential belongs to a structure that is anchored to a blockchain transaction. This provides proof of existence through confidence in the timestamp of the Assertion, via the timestamp (if supported by the blockchain) associated with the block's addition to the ledger (according to the blockchain's consensus protocol and confirmation threshold). This also allows more than one credential to be part of a single blockchain transaction; batching of credentials is commonly performed for cost savings alongside natural logical groupings of credentials (e.g. "Graduates of 2018").

Informally, the Merkle proof, along with JSON-LD normalization, allows you to confirm that the local data matches an expected hash, and that hash combined with redacted data (and functions of redacted data), combine to a value that matches the value on the blockchain.

More precisely, during verification, the hash of the present credential is calculated to ensure it matches `targetHash`. The `proof` section is how to merge this local credential with one-way hashes of other credentials (to avoid revealing the contents) into a tree structure, the base of which should equal `merkleRoot`. Lastly, the `merkleRoot` value is the expected value on the blockchain, which we can independently confirm. In the `anchors` part of the proof, the `type` and `sourceId` incidate which blockchain, transaction id, and transaction field to fetch; this value should equal the `merkleRoot` value.

This example shows an issuer identified with a DID (`did:example:issuer_did`). This enables an improved authenticity check during the Blockcerts verification process, which will ensure the public key that signed the issuing transaction is authorized by the DID document associated with the issuer's DID.

This format is proposed for Blockcerts v3, and has the advantage of unifying Blockcerts more tightly into Verifiable Credentials / Open Badges, as opposed to a separate Open Badge extension.

IMPLEMENTING USER STORIES FOR OPEN BADGES TOOLS

The two different options outlined in this paper pose different distributions of cost and complexity on Open Badges Applications in the core roles of Validator, Issuer, Host, and Displayer. When choosing which option to pursue to implement the capabilities described above, understanding the implementation path for each option enables cost/benefit analysis.

Validator

The Open Badges Validator is the main application that needs to be updated to support these features. The validator will need to be modified to detect that an input string is valid JSON-LD and of type `Credential`. The validator then queues a relevant procedure to intake the document that will be an alternative path to the existing procedures for intake based on a hosted URL input, image input, JWS string input, and existing JSON strings of Assertion, BadgeClass or Profile.

Validating Option 2

Because internally, Option 2 looks entirely like an existing Assertion, the validator will not need to make many modifications to its internal expectations about how Open Badges objects are published.

Resolving DIDs in this example occurs in neat parallel to resolving HTTP(s) identifiers into usable documents, but instead of the HTTP-based integrity verification tasks, DID-specific tasks will be queued, including:

- Resolve document from the DID.
- Verify DID Document signature (if signed) and requirement for it to be signed for particular contexts.
- Verify proofs for a few relevant Linked Data Signature Suites.
- Confirm badge object signing key is authorized to sign on behalf of DID-identified entity.

Validating Option 1

Option 1 requires some changes to the Validator in addition to the DID resolution and signature verification changes.

- Update expectation for client that the Verifiable Credential style class can appear in the resultant data graph in place of the "bare" Assertion class. This structure is fundamentally the same as that already appears in the graph for Open Badges Endorsements, which are referenced from within the objects discovered while processing the verification input.
- Add procedure to queue up tests for the `obi:holds` claim in an VC that is found as part of processing.
- For convenience, the validator should be able to deliver a version of the VC translated into the 2.0 Assertion style to serve Backpacks and Displayers that cannot yet adapt to the Option 2 VC as Assertion model. This legacy version should be set as the "validationSubject" in the validation report.

Issuer

The `OpenBadgeCredential` wrapper is an optional verification type for Open Badges. That means issuers don't need to implement it in order to continue to issue valid Open Badges. Once the validator is updated, issuers can publish badges using these options, with very little distinction between them.

To use a DID as a recipient identifier, badges can be published with any method supported by the validator that uses the DID for recipient identification. It may be the subject identifier of a claim (as in Option 1) or it may be hashed in an IdentityObject (Option 1 or 2), for instance.

In order to use a DID as an issuer identifier, the Issuer Service will need to create a new DID or gain access to an authorized signing key created by the DID Owner user (e.g. "Alice"), for which the validator will be able to confirm authorization. For example, Alice could use the features of her DID Resolver Method to add the Issuer Service's signing key to her DID Document. Some implementing Issuer services may register the DID themselves and act as its owner, instead of authenticating and gaining access to a DID created by a user. This would likely not allow the user of the Issuer System who created the Issuer Profile to independently act as the DID owner themselves outside of the system, but it is a potentially valid path to implementation.

Extending the possibilities explored by example above, one emerging complementary technology is Linked Data Object Capabilities. Under a system that uses this, Alice will act as her own DID Owner and will grant the Issuer Service the capabilities to perform several relevant Open Badges actions on her behalf, such as Issuing. When the issuing system invokes these capabilities, it will reference the capability chain in its proof.

Capabilities are safer than adding an external service's key to your DID Document as if it were your own, because Capabilities may be attenuated to only grant specific actions to the capability holder. To the validator, if the Validator supported Linked Data Capabilities, it would appear as if Alice issued a signed Verifiable Credential Open Badge herself, though the audit trail would be available to determine that a key held by the Issuer Service invoked an `Issue` capability in order to sign the valid proof. This topic is explored in more detail in a complementary paper about Open Badges Peer Claims.

Another alternative would be for Alice to sign new Assertions interactively, using a private key held only on a trusted device. The issuer service could prepare the Assertions for signature and then obtain the signature from Alice directly.

Host/Backpack

Backpacks mostly rely on the verification information returned from a validator, so if the validator can be updated to present the Open Badges data in a predictable supported format, hosts don't need to specially support the Verifiable Credentials wrapper.

Verifying that badges belong to users of the Host service is one of their most important capabilities, however, and the introduction of DIDs as recipient identifiers introduces significant new requirements around that, similar in scope to issuer support.

There are some emerging capabilities being described for the Authentication of users identified by a DID who are interacting with a system via browser. These are in very early draft stages, so backpacks that wish to resolve and authenticate users as owners of DIDs will likely need to implement an interactive process. These features could be implemented in Host services or via external identity providers.

Collaboration with implementers of DID-resolver and authentication services seems like a fruitful direction in order to achieve a good pace of adoption and even implementation support across an ecosystem for the various resolver methods and authentication methods that will begin to proliferate. If an external identity provider is trusted by an Open Badges Host, a connection could be made via OAuth2 or similar protocol that may already be built into a backpack product.

DID owners may provide one or more methods of authentication in their DID document, and relying services may select an authentication method from those available. They would establish a communication channel with a user attempting to log in to send an authentication challenge to the user's selected authentication device.

For example, if the user keeps their signing key(s) on a mobile phone, they could be directed to scan a QR Code to obtain authentication challenge data, which they would sign and submit, completing an authentication process on the device where they started the request, upon which, they could be sent back to the Host, authenticated as the owner of the DID.

Once a Host has authenticated that a particular local user controls a specific DID, it may accept badges into that user's account that identify their recipient using that DID.

Displayer

Displayers, like Hosts, rely on the return value of the validator to display trustworthy data, so if the validator returns essentially the same data structure, very few updates will be needed to support awards in VC format. Prototypes should negotiate a balance between adapting Displayers to handle the VC that makes a `obi:holds` claim and the validator returning a 2.0 Assertion style object in the graph that it synthesizes.

NEXT STEPS

An implementation of these capabilities will be best informed by prototypes of experimental branches of open source Open Badges software that fills the different ecosystem roles.

Our primary goal with this proposal is to gain feedback from the Open Badges and Verifiable Credentials communities. We have planned implementations to prove out these concepts, including: - Open Badges Validator support for JSON-LD signature verification using Option 1 with Option 2 as a fallback. - Issuing Blockerts credentials that can be validated in the format described here. - Adding support for Blockcerts blockchain pointer in validator. - Collaborating with those in the Rebooting Web of Trust Community to integrate at least one method of DID Authentication into an Open Badges Host to connect reliably.

The lessons learned from these prototypes will feed use cases to submit in the next stages of standardization of

Open Badges and related technologies.

ADDITIONAL CREDITS

Lead Authors: Nate Otto (Concentric Sky) & Kim Hamilton Duffy (Learning Machine)

Additional Contributors: Kulpreet Singh (Chlu), Luiz Gustavo Ferraz Aoqui (IBM)

About Rebooting the Web of Trust

This paper was produced as part of the Rebooting the Web of Trust VI design workshop. On March 6th to 8th, 2018, over 40 tech visionaries came together in Santa Barbara, California to talk about the future of decentralized trust on the internet with the goal of writing 3-5 white papers and specs. This is one of them.

Named Sponsors List: Sovrin Foundation, PTB Holdings

Workshop Credits: Christopher Allen (Founder), Joe Andrieu, PMP (Producer and Facilitator), Shannon Appelcline (Editor-in-chief), Erica Connell (Event Coordinator), Claire Rumore (Graphical Recorder), and The Narrative Loft (Venue)

RWOT Leadership Team: Christopher Allen, Joe Andrieu, Kim Hamilton Duffy, Manu Sporny, and Heather Vescent

Thanks to our other contributors and sponsors!

What's Next?

The design workshop and this paper are just starting points for Rebooting the Web of Trust. If you have any comments, thoughts, or expansions on this paper, please post them to our GitHub issues page:

https://github.com/WebOfTrustInfo/rebooting-the-web-of-trust-spring2018/issues

The next Rebooting the Web of Trust design workshop is scheduled for for Fall 2018. If you'd like to be involved or would like to help sponsor these events, email:

rwot-leadership@googlegroups.com
