Identity Hub Attestation Flows and Components

A White Paper Draft from Rebooting the Web of Trust VI

by Daniel Buchner (Daniel.Buchner@microsoft.com),
Cherie Duncan (Cherie.Duncan@dominode.com),
John Toohey – john.toohey@dominode.com,
Ron Kreutzer (ron@pillarproject.io), and
Stephen Curran (swcurran@cloudcompass.ca)

This paper is a draft, expected to undergo changes through the Rebooting the Web of Trust VII design workshop.

ABSTRACT

In this document, we define a set of user flows and describe the associated Action Objects that support a Hub-centric approach to the request, issuance, presentation, verification, and revocation of interoperable attestations. This document extends the Identity Hub Explainer.

1 INTRODUCTION

In the digital identity space, Hubs let you securely store and share data. A Hub is a datastore containing semantic data objects at well-known locations. An identity needs to be able to prove that some data is true to another entity that requests it. These attestations are that method of proof. In the digital world, the requester may be software, and the response may or may not require involvement of the individual/identity who the proof is being made against. These examples and flows depict how attestations are requested and resolved.

2 EXAMPLE USE CASES

We use examples here to give guidance/suggestions for how attestations can be used with real-world examples. The overall use case is a person, Alice, who registers for College using a process that includes using an attestation she possesses to prove she has received some required immunizations. After graduation, Alice requests an attestation from the College that she has graduated, and presents that attestation to her professional profile on a professional network.

Agents

We use the term "User Agent" (UA) to refer to an app on a smartphone or other device that has access to DID-linked keys and the power to do things on behalf of a DID owner (Alice). This could also be referred to as a digital wallet. Similarly, we use the term "Enterprise Agent" (EA) to refer to the comparable component representing an Organization – e.g. a College or professional network. A UA and EA are conceptually the same, but while the UA is likely a personal device (laptop, tablet, phone), an EA is likely a service that processes requests based on business rules and data held in back-end systems. Note that an EA might need input from a specific member of the organization to complete the processing of a request. In that case, the EA might contact that user through that person's User Agent (although there are many other possibilities).

Sites

In the examples below, "Sites" are assumed to be Web or Mobile Site – user interfaces that allow a user (in our case, Alice) to trigger the start of a process. There are many other ways to trigger the start of such a process.

Decentralized IDs (DIDs), Documents and Attestations

Each of Alice's Decentralized Identifiers (DIDs) referenced in the scenarios is generated and held by her user agent (UA) and used for a specific purpose - for example her relationship with the College. Her DIDs are not necessarily correlated to any other identifiers that make up her identity. Per the W3C DID Specification, a DID Document is associated with a DID that contains information about the public keys and service endpoints for that DID. Thus, given a DID and DID Document for another Identity, an entity has a mechanism to resolve and communicate with the Identity Owner of the DID. DIDs may be public and stored on a publicly available Distributed Ledger, with their associated DID Document found via the DIF Universal Resolver, or may be pairwise private DIDs, where two Identities directly exchange DIDs/DID Documents.

An attestation is something (such as a Verifiable Credential) issued by an entity to a holder (often the subject of the attestation) so that the holder can prove to others that they hold the attestation. In one of the examples below, for instance, Alice wants to receive a graduation attestation from the College so that she can present (prove) that attestation to a professional network.

Interface Guidelines: Hubs, Agents and Identity Owners

Some basic guidelines are defined about Hubs, Agents, and their Identity Owners:

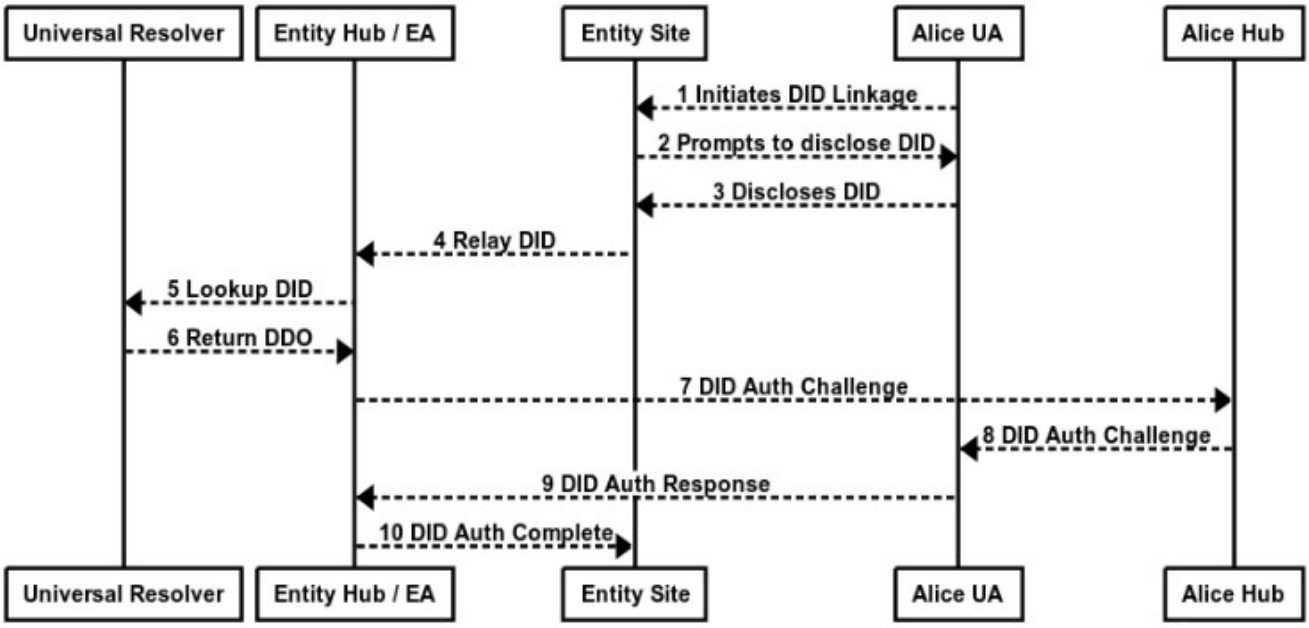
- Private keys are accessible only to Agents (User and Enterprise), thus any encrypting/signing of information must be done by an Agent.
- In general, Hubs are addressable using the service pointers located in a DID Document, and Agents are addressed via a user's Hub. The only exception is invocation of a User Agent through direct mechanisms, like a deep link on a mobile site, a QR code on a Web site scanned by a User Agent, or a Bluetooth/NFC data exchange.
- Hubs generally have limited, generic functionality, and any decision making must be made at the Agent level via a user app/device (User Agents) or more automated business services (Enterprise Agents).

For simplicity, we show the Hub and Enterprise Agent as a single entity in the following scenarios. In typical implementations, they will be separate entities that communicate to accomplish their respective activities.

2.1 Alice Links to an Entity

In order to communicate a request for attestation to an entity (in our examples, Alice), a user will first need to establish a connection between her user agent and the entity she will interact with. This is necessary for all follow-on scenarios.

Alice wants to transact with the entities described in the scenarios with the intent to receive or exchange attestations. First and foremost, the entity must verify that Alice is the owner of the decentralized identifier she claims. In order to find Alice's user agent, we leverage the Universal Resolver (UR) to lookup Alice's Decentralized Identifier (DID) to find her DID Document (DDO). The keys located in Alice's DDO are used to authenticate Alice's ownership of the DID and to determine access to Alice's hub and user agent.



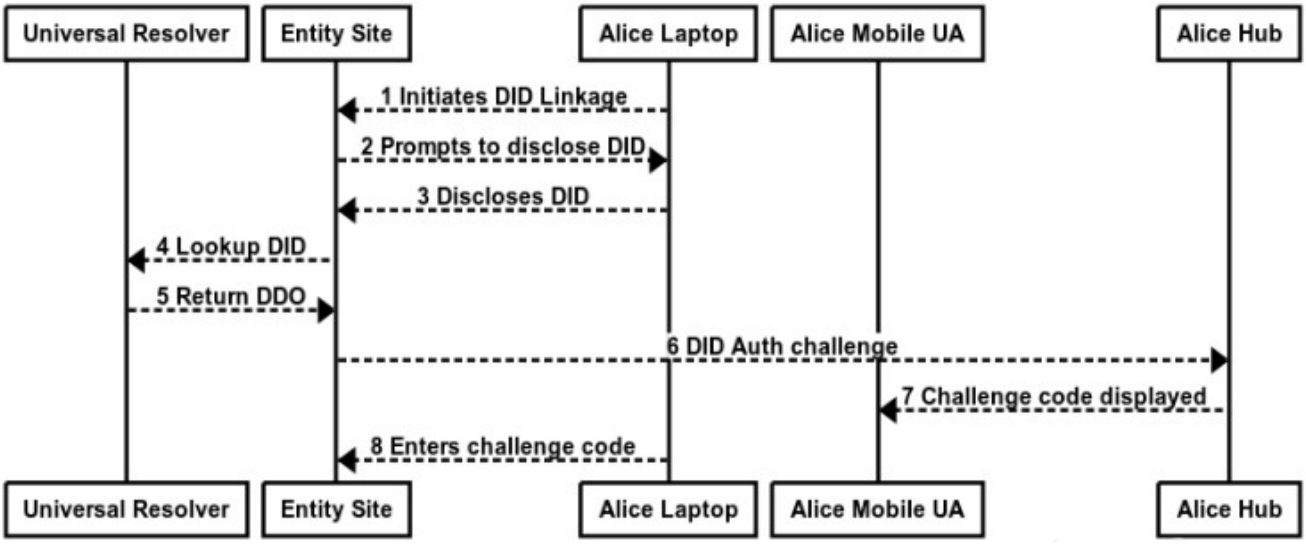
1. Alice navigates to an entity's website and clicks a link to initiate a DID linkage with the entity. The content received from clicking the link includes DID information about the Enitity that Alice should use for the relationship.
    1. Alice may have to use the Universal Resolver to access the DID Document associated with the DID.
2. The entity prompts for Alice to disclose a DID that represents her digital identity.
    1. If the website was accessed via a laptop/desktop, the website typically displays a QR Code, and Alice uses her mobile wallet app to scan the QR. If the website was accessed via her mobile device, a protocol handler raises Alice's AU app.
3. Alice selects an existing DID or creates a new DID for this relationship and sends the DID to the Entity Site.
4. The Entity Site passes the DID to the Entity's Enterprise Agent to initiate the DID Auth response.
5. The EA uses the Universal Resolver (UR) to request retrieval of the DID Document that matches the

provided DID.

6. The DID Document is returned to the EA.
7. The EA initiates the DID Auth process by issuing a challenge to Alice's Hub.
8. Alice's Hub passes the DID Auth challenge to Alice's User Agent for signing.
9. Alice's User Agent proves her identity with a signed response to the auth challenge.
10. The Entity Hub confirms the response and notifies the Entity Site with a successful login.

2.1.1 DIF Identity Hub 2FA

A second identity linking scenario to consider is when Alice is registering with the site using a device that is not a UA, yet she still wants to use her UA to establish the connection. In this case, Alice discloses a DID connected to her UA to the site, the site contacts the UA and the mobile device containing the UA displays a code for Alice to use. Alice enters the code into a form on the site, proving that she controls the DID.



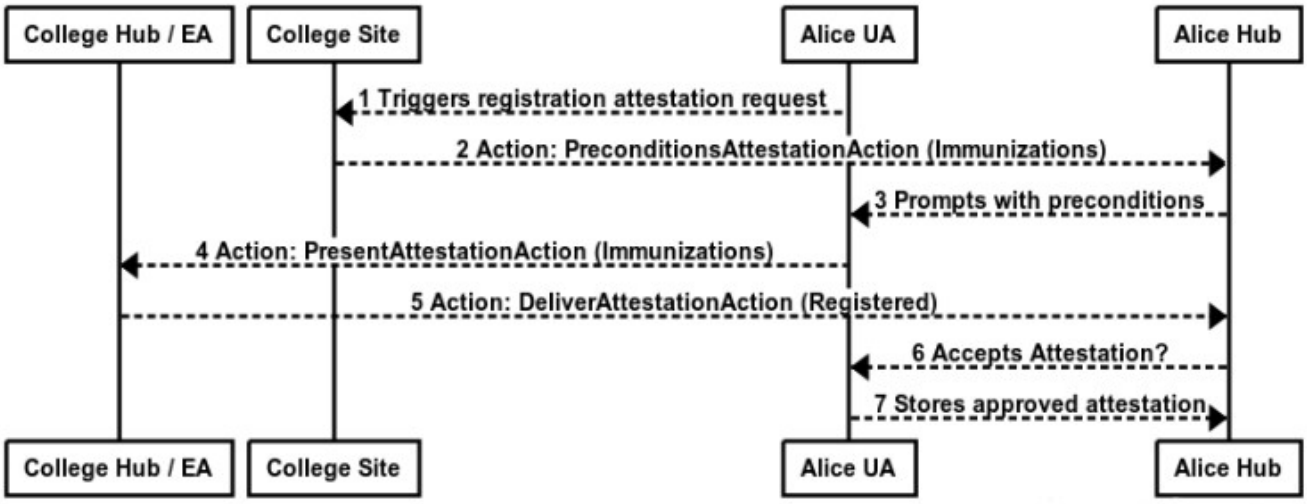
1. Alice navigates to an entity's website and clicks a link to initiate a DID linkage with the entity.
2. The entity prompts for Alice to disclose a DID that represents her digital identity.
3. Alice selects an existing DID and sends the DID to the Entity Site.
4. The Entity/EA uses the Universal Resolver (UR) to request retrieval of the DID Document that matches the provided DID.
5. The DID Document is returned to the Entity/EA.
6. The EA initiates the DID Auth process by issuing a challenge to Alice's Hub.
7. Alice's Hub passes the DID Auth challenge to Alice's User Agent for signing.
8. Alice's User Agent processes the challenge and displays a code expected by the Entity Site on the mobile device.
9. Alice enters the code on her laptop and the Entity Site confirms the response, resulting in a successful login.

2.2 Alice Must Provide Preconditional Proof

Alice is attempting to register for college and her DID is already linked to the College. In this example, for Alice to get admitted to the College, she must prove that she previously has received appropriate immunizations.

Assumptions

- Alice is linked to the College via her DID.
- Alice has an Identity Hub accessed via an application on her mobile device.
- Alice has a verified digital attestation for her previous immunizations.



1. Alice initiates a Registration request on the College Site.
2. The College EA determines there are preconditions for Registration: she must prove she has the required immunizations. The College EA initiates a request for presentation of the preconditions.
3. Alice is prompted by her UA to provide the preconditions.
4. Alice selects the correct attestation to use and her UA sends them back to the College Hub.
5. The College EA processes the preconditions and sends a Registered Student attestation to Alice's Hub.
6. Alice accepts the request to accept/store the college registration attestation.
7. Alice's Hub stores the Registered Student attestation and broadcasts it to her connected devices.

Referenced Action Objects

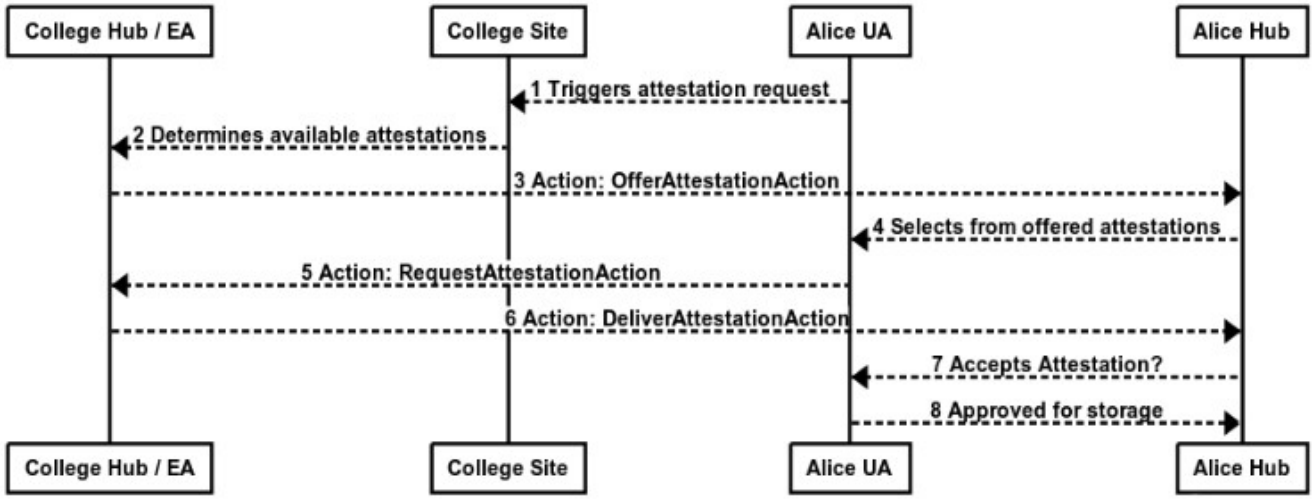
- PreconditionsAttestationAction
- PresentAttestationAction
- DeliverAttestationAction

2.3 Alice Obtains a Diploma Attestation

In this example, Alice has graduated from college and wants to acquire a digital diploma attestation.

Assumptions

- Alice is linked to the College via her DID.
- Alice has an Identity Hub accessed via an application on her mobile device.
- Alice has graduated from College.



1. Alice initiates a request through the College website to obtain an attestation regarding her graduation.
2. College website reaches out to its Enterprise Agent service to determine what attestations are available for Alice.
3. The College's EA sends an attestation offer to Alice's Hub.
4. Alice's UA receives a Action from the College EA that contains the attestations it can provide. Alice selects the attestations she wants.
5. Alice's UA sends an attestation request for her selected attestations to the College's Hub.
6. The EA delivers the attestations to Alice's Hub.
7. Alice is prompted to accept or deny the attestation.
8. Alice accepts the attestation and stores it across her Hubs and devices.

Referenced Action Objects

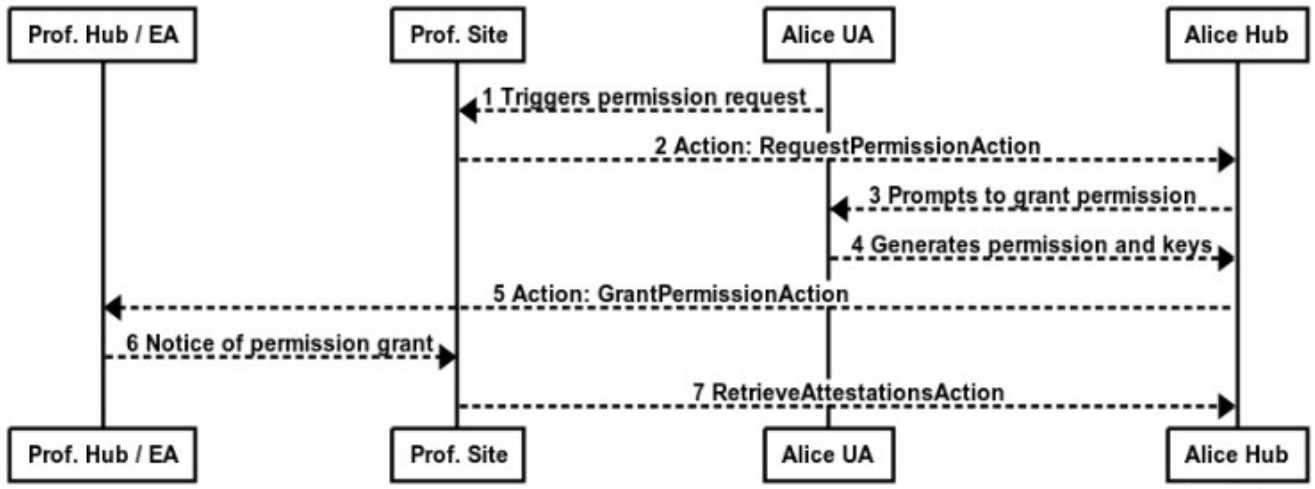
- OfferAttestationAction
- RequestAttestationAction
- DeliverAttestationAction

2.4 Alice Shares Her Education Verification, and Future Updates, with a Professional Networking Site

Alice has graduated from college, possesses an attestation from the College, and wants to share her existing and future education attestations with a professional networking site.

Assumptions

- The site has linked Alice to her DID via DID Auth.
- Alice has an Identity Hub, accessible via an app on her mobile device.
- Alice possesses an attestation for her college diploma.



1. Alice navigates to the professional network site and initiates the flow to grant access to her educational attestations.
2. The website sends a `RequestPermissionAction` to Alice's Hub.
3. Alice's Hub relays the request to Alice's UA, which prompts her to grant/deny permission.
4. Alice grants permission to access her current and future educational attestations by pushing a signed permission object and DID-specific keys to her Hub.
5. Alice's Hub stores the keys she generated for the professional networking site and relays an Action to the professional network's Hub to provide notice that their permission request has been granted.
6. The professional networking site is notified that the permission has been granted.
7. At any time in the future, the professional networking site can retrieve Alice's education credentials from Alice's Hub, based on the permissions she provided and using the private key held by the professional networking site.
    - Should the permission later be removed, the Prof Site's ability to retrieve updated credentials will be removed.

Referenced Action Objects

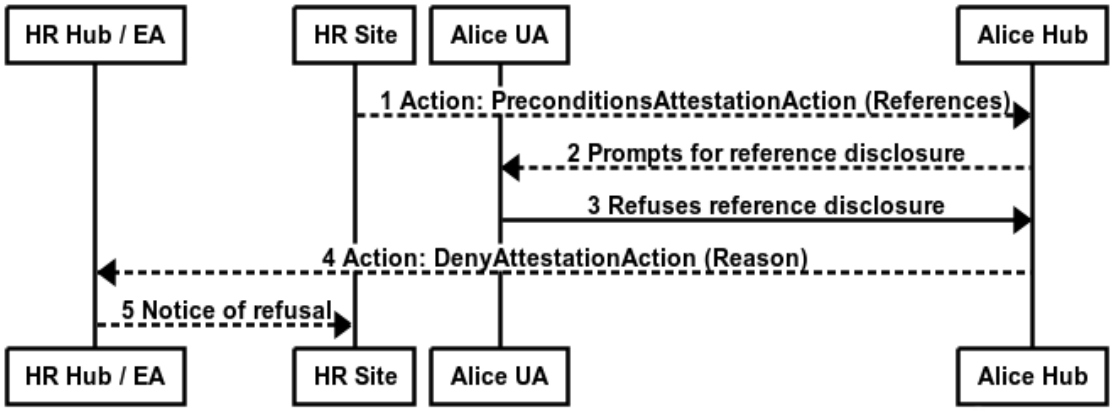
- RequestPermissionAction
- GrantPermissionAction
- RetrieveAttestationsAction

2.5 Alice Applies for a Job and Refuses to Provide References

Alice is applying for a job and has connected with the HR department via her DID. Alice has already provided some basic attestations about her right to work, name, address, etc. But when she receives a request for her references Alice refuses/denies the request as by the time this request comes in Alice has already accepted a position somewhere else (for example).

Assumptions

- Alice is linked to Company's HR via her DID.
- Alice has an Identity Hub accessed via an application on her mobile device.
- Alice has a verified digital attestation for her references but does not wish to share them at this time, or Alice does not have references in her digital wallet yet.

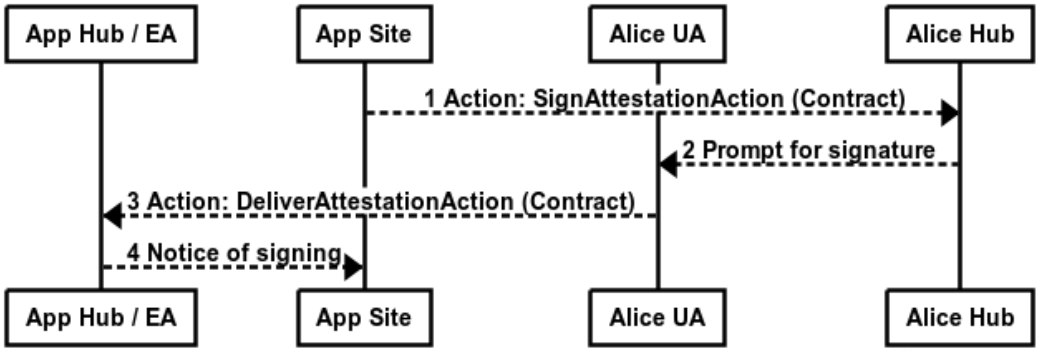


1. HR initiates a request for references via Alice's Hub.
2. Alice's Hub finds appropriate Attestations and provides them to Alice's User Agent.
3. For whatever reason, Alice refuses (via her agent) to provide references at this time.
4. Alice's Hub notifies HR Hub of the refusal, with optional reason for refusal.
5. A notification is sent to HR with the refusal details (generic or specific to the scenario).

Referenced Action Objects

- PreconditionAttestationAction
- DenyAttestionAction

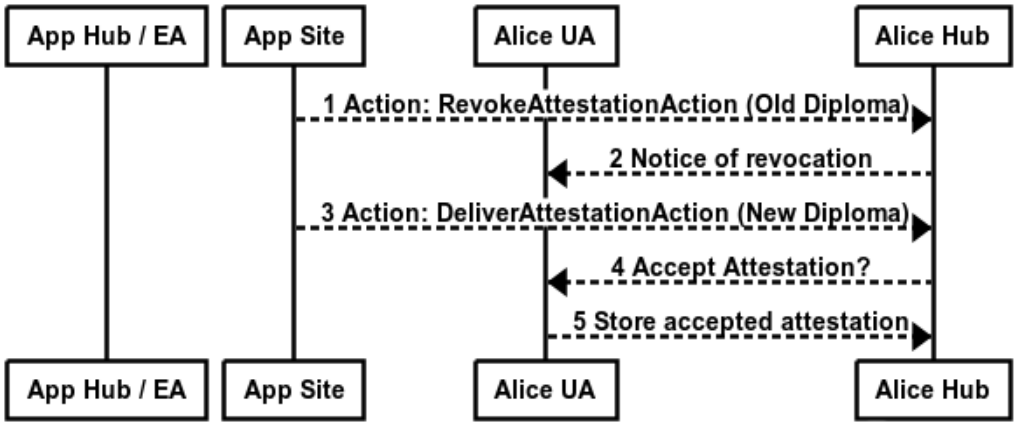
2.6 A Bank Sends Alice a Contract that Requires her DID signature, which She Signs and Delivers Back to the Bank.



Referenced Action Objects

- SignAttestionAction
- DeliverAttestationAction

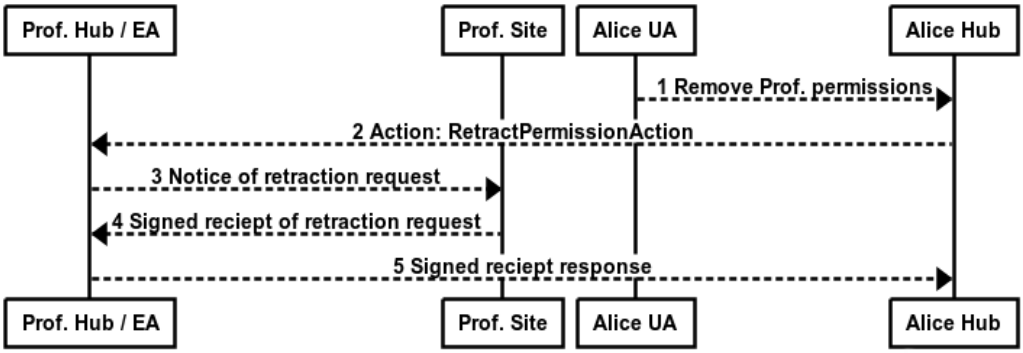
2.7 The College Determines Alice Was Issued a Nursing Certificate Instead of Her CS diploma, so They Revoke the Attestation and Issue the Correct One.



Referenced Action Objects

- RevokeAttestationAction
- DeliverAttestationAction

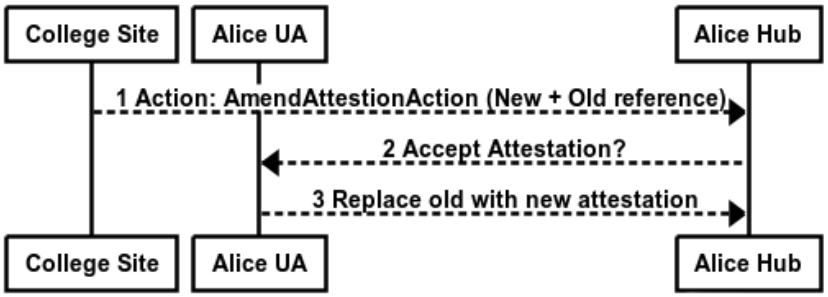
2.8 Alice Retracts Data Access Permission from a Professional Networking Site.



Referenced Action Objects

- RetractPermissionAction

2.9 Alice's College Discovers they Made a Mistake on her Diploma Attestation, and Sends her an Amended Attestation with the Correct Info.



Referenced Action Objects

- AmendAttestionAction

3 ACTION OBJECTS

Identity Hub attestation handling relies on the passage and recognition of common Action types that Hubs, User Agents, and consuming apps/services understand. In order to ensure that the flows related to attestations are precise and maximally descriptive of their intent, the Identity Hub spec will define its own Action objects for each of the relevant attestation actions. These objects are extensions of the Schema.org Action object, the schema origin of which shall be schema.identity.foundation. These objects are strictly a shared means of communicating and facilitating the various activities related to attestations; they do not infer or require a specific type of proof format or material be used within them.

Note that each Action returns only a status of whether the Action was successfully (or not) transmitted. The result of processing the request is conveyed to the caller via a subsequent Action.

The following is a description of the objects and examples that encompass their structure and properties:

3.1 RequestAttestationAction

The Holder requests an attestation from an Issuer.

- Type of attestation wanted
- List of tag strings to describe the attestation
- Detailed, human-readable description of the attestation being requested (mostly for UAs to display to users)
- Who is the attestation for?
- What format do you need it in?
- Enable passing of preconditions
- Option to set a deadline for issuance/fulfillment

```
{
    "@context": "http://schema.identity.foundation/",
    "@type": "RequestAttestationAction",
    "identifier": UNIQUE_ID,
    "for": ["did:foo:123-456"],
    "format": CLAIM_FORMAT,
    "expiration": EPOCH_TIME,
    "description": "Province of British Columbia Driver's License",
    "tags": ["license", "driving", "permit", "DL", "driver's license"],
    "preconditions": ARRAY_OF_PRECONDITION_PROOFS (optional)
}
```

3.2 DenyAttestationAction

In response to a request for an Attestation, a Verifier/Issuer informs a Holder that the attestation cannot be provided. This **`Action`** inherits from schema.org's **`RejectAction`**.

- Linked attestation action ID
- Reason for refusing the Request Attestation Action.

```
{
    "@context": "http://schema.identity.foundation/",
    "@type": "DenyAttestationAction",
    "identifier": UNIQUE_ID,
    "purpose": "We cannot issue your diploma, you have not graduated."
}
```

3.3 PreconditionsAttestationAction

In response to a request for an Attestation, a Verifier/Issuer informs a Holder a list of Pre-Conditions that must be met before the requested Attestation can be issued.

- Linked attestation action ID
- Specify set of preconditions, each with their own descriptors

```
{
    "@context": "http://schema.identity.foundation/",
    "@type": "PreconditionsAttestationAction",
    "identifier": UNIQUE_ID,
    "preconditions": ARRAY_OF_PRECONDITION_DESCRIPTORS
}
```

3.4 OfferAttestationAction

In response to a request for an Attestation that cannot be issued because that type is not available, provide to the Holder a list of attestations that ARE available.

- For each attestation type available to the requester:
  - ⇨ Type of attestation
  - ⇨ List of tag strings to describe the attestation
  - ⇨ Detailed, human-readable description of the attestation being requested (mostly for UAs to display to users)
  - ⇨ Formats available for the attestation

```
{
    "@context": "http://schema.identity.foundation/",
    "@type": "OfferAttestationAction",
    "identifier": UNIQUE_ID,
    "availableAttestations": ARRAY_OF_ATTESTATION_DESCRIPTORS
}
```

3.5 DeliverAttestationAction

Used by any party that delivers a finalized attestation to a target entity. This **`Action`** inherits from schema.org's **`SendAction`**.

- Linked attestation action ID
- Payload of the proof material
- Format of the proof material
- Time delivered

```
{
    "@context": "http://schema.identity.foundation/",
    "@type": "DeliverAttestationAction",
    "identifier": "UNIQUE_ID",
    "object": ATTESTATION_PAYLOAD,
    "description": "Province of British Columbia Driver's License",
    "tags": ["license", "driving", "permit", "DL", "driver's license"]
}
```

3.6 PresentAttestationAction

This Action is the envelop used to present an attestation to an inspecting party.

- List of tag strings to describe the attestation
- Detailed, human-readable description of the attestation being requested (mostly for UAs and EAs to reason over and use in display)
- Format of the attestation payload
- The attestation payload

```
{
    "@context": "http://schema.identity.foundation/",
    "@type": "PresentAttestationAction",
    "object": ATTESTATION_PAYLOAD,
    "description": "MIT Diploma for B.S. in Computer Science",
    "tags": ["diploma", "degree"]
}
```

3.7 SignAttestationAction

A party sends a Action to a target prompting them to sign the provided attestation payload. This **Action** inherits from schema.org's **EndorseAction**.

- Linked attestation action ID
- Payload of the proof material
- Format of the proof material
- Time delivered

```
{
    "@context": "http://schema.identity.foundation/",
    "@type": "SignAttestationAction",
    "identifier": UNIQUE_ID,
    "object": ATTESTATION_PAYLOAD,
    "description": "Loan for 123 Main Street, Anytown USA"
}
```

3.8 RevokeAttestationAction

The party that previously supplied an attestation sends a notice to the attestation owner/holder that issuing party has revoked the attestation. This **Action** inherits from schema.org's **DeactivateAction**.

- Attestation ID
- Revocation code - array of revocation codes (look for an existing standard)
- Reason for revocation - array of human-readable descriptions of the reason, or URI

```
{
    "@context": "http://schema.identity.foundation/",
    "@type": "RevokeAttestationAction",
    "identifier": UNIQUE_ID,
    "object": ATTESTATION_PAYLOAD,
    "result": REVOCATION_RECORD,
    "purpose": "Your driver's license was revoked."
}
```

3.9 AmendAttestationAction

Used to update an attestation. Requires past ID, optionally including previous attestation. This **Action** inherits from schema.org's **ReplaceAction**.

- Attestation ID
- Change delta of some kind
- Reason for amendment - array of human-readable descriptions of the reason, or URI

```
{
    "@context": "http://schema.identity.foundation/",
    "@type": "AmendAttestationAction",
    "identifier": UNIQUE_ID,
    "object": ATTESTATION_PAYLOAD,
    "purpose": "Your driver's license was amended with your latest picture"
}
```

3.10 RequestPermissionAction

Request permission for access to a DID's Identity Hub data. This **Action** inherits from schema.org's **AuthorizeAction**.

- Permission being requested
- Intended use of data being requested

```
{
    "@context": "http://schema.identity.foundation/",
    "@type": "RequestPermissionAction",
    "object": PERMISSION_OCAP,
    "purpose": "Display and filtering on a professional network",
}
```

3.11 GrantPermissionAction

The party that allows a permission sends a notice to the requesting party to let them know the permission has been granted. This **Action** inherits from schema.org's **AcceptAction**.

- Permission being requested
- Intended use of data being requested

```
{
    "@context": "http://schema.identity.foundation/",
    "@type": "GrantPermissionAction",
    "object": PERMISSION_OCAP
}
```

3.12 DenyPermissionAction

The party evaluating the permission request does not grant the permission and sends the requesting party a notice of the denial. This **Action** inherits from schema.org's **RejectAction**.

There is not currently an example of this action in the scenarios in Section 2 of this document.

- Permission being requested
- Intended use of data being requested

```
{
    "@context": "http://schema.identity.foundation/",
    "@type": "DenyPermissionAction",
    "object": PERMISSION_OCAP,
    "purpose": "I do not want to allow you access at this time",
}
```

3.13 RetractPermissionAction

The party that has previously issued a permission granting access sends a notice to the affected party to let them know the permission has been retracted. This **Action** inherits from schema.org's **DeleteAction**.

- Permission being retracted

```
{
    "@context": "http://schema.identity.foundation/",
    "@type": "RetractPermissionAction",
    "object": PERMISSION_OCAP,
    "purpose": "I no longer want you to have access to my attestations",
}
```

3.14 RetrieveAttestationsAction

Used by any party that has been granted permission access to a set of Attestations via the GrantPermissionAction to retrieve a set of Attestations.

```
{
    "@context": "http://schema.identity.foundation/",
    "@type": "RetrieveAttestationsAction",
    "identifier": "UNIQUE_ID",
    "object": ATTESTATION_PAYLOAD,
    "description": "Province of British Columbia Driver's License",
    "tags": ["license", "driving", "permit", "DL", "driver's license"]
}
```

4 GLOSSARY

- Decentralized Identifier: Decentralized Identifiers (DIDs) are a new type of identifier for verifiable, "self-sovereign" digital identity. DIDs are fully under the control of the DID subject, independent from any centralized registry, identity provider, or certificate authority.
- DID: Decentralized Identifier
- DID Auth: Authentication of an Identity by verifying the Identity's control of its DID
- DID Document: The control document that specifies keys, service endpoints, and other basic details about a DID.

- DDO: Abbreviation for a DID Document
- EA: Enterprise Agent: a HUB-aware service that integrates with an Enterprise's backend systems and representatives to process HUB requests. Conceptually equivalent to a person's UA, but for an organization.
- UA: Abbreviation for User Agent
- Universal Resolver: A mechanism of getting the DID Document associated with a DID across any (supported) DID implementation
- UR: Abbreviation for Universal Resolver
- User Agent: a smartphone-based digital wallet, browser

5 TECHNICAL & SPEC IMPLICATIONS

- For the Hub /permission spec: add optional timeout for permissions

---

ADDITIONAL CREDITS

Lead Author: Daniel Buchner (Daniel.Buchner@microsoft.com),

Additional Authors: Cherie Duncan (Cherie.Duncan@dominode.com), John Toohey (john.toohey@dominode.com), Ron Kreutzer (ron@pillarproject.io), and Stephen Curran (swcurran@cloudcompass.ca)

Sequence Diagrams: Created at WebSequenceDiagrams (https://www.websequencediagrams.com/)

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 September 26th-28th, 2018 in Mississauga, Ontario. If you'd like to be involved or would like to help sponsor these events, email:

rwot-leadership@googlegroups.com
