# Veres One DID Method 1.0

*A decentralized identifier method for the Veres One Ledger*

*Draft Community Group Report 01 October 2017*

by Manu Sporny, Digital Bazaar & Dave Longley, Digital Bazaar

**ABSTRACT**

The Veres One Ledger is a permissionless public ledger designed specifically for the creation and management of decentralized identifiers (DIDs). Veres One DIDs are self-sovereign identifiers that may be used by people, organizations, and digital devices to establish an identifier that is under their control. Veres One DIDs are useful in ecosystems where one needs to issue, store, and use Verifiable Claims. This specification defines how a developer may create and update DIDs in the Veres One Ledger.

**STATUS OF THIS DOCUMENT**

This specification was published by the Credentials Community Group. It is not a W3C Standard nor is it on the W3C Standards Track. Please note that under the W3C Community Contributor License Agreement (CLA) there is a limited opt-out and other conditions apply. Learn more about W3C Community and Business Groups.

Comments regarding this document are welcome. Please file issues directly on GitHub, or send them to public-credentials@w3.org (subscribe, archives).

Work on this specification has been funded in part by the United States Department of Homeland Security's Science and Technology Directorate under contract HSHQDC-17-C-00019. The content of this specification does not necessarily reflect the position or the policy of the U.S. Government and no official endorsement should be inferred.

Work on this specification has also been supported by the Rebooting the Web of Trust group facilitated by Christopher Allen, Shannon Appelcline, Kiara Robles, Kaliya Young, Brian Weller, and Betty Dhamers.

If you wish to make comments regarding this document, please send them to public-credentials@w3.org (subscribe, archives).

**TABLE OF CONTENTS**


- 1. Introduction
- 2. Core Data Model
- 3. Basic Concepts
  - 3.1 Authentication
  - 3.2 Authorization
  - 3.3 Service Descriptions
- 4. Operations
  1. 4.1 Discovering Service Endpoints
  2. 4.2 Creating a DID
  3. 4.3 Updating a DID Document
  4. 4.4 Delegating Control
  5. 4.5 Key Rotation and Transferring Control
  6. 4.6 Recovering a DID
- 5. Appendix A: Examples
  - 5.1 Typical DID Document
  - 5.2 Legacy DID Document


## 1. INTRODUCTION

Issue 1

TBD: This section will provide a gentle introduction to the purpose of the Veres One Ledger, expanding upon the abstract of the document.

## 2. CORE DATA MODEL

Issue 2

TBD: This section will describe the use of the Web Ledger, JSON-LD, and the DID spec to build the Veres One Ledger.

## 3. BASIC CONCEPTS

### 3.1 Authentication

Authentication is the process the ledger uses to determine if an entity is associated with a DID.

*Example 1: Expressing authentication credentials*

```
{
  "@context": "https://w3id.org/veres-one/v1",
  "id": "did:v1:215cb1dc-1f44-4695-a07f-97649cad9938",
  "authenticationCredential": [... array of acceptable authentication
```

```
credentials ...]
}
```

A detailed example of a valid set of authentication credentials follows:

*Example 2: Detailed example of authentication credentials entry*

```
{
  "@context": "https://w3id.org/veres-one/v1",
  "id": "did:v1:215cb1dc-1f44-4695-a07f-97649cad9938",
  "authenticationCredential": [{
    "type": "RsaSignature2017",
    "publicKey": {
      "id": "did:v1:215cb1dc-1f44-4695-a07f-97649cad9938/keys/2",
      "type": "CryptographicKey",
      "owner": "did:v1:215cb1dc-1f44-4695-a07f-97649cad9938",
      "publicKeyPem": "-----BEGIN PUBLIC KEY...END PUBLIC KEY-----\r\n",
    }
  }]
}
```

#### 3.2 Authorization

Authorization is the process the ledger uses to determine what an entity may to do the DID Document.

*Example 3*

```
{.nohighlight title=""}

{
  "@context": "https://w3id.org/veres-one/v1",
  "id": "did:v1:215cb1dc-1f44-4695-a07f-97649cad9938",
  "authorizationCapability": [... array of capability descriptions ...]
}
```

A detailed example of a valid set of authorization capability descriptions follows:

*Example 4*

```
{.nohighlight title=""}

{
  "@context": "https://w3id.org/veres-one/v1",
  "id": "did:v1:215cb1dc-1f44-4695-a07f-97649cad9938",
  // proof of update authorization may be provided by digital wallet + friend OR
  // by mobile phone
  "authorizationCapability": [{
    // this entity may update any field in this DID Document using any
    // authentication mechanism understood by the ledger
    "permission": "UpdateDidDocument",
```

```
    "entity": "did:v1:215cb1dc-1f44-4695-a07f-97649cad9938"
  }, {
    // this entity may update the authenticationCredential field in this
    // DID Document as long as they authenticate with RsaSignature2017
    "entity": "did:v1:b5f8c320-f7ca-4869-85e6-a1bcbf825b2a",
    "permission": "UpdateDidDocument",
    "field": ["authenticationCredential"],
    "permittedProofType": [{
      "proofType": "RsaSignature2017"
    }]
  }, {
    // anyone may update the authenticationCredential and writeAuthorization
    // fields as long as they provide a specific multi-signature proof
    "permission": "UpdateDidDocument",
    "field": ["authenticationCredential", "authorizationCapability"],
    "permittedProofType": [{
      "proofType": "RsaSignature2017",
      "minimumSignatures": 3,
      "authenticationCredential": [{
        "id": "did:v1:304ebc3e-7997-4bf4-a915-dd87e8455941/keys/123",
        "type": "RsaCryptographicKey",
        "owner": "did:v1:304ebc3e-7997-4bf4-a915-dd87e8455941",
        "publicKeyPem": "-----BEGIN PUBLIC KEY...END PUBLIC KEY-----\r\n"       0
      }, {
        "id": "did:v1:0f22346a-a360-4f3e-9b42-3366e348e941/keys/foo",
        "type": "RsaCryptographicKey",
        "owner": "did:v1:0f22346a-a360-4f3e-9b42-3366e348e941",
        "publicKeyPem": "-----BEGIN PUBLIC KEY...END PUBLIC KEY-----\r\n"
      }, {
        "id": "did:v1:a8d00377-e9f1-44df-a1b9-55072e13262a/keys/abc",
        "type": "RsaCryptographicKey",
        "owner": "did:v1:a8d00377-e9f1-44df-a1b9-55072e13262a",
        "publicKeyPem": "-----BEGIN PUBLIC KEY...END PUBLIC KEY-----\r\n"
      }]
    }]
}
```

#### 3.3 Service Descriptions

Services may be listed by including them at the top-level of the DID Document.

*Example 5: Simple example of a service description*

```
{
  "@context": "https://w3id.org/veres-one/v1",
  "id": "did:v1:215cb1dc-1f44-4695-a07f-97649cad9938",
   "credentialRepositoryService": "https://wallet.veres.io/"
}
```

A detailed example of the expression of a service description follows:

*Example 6*

```
{.nohighlight title=""}

{
  "@context": "https://w3id.org/veres-one/v1",
  "id": "did:v1:215cb1dc-1f44-4695-a07f-97649cad9938",
  "credentialRepositoryService": [{
    // the verifiable credential repository service
    "id": "did:v1:5d6c3b20-56a9-42e1-bfc8-ed7e685c9039",
    "type": "VerifiableCredentialRepository",
    "url": "https://wallet.veres.io/",
    "description": "Pat Doe's Digital Wallet"
  }]
}
```

#### 4. OPERATIONS

Every conforming Veres Ledger node *MUST* expose at least the following HTTP endpoints:

| Service | Example URL | Description |
|---|---|---|
| veresOneCreateService | POST /dids | Create a new DID. |
| veresOneReadService | GET /dids/{did} | Gets an existing DID Document. |
| veresOneUpdateService | POST /dids/{did} | Update an existing DID Document. |

#### 4.1 Discovering Service Endpoints

A website may provide service endpoint discovery by embedding JSON-LD in their top-most HTML web page (e.g. at `https://example.com/`):

*Example 7: Example of HTML-based service description*

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>Example Website</title>
    <link rel="stylesheet" href="style.css">
    <script src="script.js"></script>
    <script type="application/ld+json">
{
  "@context": "https://w3id.org/veres-one/v1",
  "id": "https://example.com/",
  "name": "Example Website",
  "veresOneCreateService": "https://example.com/veres-one/dids",
```

```
  "veresOneReadService": "https://example.com/veres-one/dids/",
  "veresOneUpdateService": "https://example.com/veres-one/dids/"
}
    </script>
  </head>
  <body>
    <!-- page content -->
  </body>
</html>
```

Service descriptions may also be requested via content negotiation. In the following example a JSON-compatible service description is provided (e.g. `curl -H "Accept: application/json" https://example.com/`):

*Example 8: Example of a JSON-based service description*

```
{
  "@context": "https://w3id.org/veres-one/v1",
  "id": "https://example.com/",
  "name": "Example Website",
  "veresOneCreateService": "https://example.com/veres-one/dids",
  "veresOneReadService": "https://example.com/veres-one/dids/",
  "veresOneUpdateService": "https://example.com/veres-one/dids/"
}
```

#### 4.2 Creating a DID

A DID is created by performing an HTTP POST of a signed DID Document to the `veresOneCreateService`. The following HTTP status codes are defined for this service:

HTTP Status

201

400

409

An example exchange of DID creation request is shown below:

*Example 9: DID creation request*

```
POST /dids HTTP/1.1
Host: example.com
Content-Type: application/ld+json
Content-Length: 1062
Accept: application/ld+json, application/json, text/plain, */*
Accept-Encoding: gzip, deflate

{
  "@context": "https://w3id.org/webledger/v1",
  "type": "WebLedgerEvent",
```

```
  "operation": "Create",
  "input": [{
    "@context": "https://w3id.org/veres-one/v1",
    "id": "did:v1:770f2d84-5e62-4caa-af95-111a3205bc84",
    "authorizationCapability": [{
      "permission": "UpdateDidDocument",
      "entity": "did:v1:770f2d84-5e62-4caa-af95-111a3205bc84",
      "permittedProofType": [{
        "proofType": "LinkedDataSignature2015"
      }, {
        "proofType": "EquihashProof2017",
        "equihashParameterAlgorithm": "VeresOne2017"
      }]
    }],
    "authenticationCredential": [{
      "id": "did:v1:770f2d84-5e62-4caa-af95-111a3205bc84/keys/1",
      "type": "CryptographicKey",
      "owner": "did:v1:770f2d84-5e62-4caa-af95-111a3205bc84",
      "publicKeyPem": "-----BEGIN PUBLIC KEY-----\r\n
        MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAmbDqPu6IKHiiIQ4d0AQ6\r\n
        PBduDhUUVqyQirvxqsdcNdKgZ2L8whBml/nTyuB4cd+hHrsfMDiHiT5kX2pbZ7Yy\r\n
        2ctWkGw8e0J94CbwVh2H15gBQBUCjLiGvVIHO2pni693qmre+3Ya2NJ8gGwPLJ7h\r\n
        TLca2b2dX0y16qu0MT0osUGGEoPsdg6ibD2pxnADS3GNPObHT12GrAuxjYFMHecF\r\n
        A4hLZ8U+MIcVmHZuokqqbcyJyjOV+kmhFNeTKFP5P5U8HA3Y42/rE1UJp/wyy7Lc\r\n
        ZAvq0t75ddXKyvYh5dkzxxeeELNKNWVxJ2yvgAr0SatLEPzxJoeYdCyU5N5E22Fj\r\n
        jQIDAQAB\r\n-----END PUBLIC KEY-----\r\n"
    }]
  }],
  "signature": [{
    "type": "EquihashProof2017",
    "equihashParameterN": 64,
    "equihashParameterK": 3,
    "nonce": "AQAAAA==",
    "proofValue": "AAAaPwABxrIAAFOKAAGo4QAAVW0AAN7cAACXcgABjEI="
  }, {
    "type": "LinkedDataSignature2015",
    "created": "2017-09-30T02:54:31Z",
    "creator": "did:v1:770f2d84-5e62-4caa-af95-111a3205bc84/keys/1",
    "signatureValue": "SNMbsPqLnB+hJFhXzS6hcpZnm9cGvSZZg7o26UYnyGYTvKder/S+Xk
      hNhXisS5385Ljlf5CXTQT5j6qYZtP8ut1Benaae8TMH17txP0CfzHbUMJFnHA1+Nru+e/Pw
      yPwuQ+VZYlXOB7g/tKVVZsxAYTKCAOJvJMIE+nlHjpB+RsKs9z4ZzVtddntqqAcvbZxV/o7
      azBFDizeJu/gHVVMncCJ00SRoOzCOZUABRJV/k68bNSAfpELkrdWx8/xvMIF8r+LWhwdKCS
      iOw4DjSwIK40yD5rOvQn/GlC+unyB8zFe60jCToz/UOJNZBiIYwo+Pwwx28Wqd4Jkb3IeDr
      /L2Q=="
  }]
}
```


If the creation of the DID was successful, an HTTP 201 status code is expected in return:

*Example 10: Successful DID creation response*

```
HTTP/1.1 201 Created
Location: https://ledger.example.com/dids/did:v1:215cb1dc-1f44-4695-a07f-
97649cad9938
Cache-Control: no-cache, no-store, must-revalidate
Pragma: no-cache
Expires: 0
Date: Fri, 14 Oct 2016 18:35:33 GMT
Connection: keep-alive
Transfer-Encoding: chunked
```

#### 4.3 Updating a DID Document

A DID is updated by performing an HTTP POST of a signed DID Document to the `veresOneUpdateService`. The following HTTP status codes are defined for this service:

| HTTP Status | Description |
|---|---|
| 201 | DID creation request was successful.<br>The HTTP `Location` header will contain the URL for the newly created DID Document. |
| 400 | DID creation request failed. |
| 409 | A duplicate DID exists. |

An example exchange for a DID update request is shown below:

*Example 11: DID Document update request*

```
POST /dids/did:v1:215cb1dc-1f44-4695-a07f-97649cad9938 HTTP/1.1
Host: example.com
Content-Type: application/ld+json
Content-Length: 1062
Accept: application/ld+json, application/json, text/plain, */*
Accept-Encoding: gzip, deflate

{
  "@context": "https://w3id.org/webledger/v1",
  "type": "WebLedgerEvent",
  "operation": "Update",
  "input": [{
    "@context": "https://w3id.org/veres-one/v1",
    "id": "did:v1:770f2d84-5e62-4caa-af95-111a3205bc84",
    "authorizationCapability": [{
      "permission": "UpdateDidDocument",
      "entity": "did:v1:770f2d84-5e62-4caa-af95-111a3205bc84",
      "permittedProofType": [{
        "proofType": "LinkedDataSignature2015"
      }, {
```

```
        "proofType": "EquihashProof2017",
        "equihashParameterAlgorithm": "VeresOne2017"
      }]
    }],
    "authenticationCredential": [{
      "id": "did:v1:770f2d84-5e62-4caa-af95-111a3205bc84/keys/1",
      "type": "CryptographicKey",
      "owner": "did:v1:770f2d84-5e62-4caa-af95-111a3205bc84",
      "publicKeyPem": "-----BEGIN PUBLIC KEY-----\r\n
        MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAmbDqPu6IKHiiIQ4d0AQ6\r\n
        PBduDhUUVqyQirvxqsdcNdKgZ2L8whBml/nTyuB4cd+hHrsfMDiHiT5kX2pbZ7Yy\r\n
        2ctWkGw8e0J94CbwVh2H15gBQBUCjLiGvVIHO2pni693qmre+3Ya2NJ8gGwPLJ7h\r\n
        TLca2b2dX0y16qu0MT0osUGGEoPsdg6ibD2pxnADS3GNPObHT12GrAuxjYFMHecF\r\n
        A4hLZ8U+MIcVmHZuokqqbcyJyjOV+kmhFNeTKFP5P5U8HA3Y42/rE1UJp/wyy7Lc\r\n
        ZAvq0t75ddXKyvYh5dkzxxeeELNKNWVxJ2yvgAr0SatLEPzxJoeYdCyU5N5E22Fj\r\n
        jQIDAQAB\r\n-----END PUBLIC KEY-----\r\n"
    }, {
      "id": "did:v1:770f2d84-5e62-4caa-af95-111a3205bc84/keys/2",
      "type": "CryptographicKey",
      "owner": "did:v1:770f2d84-5e62-4caa-af95-111a3205bc84",
      "publicKeyPem": "-----BEGIN PUBLIC KEY-----\r\n
        MIIBIj0BAQEFAKHiiIQ4d0AQ6ANBgkqhkiG9wAOCAQ8AMIIBCgKCAQEAmbDqPu6I\r\n
        xqsdcNdKgZ2L8whBml/nTyuBiHiPBduDhUUVqyQirvT5kX2pbZ7Yy4cd+hHrsfMD\r\n
        VhVIHO2pni693qmre+2ctWkGw8e0J94Cbw3Ya2NJ8gGwPLJ7hH15gBQBUCjLiGv\r\n
        0osUNPObHT12GrAuxjYFMHecFGTLca2b2dX0y16qu0MTGEoPsdg6ibD2pxnADS3G\r\n
        kqqbcyJyjOV+kmh8HA3Y42/rE1UJpA4hLZ8U+MIcVmHZuo/wyy7LcFNeTKFP5P5U\r\n
        5dkzxxezxJoeYdCyU5N5E2ZAvq0t75ddXKyvYh2FjeELNKNWVxJ2yvgAr0SatLEP\r\n
        AQABjQID\r\n-----END PUBLIC KEY-----\r\n"
    }]
  }],
  "signature": [{
    "type": "EquihashProof2017",
    "equihashParameterN": 64,
    "equihashParameterK": 3,
    "nonce": "AQAAAA==",
    "proofValue": "AAAaPwABxrIAAFOKAAGo4QAAVW0AAN7cAACXcgABjEI="
  }, {
    "type": "LinkedDataSignature2015",
    "created": "2017-09-30T02:54:31Z",
    "creator": "did:v1:770f2d84-5e62-4caa-af95-111a3205bc84/keys/2",
    "signatureValue": "Zg7o26UYnyGYTvKdSN6hcpZnm9cGvSZB+hJFhXzSer/S+XkMbsPqLn
      VMncCJ00SRoOzCOZUABRJV/azBFDizeJu/gHVk68bNSAfpELkrdWx8/xvMIF8r+LWhwdKCS
      XTQT5j6qYZtP8ut1BenahNhXisS5385Ljlf5Cae8TMH17txP0CfzHbUMJFnHA1+Nru+e/Pw
      K40yD5rOvQn/GlC+unyB8ziOw4DjSwIFe60jCToz/UOJNZBiIYwo+Pwwx28Wqd4Jkb3IeDr
      VVZsxAYTKCAOJvJMIE+nlHjpB+RyPwuQ+VZYlXOB7g/tKsKs9z4ZzVtddntqqAcvbZxV/o7
      /45H=="
  }]
}
```

If the update request for the DID was successful, an HTTP 200 status code is expected in return:

*Example 12: Successful ledger creation response*

```
HTTP/1.1 200 Success
Cache-Control: no-cache, no-store, must-revalidate
Pragma: no-cache
Expires: 0
Date: Fri, 14 Oct 2016 18:35:33 GMT
Connection: keep-alive
Transfer-Encoding: chunked
```

#### 4.4 Delegating Control

Issue 3

TBD: Explain that delegation of control is merely placing a digital wallet provider in the proofOfControl field.

#### 4.5 Key Rotation and Transferring Control

Issue 4

TBD: Explain that transferring control and rotating keys is a matter of adding and removing the appropriate keys from proofofControl and proofOfUpdateAuthorization.

#### 4.6 Recovering a DID

Issue 5

TBD: Explain that recoverying a DID is a matter of meeting the requirements under proofOfUpdateAuthorization.

## 5. APPENDIX A: EXAMPLES

### 5.1 Typical DID Document

The following is a complete example of a typical Veres One DID Document:

*Example 13*

```
{
  "@context": "https://w3id.org/veres-one/v1",
  "id": "did:v1:eaaf4df5-471d-404e-b143-652fe38cd2c7",
  "authorizationCapability": [{
    "permission": "UpdateDidDocument",
    "entity": "did:v1:eaaf4df5-471d-404e-b143-652fe38cd2c7",
    "permittedProofType": [{
      "proofType": "LinkedDataSignature2015"
    }, {
```

```
      "proofType": "EquihashProof2017",
      "equihashParameterAlgorithm": "VeresOne2017"
    }]
  }],
  "authenticationCredential": [{
    "id": "did:v1:eaaf4df5-471d-404e-b143-652fe38cd2c7/keys/1",
    "type": "CryptographicKey",
    "owner": "did:v1:eaaf4df5-471d-404e-b143-652fe38cd2c7",
    "publicKeyPem": "-----BEGIN PUBLIC KEY-----\r\n
      MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAvZXq8jX38lwndvzadCsT\r\n
      Xa2Zafdrg9I69gzfccH6XWY3Ddi/JoMuTSB1GwxKBfXpo9gjaPYsm6wCLv9Kku4x\r\n
      HL4LA1kmIalVTVDYgSO4sGK9k9oQNTY+hgUoTtdMxMShWrVy6+DIS/ZzIPyQBtbm\r\n
      9D7RojrvESmjq/OuMs6sTlC0JjEE1ijuuHY+iY7gDYcR7RGFAsi4WGbCVy6c8VqL\r\n
      29h8yGps2U+AxKr9f783VGMCk469ESHVwVyw6Jbxihn/h4TH3ZH8WTQW9rpS9GhO\r\n
      euSAA6iSH5UcmAzJZKSzaC+oghEJwMtOcgvr1F9iSn9tuHebgy9R6tHvEChhvdgz\r\n
      2wIDAQAB\r\n
      -----END PUBLIC KEY-----\r\n",
  }]
}
```

#### 5.2 Legacy DID Document

The Veres One ledger was launched in 2015, predated this specification, and as a result has a number of legacy objects that developers should be aware of. The typical format for these objects are shown below:

*Example 14*

```
{
  "@context": "https://w3id.org/identity/v1",
  "id": "did:8743453f-e45e-4ac6-b85f-4513ba4c1460",
  "idp": "did:d1d1d1d1-d1d1-d1d1-d1d1-d1d1d1d1d1d1",
  "accessControl": {
    "writePermission": [
      {
        "id": "did:8743453f-e45e-4ac6-b85f-4513ba4c1460/keys/1",
        "type": "CryptographicKey"
      },
      {
        "id": "did:d1d1d1d1-d1d1-d1d1-d1d1-d1d1d1d1d1d1",
        "type": "Identity"
      }
    ]
  },
  "publicKey": [
    {
      "id": "did:8743453f-e45e-4ac6-b85f-4513ba4c1460/keys/1",
      "type": "CryptographicKey",
      "owner": "did:8743453f-e45e-4ac6-b85f-4513ba4c1460",
```

```
      "publicKeyPem": "-----BEGIN PUBLIC KEY-----\r\nMIIBIjA...DAQAB\r\n-----END
PUBLIC KEY-----\r\n",
      "@context": "https://w3id.org/identity/v1"
    }
  ],
  "signature": {
    "type": "LinkedDataSignature2015",
    "created": "2017-07-25T17:29:49Z",
    "creator": "did:8743453f-e45e-4ac6-b85f-4513ba4c1460/keys/1",
    "signatureValue": "LJoxpV...daOLHbA=="
  }
}
```

**LATEST EDITOR'S DRAFT**

**https://w3c-ccg.github.io/didm-veres-one/**

**PARTICIPATE**

GitHub w3c-ccg/didm-veres-one

File a bug

Commit history



---

**ADDITIONAL CREDITS**

**Editors & Authors:** Manu Sporny, Digital Bazaar & Dave Longley, Digital Bazaar

---

**About Rebooting the Web of Trust**

This paper was produced as part of the Rebooting the Web of Trust V design workshop. On October 3rd through October 5th, 2017, over 50 tech visionaries came together in Cambridge, Massachusetts 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.

**Preliminary Workshop Sponsors List:** BigChainDB, Blockchain Lab, Digital Contract Design, IDEO, IPFS, Protocol Labs, Toni Lane Casserly

**Workshop Producer:** Christopher Allen

**Workshop Facilitators:** Christopher Allen, with additional paper editorial & layout by Shannon Appelcline.

**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-fall2017/issues

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

ChristopherA@LifeWithAlacrity.com

---