WebAC Authorization Delegate

The Fedora WebAC authorization module is an implementation of the W3C's still evolving draft of an RDF-based decentralized authorization policy mechanism.

W3C's definition of WebAccessControl

From the [WebAccessControl description](https://www.w3.org/2001/02/ServiceAccessControl) at the W3C website:

> WebAccessControl is a decentralized system for allowing different users and groups various forms of access to resources where users and groups are identified by HTTP URIs.

The WebAC module will enforce access control based on the Access Control List (ACL) RDF resource associated with the requested resource. In WebAC, an ACL consists of a set of Authorizations. Each Authorization is a single rule for access, such as "users alice and bob may write to resource foo", described with a set of RDF properties. Authorizations have the RDF type `http://www.w3.org/ns/auth/acl#Authorization`.

For the remainder of this document, the `http://www.w3.org/ns/auth/acl#` namespace will be abbreviated with the prefix acl:

Authorizations

The properties that may be used on an acl:Authorization are:

<table>
<thead>
<tr>
<th>Property</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>acl:accessTo</td>
<td>the URI of the protected resource</td>
</tr>
<tr>
<td>acl:agent</td>
<td>the user <em>(in the W3C WebAC ontology, the user is named with a URI, but Fedora's implementation uses string usernames instead)</em></td>
</tr>
<tr>
<td>acl:mode</td>
<td>the type of access <em>(WebAC defines several modes: acl:Read, acl:Write, acl:Append, and acl:Control; Fedora implements acl:Read and acl:Write)</em></td>
</tr>
<tr>
<td>acl:accessToClass</td>
<td>an RDF class of protected resources</td>
</tr>
<tr>
<td>acl:agentClass</td>
<td>a group of users (defined as a foaf:Group resource listing its users with the foaf:member property)</td>
</tr>
</tbody>
</table>

For a more detailed explanation of Authorizations and their properties, see [WebAC Authorizations](https://www.w3.org/2001/02/ServiceAccessControl).

Examples of Authorizations

1. The user userA can Read document foo

   ```
   @prefix acl: <http://www.w3.org/ns/auth/acl#> .
   <> a acl:Authorization ;
   acl:accessTo </foo> ;
   acl:mode acl:Read;
   acl:agent "userA" .
   ```

2. Users in NewsEditor group can Write to any resource of type ex:News

   ```
   @prefix acl: <http://www.w3.org/ns/auth/acl#> .
   @prefix ex: <http://example.org/ns#> .
   <> a acl:Authorization ;
   acl:accessToClass ex:News ;
   acl:mode acl:Read, acl:Write;
   acl:agentClass </agents/NewsEditor> .
   ```
Storing WebAC ACLs in Fedora 4

In Fedora 4, an ACL is a ldp::BasicContainer resource with the additional RDF type of http://fedora.info/definitions/v4/webac#Acl. This class is part of the Fedora WebAC ontology. Its children should each be resources of type acl:Authorization. It is given the namespace prefix webac: by convention.

Protecting Resources

A resource specifies the location of its ACL using the acl:accessControl property. If a resource itself does not specify an ACL, its parent containers are inspected, and the first specified ACL found is used as the ACL for the requested resource. If no ACLs are found, a filesystem-based ACL will be checked, the default policy of which is to deny access to the requested resource.

Example Scenarios

1. I want to allow a user with username "smith123" to have read, write access to resource http://localhost:8080/rest/webacl_box1.

   Using the two "files" below to create our Authorization and ACL resources.

   **Acl.ttl**

   ```turtle
   @prefix webac: <http://fedora.info/definitions/v4/webac#> .
   <> a webac:Acl .
   ```

   **Authorization.ttl**

   ```turtle
   @prefix acl: <http://www.w3.org/ns/auth/acl#> .
   <> a acl:Authorization ;
   acl:agent "smith123" ;
   acl:mode acl:Read, acl:Write ;
   ```

   We would execute the following commands.

   ```
   > echo "PREFIX acl: <http://www.w3.org/ns/auth/acl#>
   INSERT DATA {
   ```

2. I want to let the group "Editors" have read, write access on all the items in the collection "http://localhost:8080/rest/box/bag/collection"

   Using the two "files" below to create our Authorization and ACL resources.

   **An example**

   ```turtle
   @prefix foaf: <http://xmlns.com/foaf/0.1/> .
   <> a foaf:Group;
   foaf:member "editor1", "editor2".
   ```
We would execute the following commands.

```bash
http://localhost:8080/rest/acl
http://localhost:8080/rest/acl/auth1
> echo "PREFIX acl: <http://www.w3.org/ns/auth/acl#> 
INSERT DATA {
```

3. I would like the collection `http://localhost:8080/rest/dark/archive` to be viewable only by the groupid "Restricted", but I would like to allow anyone to view the resource `http://localhost:8080/rest/dark/archive/sunshine`.

Using the three "files" below to create our Authorization and ACL resources.

---

**Acl.ttl**

```turtle
@prefix webac: <http://fedora.info/definitions/v4/webac#> .
<> a webac:Acl .
```

**Authorization.ttl**

```turtle
@prefix acl: <http://www.w3.org/ns/auth/acl#> .
<> a acl:Authorization ;
  acl:agent "Editors" ;
  acl:mode acl:Read, acl:Write ;
```

**Auth_restricted.ttl**

```turtle
@prefix acl: <http://www.w3.org/ns/auth/acl#> .
<> a acl:Authorization ;
  acl:agent "Restricted" ;
  acl:mode acl:Read ;
```

**Auth_open.ttl**

```turtle
@prefix acl: <http://www.w3.org/ns/auth/acl#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
<> a acl:Authorization ;
  acl:agent foaf:Agent ;
  acl:mode acl:Read ;
  acl:accessTo <http://localhost:8080/rest/dark/archive/sunshine> .
```
The I would execute the following commands.

```bash
http://localhost:8080/rest/acl_lock

> curl -X PUT -H "Content-type: text/turtle" --data-binary "@Auth_restricted.ttl" "http://localhost:8080/rest/acl_lock/auth1"
http://localhost:8080/rest/acl_lock/auth1

> echo "PREFIX acl: <http://www.w3.org/ns/auth/acl#>
INSERT DATA {
<> acl:accessControl <http://localhost:8080/rest/acl_lock> .

http://localhost:8080/rest/acl_open

http://localhost:8080/rest/acl_open/auth1

> echo "PREFIX acl: <http://www.w3.org/ns/auth/acl#>
INSERT DATA {
```

4. The collection `http://localhost:8080/rest/public_collection` should be readable by anyone but only editable by users in the group Editors.

Using the three "files" below to create our Authorization and ACL resources.

### Acl.ttl

```turtle
@prefix webac: <http://fedora.info/definitions/v4/webac#> .
<> a webac:Acl .
```

### Auth1.ttl

```turtle
@prefix acl: <http://www.w3.org/ns/auth/acl#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
<> a acl:Authorization ;
  acl:agent foaf:Agent ;
  acl:mode acl:Read ;
```

### Auth2.ttl

```turtle
@prefix acl: <http://www.w3.org/ns/auth/acl#> .
<> a acl:Authorization ;
  acl:agent "Editors" ;
  acl:mode acl:Read, acl:Write ;
```

I would execute the following code:
Using the three "files" below to create our Authorization and ACL resources.

### Acl.ttl

```turtle
@prefix webac: <http://fedora.info/definitions/v4/webac#> .
<> a webac:Acl .
```

### Auth_restricted.ttl

```turtle
@prefix acl: <http://www.w3.org/ns/auth/acl#> .
<> a acl:Authorization ;
  acl:agent 'Admins' ;
  acl:mode acl:Read ;
  acl:accessTo <http://localhost:8080/rest/mixedCollection> .
```

### Auth_open.ttl

```turtle
@prefix acl: <http://www.w3.org/ns/auth/acl#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
<> a acl:Authorization ;
  acl:agent foaf:Agent ;
  acl:mode acl:Read ;
  acl:accessToClass ex:publicImage .
```

I would execute the following commands:

```shell
http://localhost:8080/rest/acl

http://localhost:8080/rest/acl/auth1

http://localhost:8080/rest/acl/auth2

> echo "PREFIX acl: <http://www.w3.org/ns/auth/acl#>
INSERT DATA {
}"
```

5. Only the ex:publicImage type objects in the container http://localhost:8080/rest/mixedCollection are viewable by anyone, all others are only viewable by the group Admins.

http://localhost:8080/rest/acl/auth1

http://localhost:8080/rest/acl/auth2

http://localhost:8080/rest/acl/auth2
> echo "PREFIX acl: <http://www.w3.org/ns/auth/acl#>
  INSERT DATA {<>

How-To Guides

- Quick Start with WebAC
- How to Use WebAC agentClass Groups

More Detailed Documentation

- Determining the Effective Authorization Using WebAC
- W3C's WebAC Ontology