LDP Position from Oracle Fusion Applications Team

Linked Data Platform Workshop
San Francisco, CA
21 April 2015

Oshani Seneviratne
Senior Software Engineer
Fusion HCM Development
Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle’s products remains at the sole discretion of Oracle.
Agenda

1. Motivations for using LDP
2. LDP in Oracle Software
3. Proposals for the Next LDP Recommendation
4. Discussion
Agenda

1 Motivations for using LDP
2 LDP in Oracle Software
3 Proposals for the Next LDP Recommendation
4 Discussion
Motivations

• Solve the data integration challenge
• Provide an abstraction for vendor specific APIs through a standards compliant interface
• Delegate the key/token management from the Oracle Fusion apps to customer data aggregators
Agenda

1. Motivations for using LDP
2. LDP in Oracle Software
3. Proposals for the Next LDP Recommendation
4. Discussion
Data Sources and Oracle Software

- Many Oracle Fusion Applications in the HCM Cloud consume data from variety of sources:

<table>
<thead>
<tr>
<th>Oracle Fusion Application</th>
<th>Data Sources</th>
</tr>
</thead>
<tbody>
<tr>
<td>Workforce Reputation Management</td>
<td>Social APIs such as Facebook, LinkedIn, Twitter</td>
</tr>
<tr>
<td>Wellness</td>
<td>Fitbit and other IoT APIs</td>
</tr>
<tr>
<td>HCMConnect</td>
<td>SMTP, SFTP, Atom, Web Services, and Enterprise Applications such as Taleo, NetSuite, Payroll</td>
</tr>
</tbody>
</table>

- We have come up with an intermediate system architecture using LDP, until the data vendors provide their data through LDP.
LDP usage in Oracle HCM

Data Cloud
- Facebook Connector
- LinkedIn Connector

Customer Cloud
- Custom Connectors:
  - Facebook Connector
  - LinkedIn Connector
- LDP Server
- DATA Store

Oracle HCM Cloud
- Our Connectors:
  - LDP Connector
  - SFTP Connector
  - SMTP Connector
- Scheduler Service
- Oracle Schema

Oracle and/or its affiliates. All rights reserved.
LDP usage in Oracle HCM
Creating Resources on the LDP Server

Data Cloud
- Facebook
- LinkedIn
- Notion
- Tableau
- ATOM
- Web Service
- UCM
- SFTP
- CSD
- SMTP

Customer Cloud
- Custom Connectors
- Facebook Connector
- LinkedIn Connector
- LDP Server

Oracle HCM Cloud
- Our Connectors
  - LDP Connector
  - SFTP Connector
  - SMTP Connector
  - Scheduler Service

1. Pull
2. PUT / POST
3. Write
LDP usage in Oracle HCM
Retrieving LDP Resources

Data Cloud
- Facebook
- LinkedIn
- Outlook
- Salesforce
- UCM

Customer Cloud
- Custom Connectors
  - Facebook Connector
  - LinkedIn Connector

Oracle HCM Cloud
- Our Connectors
  - LDP Connector
  - SFTP Connector
  - SMTP Connector

Scheduler Service
- Runs
- Get Resource
- Retrieve
- Delete (optional)
- Update

Oracle Schema

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.
LDP usage in Oracle HCM
Updates from the HCM Cloud

Data Cloud

Customer Cloud

Custom Connectors

Oracle HCM Cloud

Our Connectors

Scheduler Service
LDP Containment Hierarchy:
Different Social Profiles Managed within LDP Containers
Relationships from the external data sources are represented in the containers
Inferences made at the Global Profile Level based on relationships from external data sources
Agenda

1. Motivations for using LDP
2. LDP in Oracle Software
3. Proposals for the Next LDP Recommendation
4. Discussion
Overview of Proposals

1. Access Control
2. Multiple Containment Hierarchies
3. Push Capabilities
4. Resource Filtering Semantics
1. Access Control

• LDP Servers should be able to handle:
  – Authentication
  – Authorization Policies
  – Usage Policies
Authentication

- The auth token includes attributes or roles of the agent.
- The LDP Server can indicate that the resource requested is protected in the response.

<table>
<thead>
<tr>
<th>Request</th>
<th>Response</th>
</tr>
</thead>
</table>
| GET /alice HTTP/1.1  
Host: oracle.com  
Accept: text/turtle  
Authorization: Bearer token with attributes or roles of the agent | HTTP/1.1 200 OK  
Content-type: text/turtle  
Link: <ldp:ProtectedContainer>  
rel="type"  
...  
<triples> |
Authentication contd.

- If invalid token or no token, the server has two options:
  - Send a 401 Unauthorized Response.
  - Send a 303 See Other Response redirecting to an alternate public representation of the resource.
- The server may optionally include an explanation in both cases.

**Request**

```
GET /alice HTTP/1.1
Host: oracle.com
Accept: text/turtle
```

**Response**

```
HTTP/1.1 303 See Other
Location /alice/public
Content-type: text/turtle
Link: <ldp:BasicContainer> rel="type"
...
<alice> a <ldp:ProtectedContainer>,
<ldp:policy> <must_authenticate_policy>.
```
Authorization Policies

• Specify that a container/resource is subject to an authorization policy.

    Resource / Container Representation

    <alice> a ldp:Container, ldp:ProtectedContainer;
    ldp:policy <alice_access_policy> .

• The authorization policy itself can be an LDP container.

    Policy

    <alice_access_policy> a ldp:Container;
    acl:accessTo <alice>;
    acl:mode acl:Read, acl:Write, acl:Control;
    acl:agent <HCM_admin_role> .

• If no policy is specified, the parent policy should take effect.
Authentication and Authorization: Terminology
Authentication and Authorization: Application Init

Data Cloud

Customer Cloud

Custom Connectors
- Facebook Connector
- LinkedIn Connector

Oracle HCM Cloud

Our Connectors
- LDP Connector
- SFTP Connector
- SMTP Connector

1. Auth Request
2. Token
Authentication and Authorization: Requesting Resource
Usage Policies

• Similar to Authorization policies.
• But rather than controlling access to the resources, these policies specify how the resource may be used. For Example: Creative Commons Licenses

Usage Policy

<alice_usage_policy> a ldp:Container;
   cc:license cc:by-nd .

• The LDP server should return the usage policy description in the response payload.
Access Control Summary: Our Suggestions

• Introduce the following concepts:
  – ldp:ProtectedContainer, ldp:ProtectedResource
  – ldp:PolicyContainer (optional)
  – ldp:policy

• Introduce 303 redirects when the policy is not based on the auth credentials provided by the agent.
  – Provide a public representation of the container or the resource.
  – Optionally provide any explanation as to why the requested resource is not available.
Access Control Summary: Our Suggestions contd.

• When a policy is **not** specified for an LDP container / resource:
  – The parent container’s policy should take effect.
  – For every Protected Resource / Container the parent relationship should be made explicit.

  **Resource / Container Representation**

  ```xml
  <alice> a ldp:Container, ldp:ProtectedContainer;
  ldp:parent <global_profiles> .
  ```

  – **ldp:parent** is the inverse functional property of ldp:contains.
    • Provides a back-link to the parent
2. Multiple Membership Containers for Resources

• In certain cases, we need a resource to have membership in more than one container. i.e:

  ParentContainer1  ldp:contains  Resource1
  ParentContainer2  ldp:contains  Resource1

• This can aid in resource and container discovery based on a certain topic, alternate categorization, etc.

• Here is an example:
LDP Containment Hierarchy: Different Social Profiles Managed within LDP Containers

- People Container
  - Global Profile Container
    - LinkedIn Container
      - Profile
      - Skills
      - Endorsements
      - Posts
    - Facebook Container
      - Profile
      - Depiction
      - Posts
    - Twitter Container
      - Profile
      - Tweets
      - Followers
Different Social Profiles Managed within **Multiple** LDP Containers

- People Container
- Global Profile Container
- Facebook Container
- Twitter Container
- LinkedIn Container

- Posts Container
- Followers
- Tweets
- Profile
- Skills
- Endorsements
- Depiction

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.
# Creating a New Container with Existing Resources

<table>
<thead>
<tr>
<th>Request</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>POST /people HTTP/1.1</td>
<td>HTTP/1.1 201 Created</td>
</tr>
<tr>
<td>Host: oracle.com</td>
<td>Location: <a href="http://oracle.com/people/posts/">http://oracle.com/people/posts/</a></td>
</tr>
<tr>
<td>Link: <a href="">ldp:Container</a>; rel=&quot;type&quot;</td>
<td>Link: <a href="">ldp:Container</a> rel=&quot;type&quot;</td>
</tr>
<tr>
<td>Slug: posts</td>
<td>Content-Length: 0</td>
</tr>
<tr>
<td>&lt;&gt; a ldp:Container;</td>
<td></td>
</tr>
<tr>
<td>dcterms:title “All Posts”;</td>
<td></td>
</tr>
<tr>
<td>ldp:contains</td>
<td></td>
</tr>
<tr>
<td><a href="http://oracle.com/people/linkedin/posts">http://oracle.com/people/linkedin/posts</a>,</td>
<td></td>
</tr>
<tr>
<td><a href="http://oracle.com/people/facebook/posts">http://oracle.com/people/facebook/posts</a></td>
<td></td>
</tr>
</tbody>
</table>
## Adding Existing Resources to another Existing Container

<table>
<thead>
<tr>
<th>Request</th>
<th>Response</th>
</tr>
</thead>
</table>
| POST /people/posts HTTP/1.1  
Host: oracle.com  
Link: &lt;ldp:Container&gt;; rel="type"  

&gtrless; ldp:contains  
&lt;http://oracle.com/people/twitter/tweets&gt; . | HTTP/1.1 200 OK |
Multiple LDP Containment: Caveats

If an existing resource from a different domain/origin is added:

• There can be access control issues, especially if the resource is protected
• Complications on resource ownership

⇒ Solution: Only Allow Same Origin Multiple LDP Containment

If not validated on insert, this feature can lead to cyclic containment!

⇒ Solution: If ldp:contains is in the payload, prompt the LDP Server to check for cycles.
Multiple LDP Containment: Caveats contd.

If the resource and the parent containers are of same domain/origin, but:

• The parents have different access control policies to that of the child resource:
  => Solution: The child resource’s access control policy takes precedence

• The child resource has no policy, and:
  – The parents have different access control policies:
    => Solution: The child resource inherits the most restrictive policy.
  – The parents have conflicting access control policies:
    => Solution: Raise an exception or send 405 Method Not Allowed
3. Push Capabilities

- Some resources/containers on the LDP Server can update frequently.
- In such cases, the consumers, such as the LDP Connector, can attach a triple to the resource/container to indicating that it is an observer of that LDPRS.

  Container Definition with an Observer

  ```xml
  <alice> a ldp:BasicContainer;
  ```

- When new triples are added, or when the existing triples are changed, the LDP Server can send POST requests to the specified observer.

- Additional Authorization Privilege ‘observe’:
  - Enable observers to write to the container or resource observed
  - Make the promise to the observers to send HTTP Post requests
Push Capabilities: Registering An Observer

Data Cloud
- Facebook Connector
- LinkedIn Connector

Customer Cloud
- Custom Connectors
  - Facebook Connector
  - LinkedIn Connector

Oracle HCM Cloud
- Our Connectors
  - LDP Connector
  - SFTP Connector
  - SMTP Connector

1. Insert Observer + Token Request
2. Evaluate observer authz policy
3. Scheduler Service
Push Capabilities: Receiving Updates

1. Pull Data
2. Update Resource
3. Find Observers for Resource
4. Post Updates

Data Cloud
- Facebook Connector
- LinkedIn Connector

Customer Cloud
- Custom Connectors
  - Facebook Connector
  - LinkedIn Connector
- LDP Server
- Observer

Oracle HCM Cloud
- Our Connectors
  - LDP Connector
  - SFTP Connector
  - SMTP Connector
- Scheduler Service

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.
4. Resource Filtering Semantics

- What is supported now:

<table>
<thead>
<tr>
<th>Request</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET /alice HTTP/1.1 Host: oracle.com Accept: text/turtle Prefer: return=representation include=&quot;ldp:PreferMinimalContainer&quot;</td>
<td>HTTP/1.1 200 Ok Content-type: text/turtle Link: <a href="">ldp:BasicContainer</a> rel=&quot;type&quot; ...</td>
</tr>
<tr>
<td></td>
<td>&lt;alice&gt; a <a href="">ldp:BasicContainer</a>, ...</td>
</tr>
</tbody>
</table>

- Remove the prefix ‘Prefer’ from the containment, membership and minimal-container triples for clarity in the use with the ‘omit’ parameter.

- Include additional resource query and filter semantics:
  - options such as:
    - ldp:ChildContainers
    - ldp:BinaryResources
    - ldp:CCBYLicencedResources
Agenda

1. Motivations for using LDP
2. LDP in Oracle Software
3. Proposals for the Next LDP Recommendation
4. Discussion
Questions?

oshani.seneviratne@oracle.com