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

Agenda

ORACLE®

Agenda

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

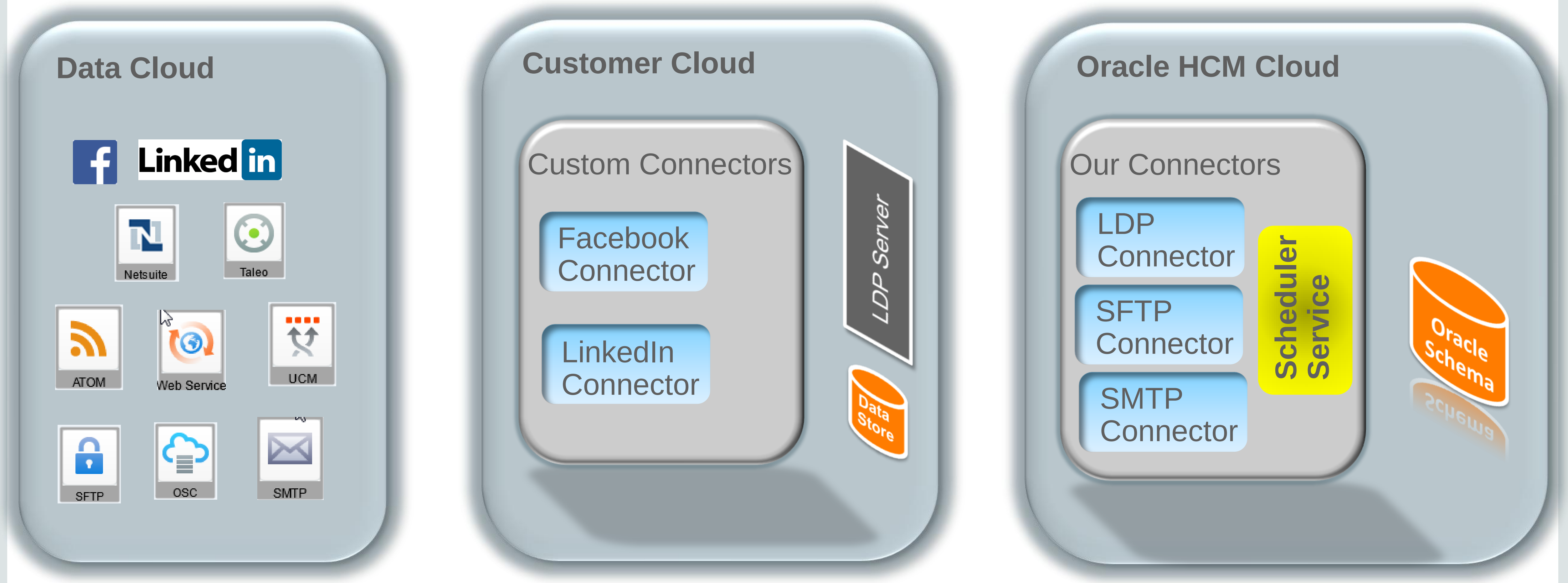
Data Sources and Oracle Software

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

Oracle Fusion Application	Data Sources
Workforce Reputation Management	Social APIs such as Facebook, LinkedIn, Twitter
Wellness	Fitbit and other IoT APIs
HCMConnect	SMTP, SFTP, Atom, Web Services, and Enterprise Applications such as Taleo, NetSuite, Payroll

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

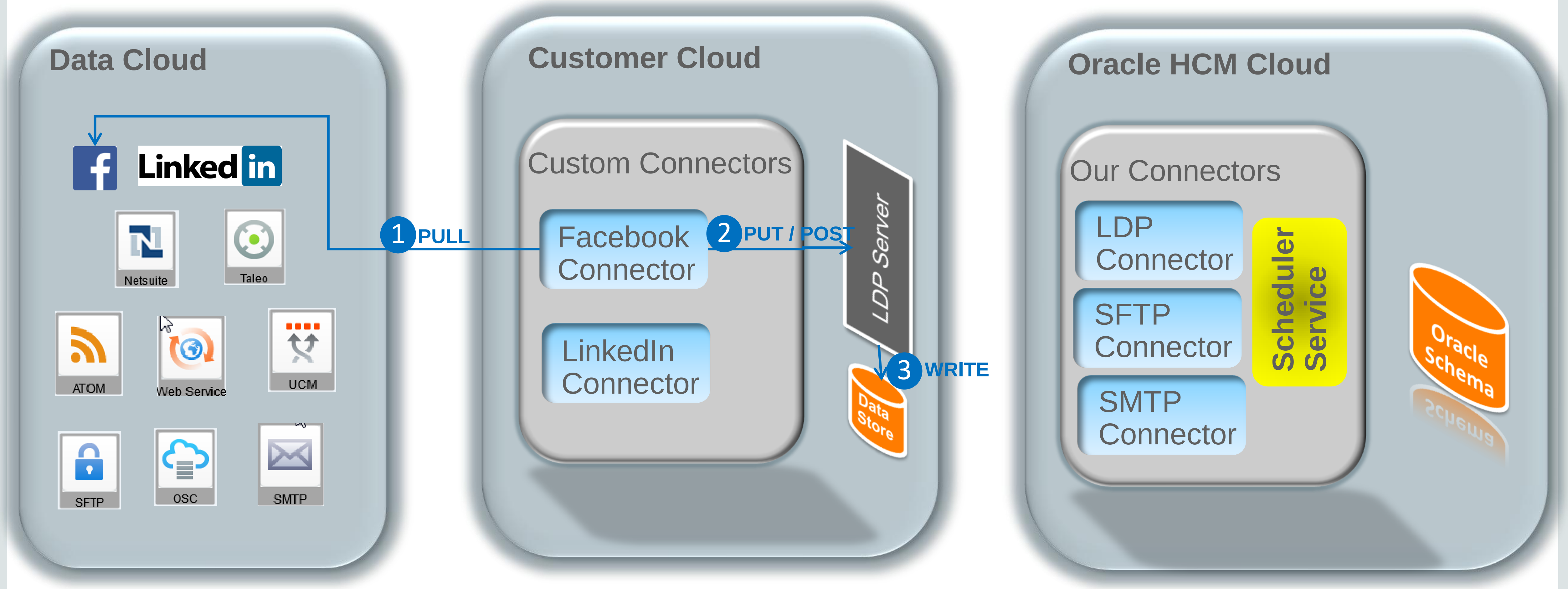
LDP usage in Oracle HCM



ORACLE

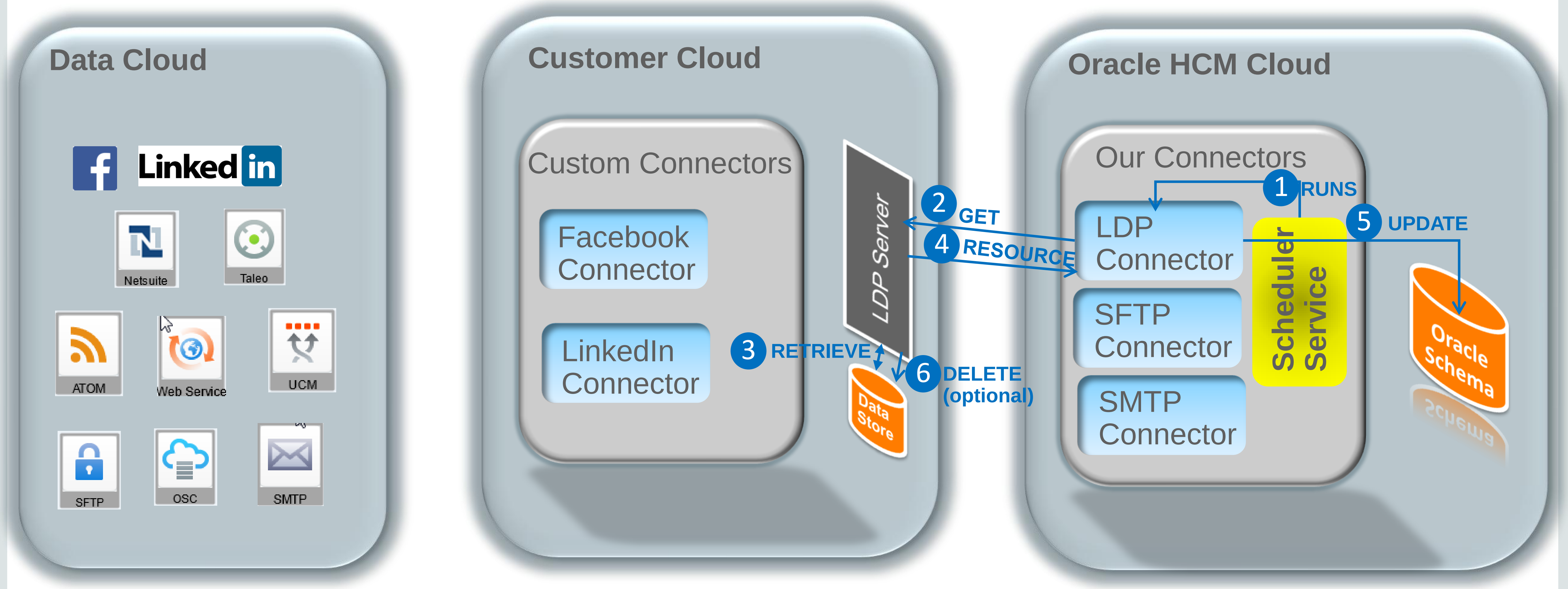
LDP usage in Oracle HCM

Creating Resources on the LDP Server

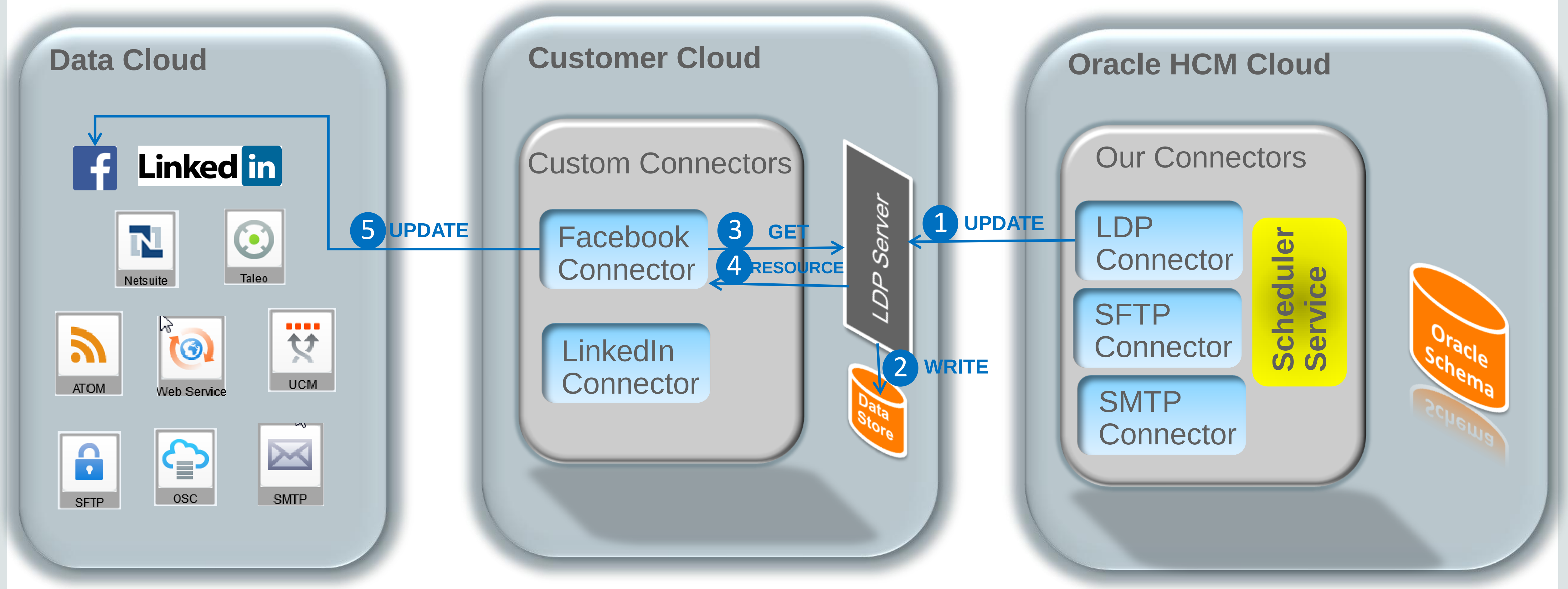


LDP usage in Oracle HCM

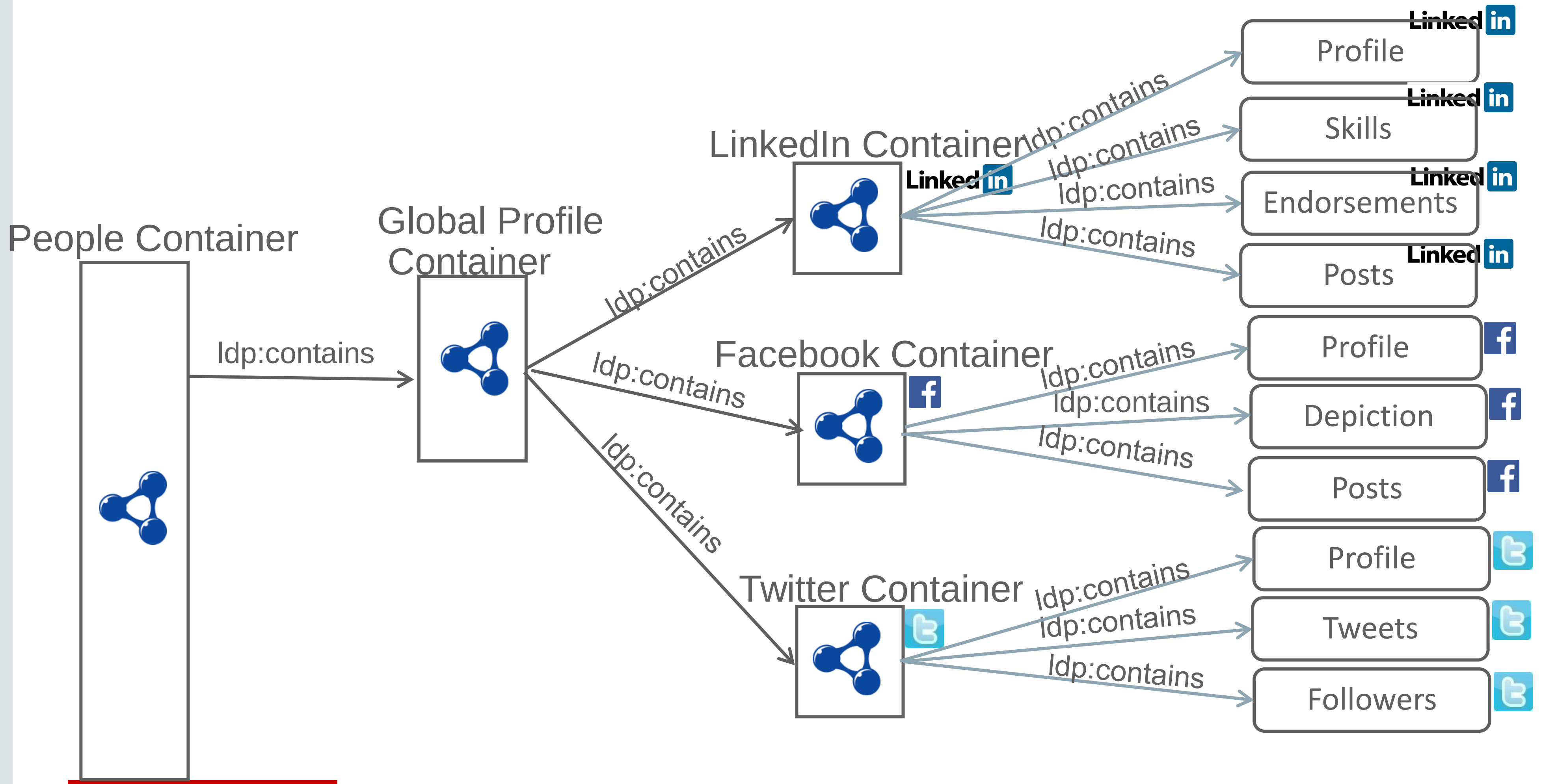
Retrieving LDP Resources



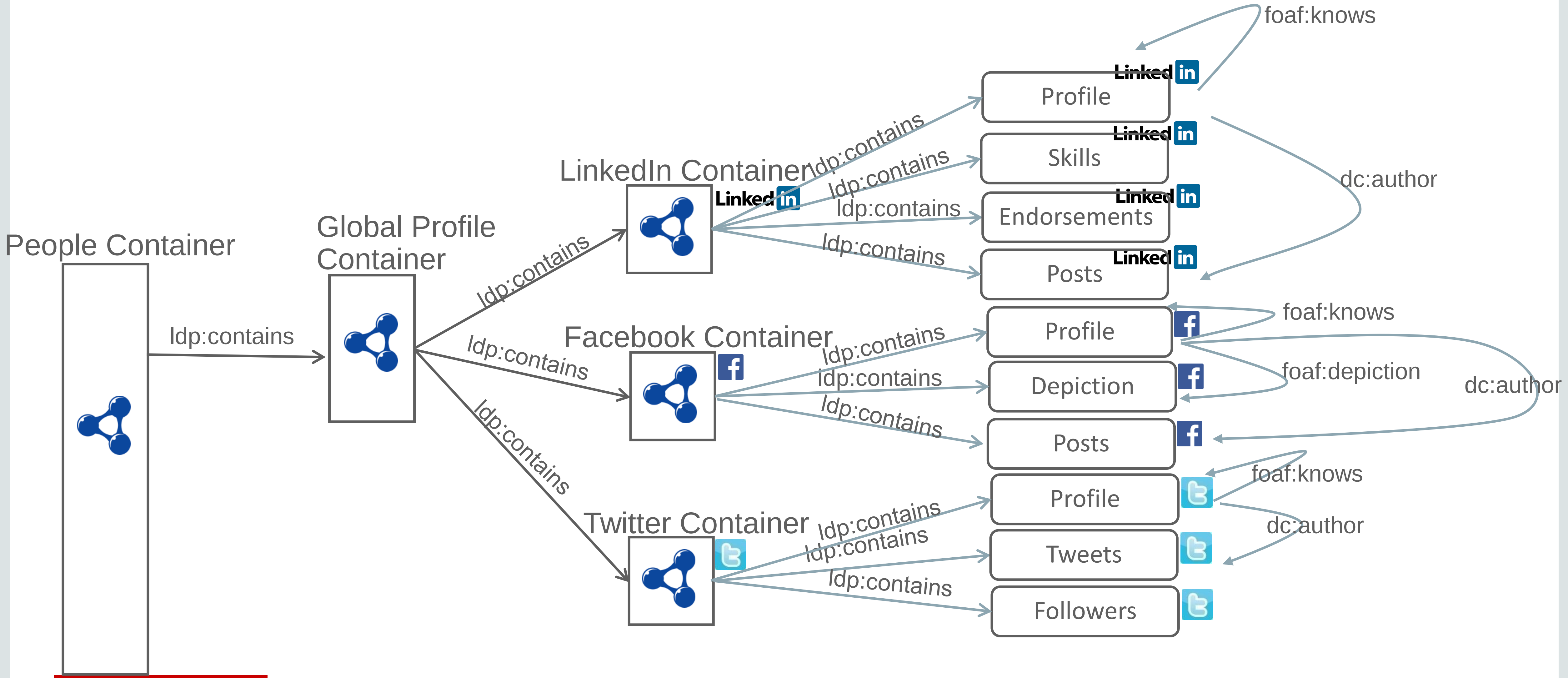
LDP usage in Oracle HCM
Updates from the HCM Cloud



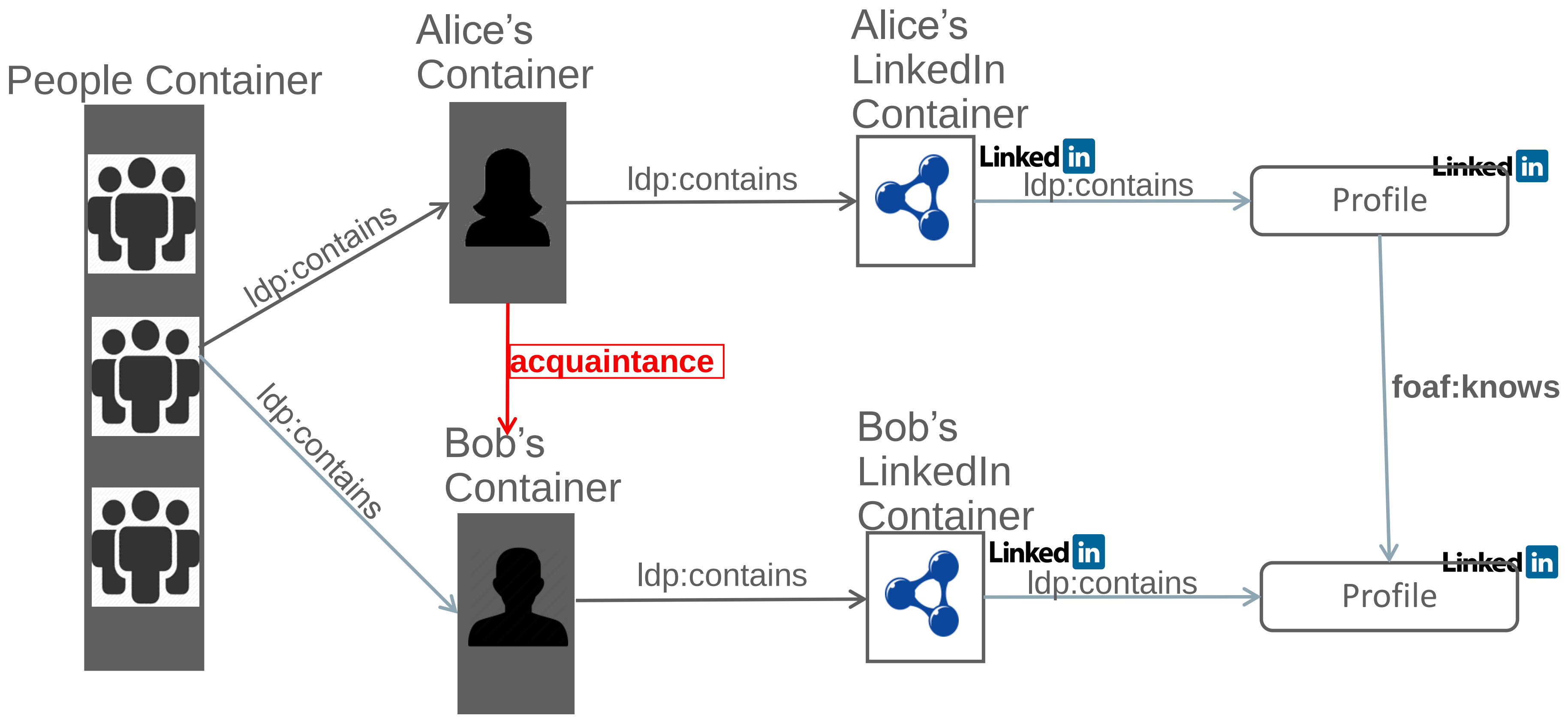
LDP Containment Hierarchy: Different Social Profiles Managed within LDP Containers



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

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.

Request

```
GET /alice HTTP/1.1
Host: oracle.com
Accept: text/turtle
Authorization: Bearer token with
attributes or roles of the agent
```

Response

```
HTTP/1.1 200 OK
Content-type: text/turtle
Link: <ldp:ProtectedContainer>
rel="type"
…
<triples>
```

ORACLE

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>.
```

ORACLE

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> .
```

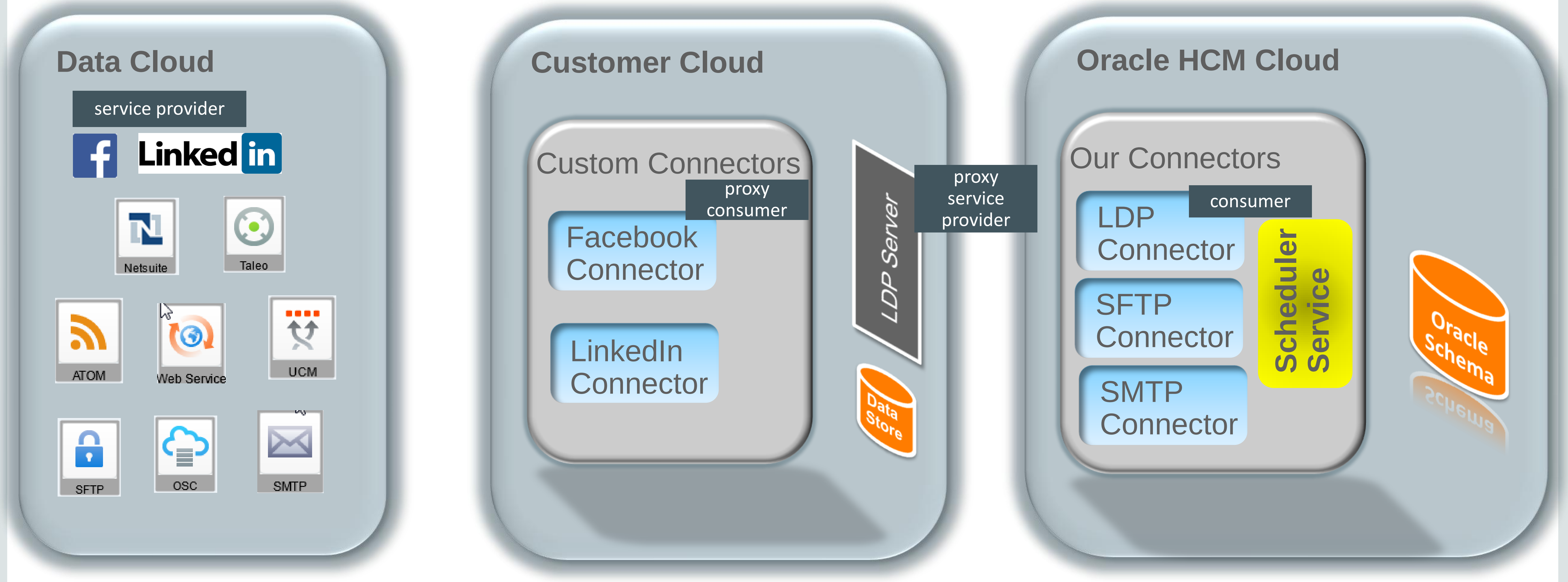
- The authorization policy itself can be an LDP container.

Policy

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

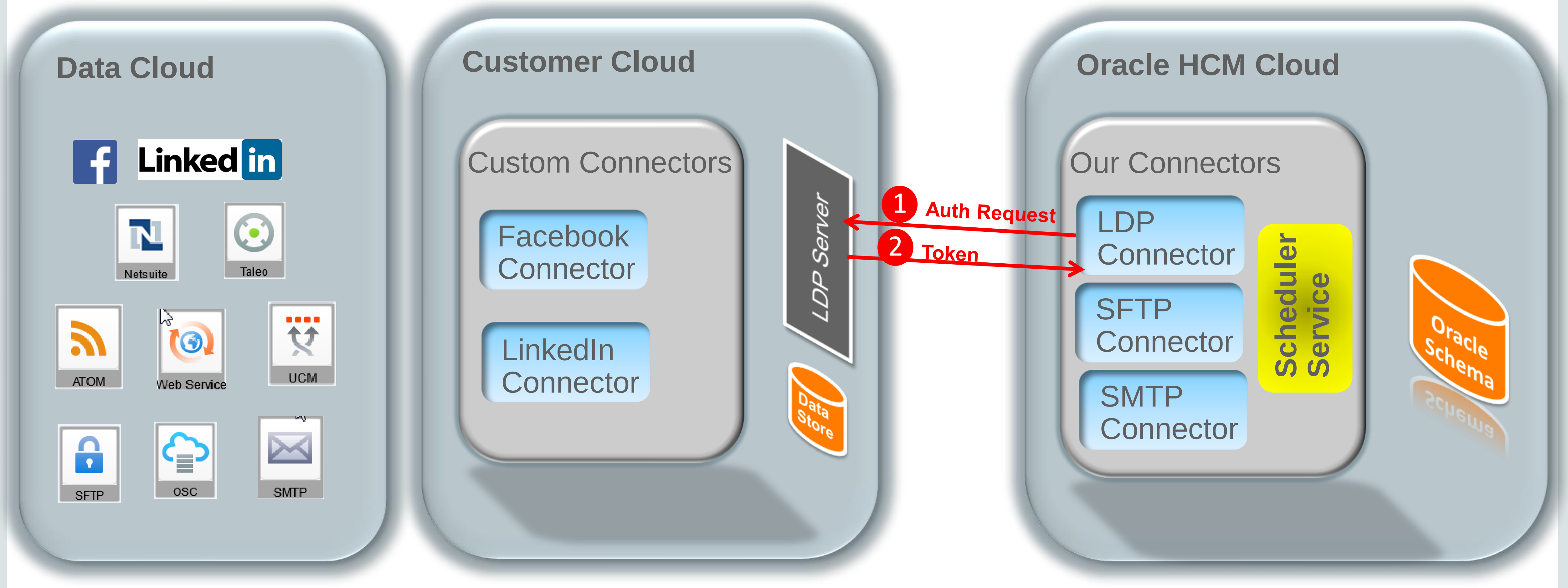
- If no policy is specified, the parent policy should take effect.

Authentication and Authorization: Terminology

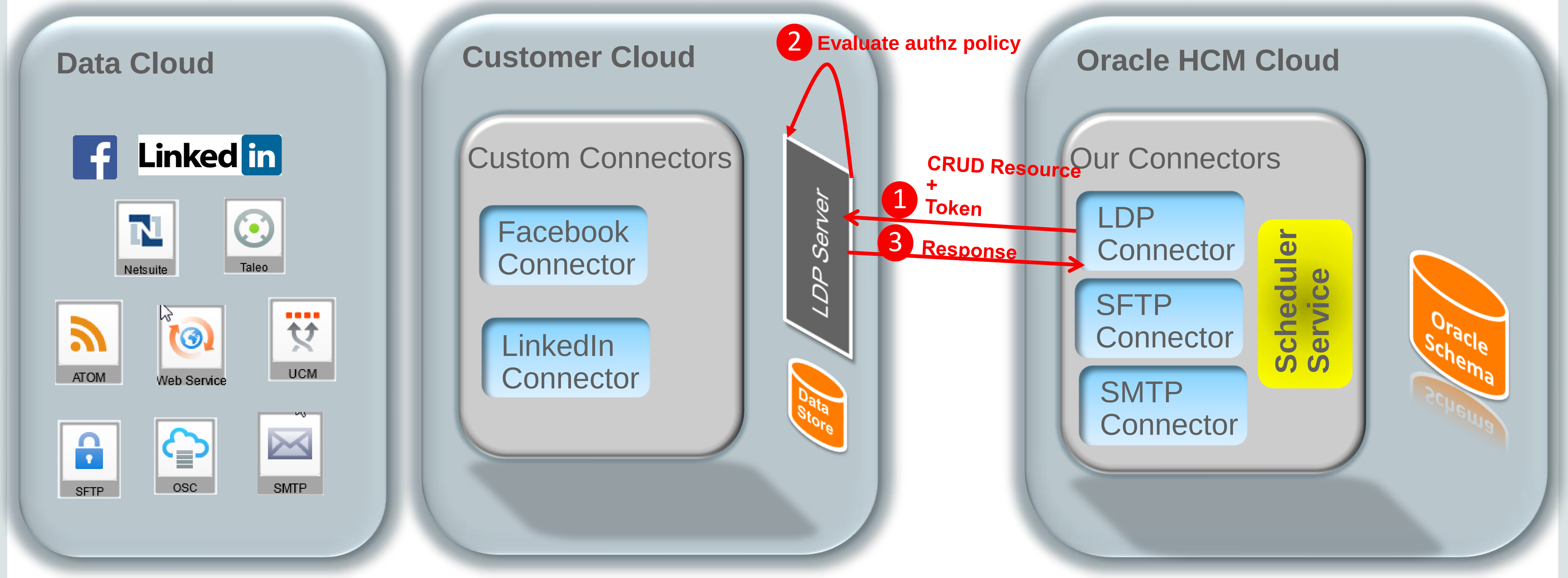


ORACLE

Authentication and Authorization: Application Init



Authentication and Authorization: Requesting Resource



ORACLE

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

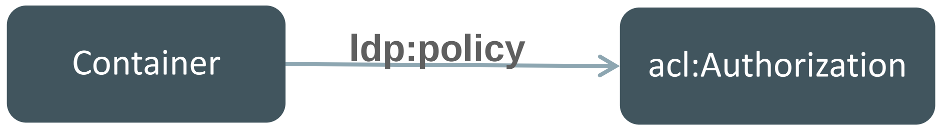
Usage Policy

```
<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

```
<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

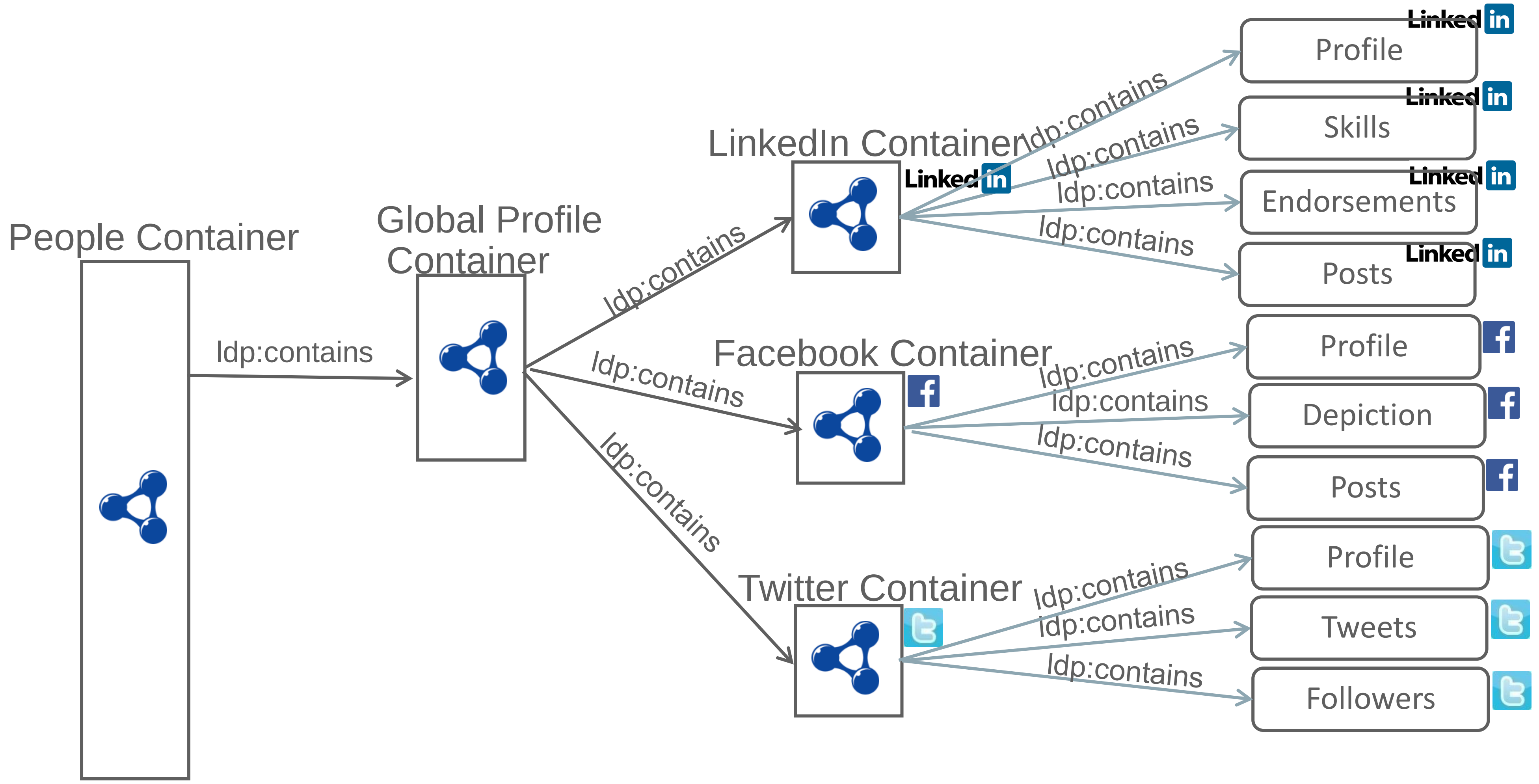
- In certain cases, we need a resource to have membership in more than one container. i.e:

```
        ParentContainer1 ldp:contains     Resource1
        ParentContainer2 ldp:contains     Resource1
```

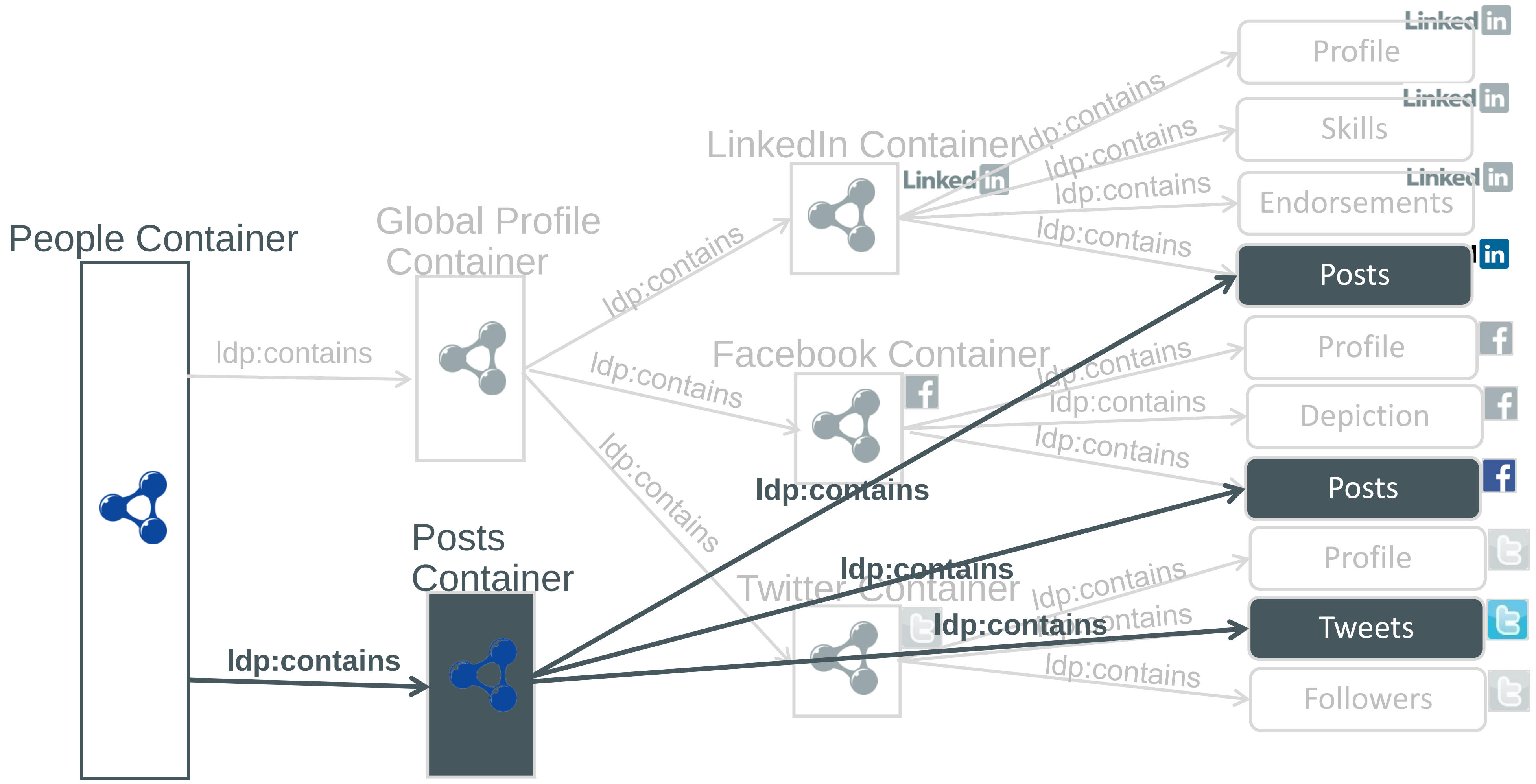
- This can aid in resource and container discovery based on a certain topic, alternate categorization, etc.
- Here is an example:

ORACLE

LDP Containment Hierarchy: Different Social Profiles Managed within LDP Containers



Different Social Profiles Managed within **Multiple** LDP Containers



Creating a New Container with Existing Resources

Request

```
POST /people HTTP/1.1
Host: oracle.com
Link: <ldp:Container>; rel="type"
Slug: posts

<> a ldp:Container;
   dcterms:title "All Posts";
   ldp:contains
<http://oracle.com/people/linkedin/posts>,
<http://oracle.com/people/facebook/posts .
```

Response

```
HTTP/1.1 201 Created
Location:
http://oracle.com/people/posts/
Link: <ldp:Container> rel="type"
Content-Length: 0
```

ORACLE

Adding Existing Resources to another Existing Container

Request

```
POST /people/posts HTTP/1.1
Host: oracle.com
Link: <ldp:Container>; rel="type"

<> ldp:contains
<http://oracle.com/people/twitter/tweets> .
```

Response

```
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.

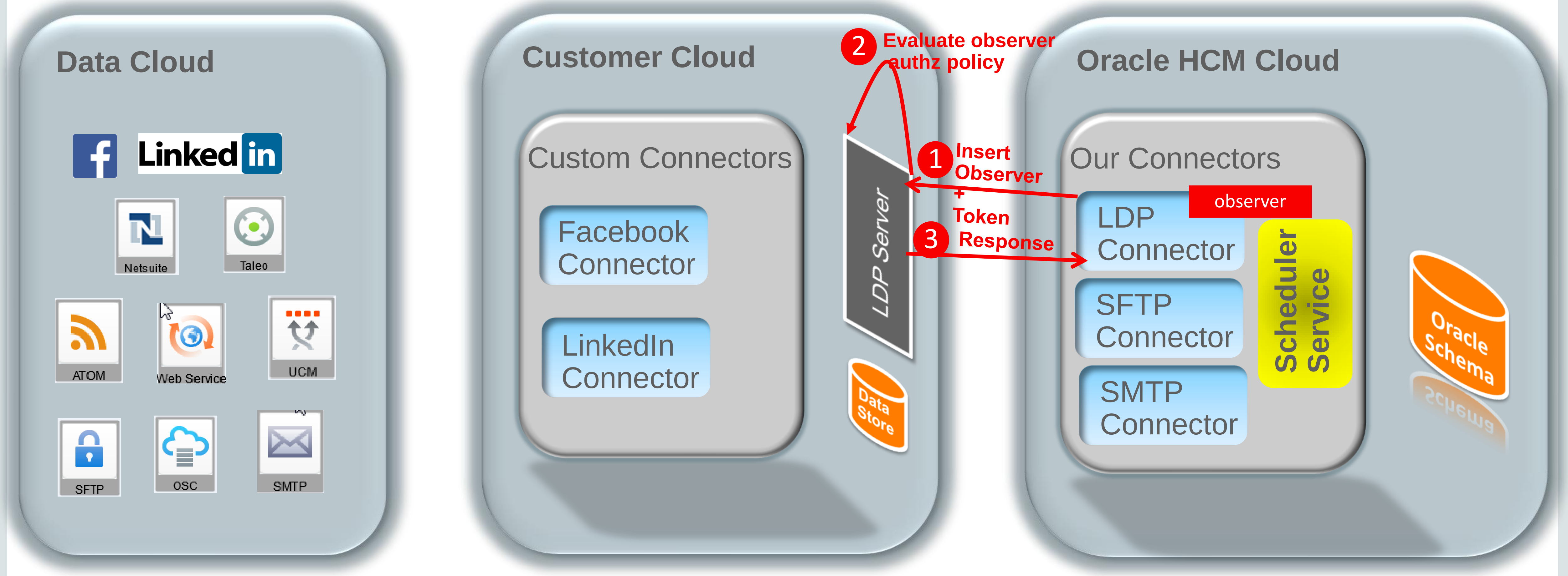
Container Definition with an Observer

```
<alice> a ldp:BasicContainer;
        ldp:observer <http://oracle.com/hcm-cloud:1234/updates> .
```

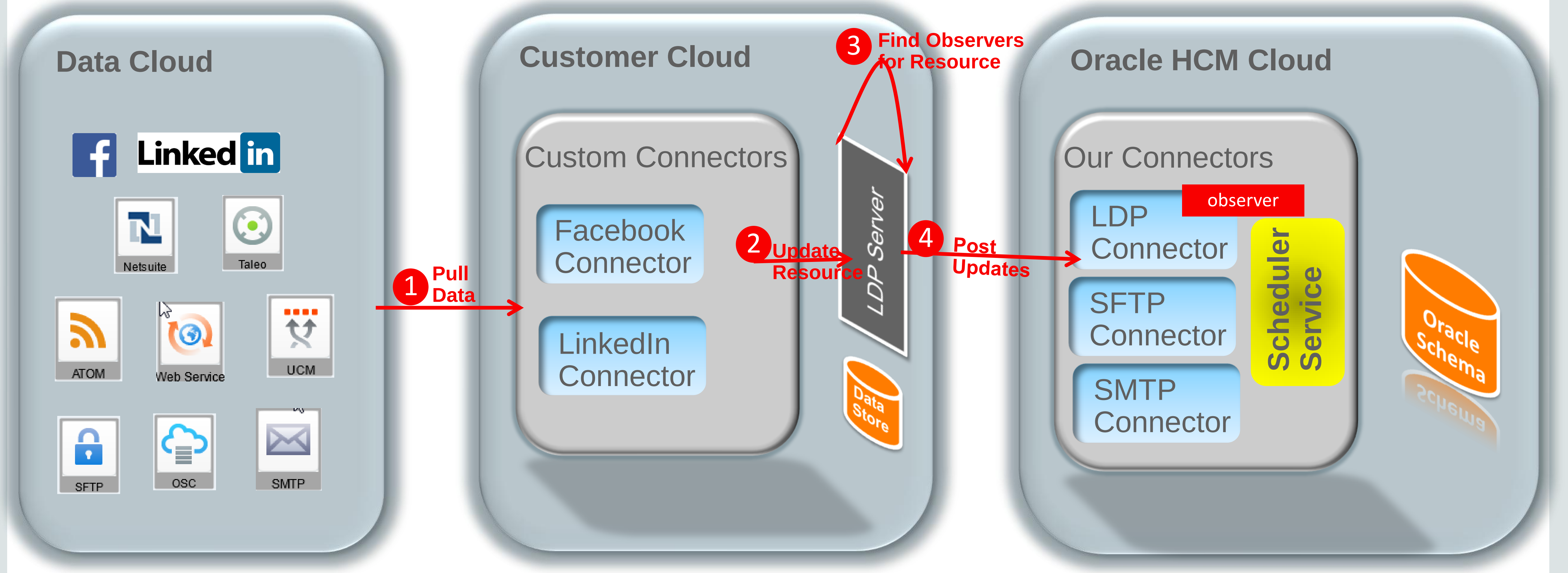
- When new triples are added, or when the existing triples are changed, the LDP Server can send POST requests to the specified observer.
- Additional Authorization Privilege **'observe'**:
  - Enable observers to write to the container or resource observed
  - Make the promise to the observers to send HTTP Post requests

ORACLE

Push Capabilities: Registering An Observer



Push Capabilities: Receiving Updates



4. Resource Filtering Semantics

- What is supported now:

Request

```
GET /alice HTTP/1.1
Host: oracle.com
Accept: text/turtle
Prefer: return=representation
include="ldp:PreferMinimalContainer"
omit="ldp:PreferMinimalContainer"
```

Response

```
HTTP/1.1 200 Ok
Content-type: text/turtle
Link: <ldp:BasicContainer> rel="type"
…
<alice> a <ldp:BasicContainer>,
…
```

- 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

ORACLE

Agenda

Questions?

oshani.seneviratne@oracle.com

ORACLE