Registry Extraction API

ⓘ As of Q3 2015, the CIFER Project has been transitioned to the TIER-Data Structures and APIs Working Group.

- Use Cases
- Models
- Notification Strawman
    - Pull
    - Push
    - Register
- Read API
- Search API
- Open Questions

The Registry Extraction API is intended to make data available from an identity registry to downstream systems.

Use Cases

1. Provisioning LDAP
2. Connecting to a provisioning system (directly or via a message queue/ESB/etc)
3. Tight binding of applications to the Registry (including an Identity Console)

Models

Currently, two models of obtaining data from the Registry are being considered. They are not mutually exclusive.

1. Read API: A fairly rich, RESTful API to obtain data from the Registry on demand.
2. Notification API: When data is changed in the Registry, a notification is made available (specific mechanics TBD). Initially, this design will be simple, conveying only that a change happened for a person, and possibly providing all of that person's data. The recipient of the message will need to determine what, if anything, is of interest. This API can operate via push or pull, although a given Registry need not support both.
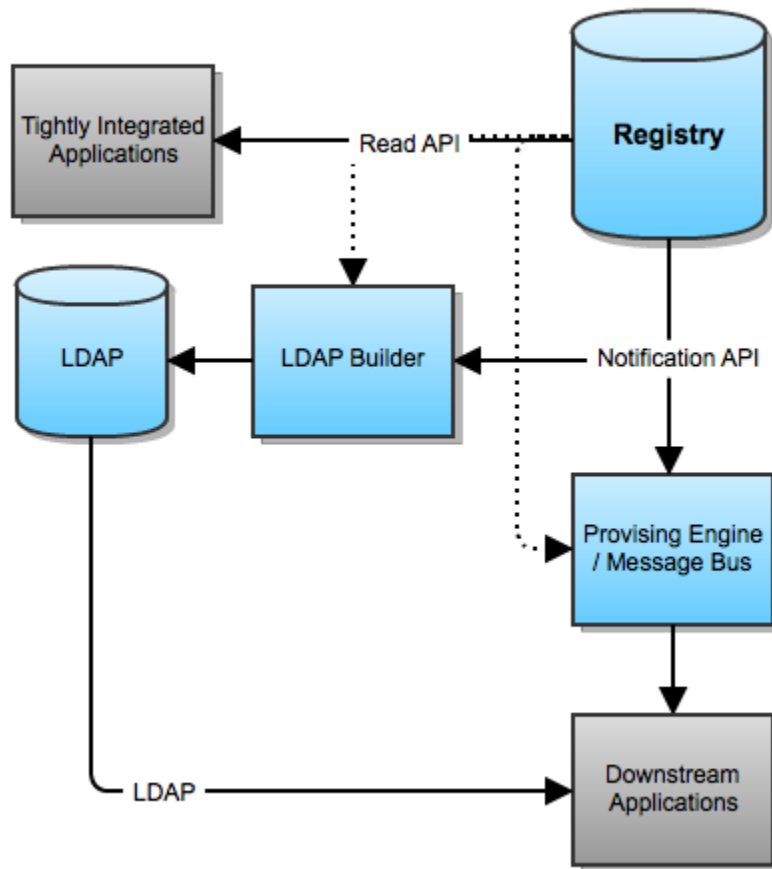
Notification Strawman

Pull

In pull operation, a client consuming notifications pulls changes from the Registry:

```
GET /v1/events?since=<serialnumber>

200 OK
{
  "events":
  [
    {
      "serialNumber":103,
      "sor":"registry",
      "entity":"/v1/people/enterprise/49873",
      "timestamp":"2012-10-04T03:10:14.123Z",
      "comment":"Updated name",
      "messageType":"diff",
      "attributes":{
        "names":[
          {
            "type":"official",
            "given":"Pat",
            "family":"Lee"
          }
        ]
      }
    },
    {
      "serialNumber":104,
      "sor":"registry",
      "entity":"/v1/people/enterprise/49873",
      "timestamp":"2012-10-04T03:10:19.100Z",
      "comment":"Updated telephonenumber",
      "messageType":"diff",
      "attributes":{
        "telephoneNumbers":[
          {
            "type":"official",
            "number":"+1 646 555 1234",
            "verified":"no"
          }
        ]
      }
    }
  ]
}
```

- *serialNumber* allows a client to pull new changes by indicating the last change it saw. As such, the value must be comparable, with later events having later serial numbers.
- *sor* designates the system of record that triggered the change.
- *entity* is a reference to a resource obtainable via the Read API.
- *timestamp* is when the event was recorded, which is not necessarily when the change happened.
- *comment* is an optional human readable comment describing the change.
- Attributes may be passed in a message. When this is done
  - *messageType* indicates whether only the changed attributes are passed ("diff"), only a subset of attributes are passed ("partial") or the full record is passed ("full").
  - *attributes* is of the same format as the Read API.
- TBD: Can a client send a filter to restrict the events it wishes to see?

An individual event can be retrieved by ID, or via a special request for the latest event.

```
GET /v1/events/<serialnumber>

200 OK
{
  "serialNumber":103,
  "sor":"registry",
  "entity":"/v1/people/enterprise/49873",
  "timestamp":"2012-10-04T03:10:14.123Z",
  "comment":"Updated name"
}
```

```
GET /v1/events/latest

200 OK
{
  "serialNumber":103,
  "sor":"registry",
  "entity":"/v1/people/enterprise/49873",
  "timestamp":"2012-10-04T03:10:14.123Z",
  "comment":"Updated name"
}
```

Push

In push operation, the Registry sends changes to an endpoint (presumably a message queue or something similar):

```
PUT /v1/events/<serialnumber>
{
  "serialNumber":103,
  "sor":"registry",
  "entity":"/v1/people/enterprise/49873",
  "timestamp":"2012-10-04T03:10:14.123",
  "comment":"Updated name"
}
```

- *sor* is an identifier to indicate the source (eg: Registry, GMS, etc) of the event.
- *entity* is a reference to a resource obtainable via the Read API.
- *timestamp* is when the event was recorded; if omitted *now* is assumed.
- *comment* is an optional human readable comment describing the change.
- Attributes may be passed in a message. When this is done
    - *messageType* indicates whether only the changed attributes are passed ("diff") or whether the full record is passed ("full").
    - *attributes* is of the same format as the Read API.

Register

The register operation can be used by other systems to trigger change notifications via the Registry.

```
POST /v1/events
{
  "sor":"gms",
  "entity":"/v1/people/enterprise/49873",
  "timestamp":"2012-10-04T03:10:14.123",
  "comment":"Updated membership"
}

200 OK
{
  "serialNumber":"108"
}
```

- *sor* is an identifier to indicate the source (eg: Registry, GMS, etc) of the event.
- *entity* is a reference to a resource obtainable via the Read API.
- *timestamp* is when the event was recorded; if omitted *now* is assumed.
- *comment* is an optional human readable comment describing the change.
- Attributes may be passed in a message. When this is done
    - *messageType* indicates whether only the changed attributes are passed ("diff") or whether the full record is passed ("full").
    - *attributes* is of the same format as the Read API.

Read API

The Read API retrieves the set of attributes associated with a Registry Person. It is not quite the inverse of the write API, since the write API operates on an SOR Person Role.

The Person is identified using an identifier type (*enterprise* in this example) and the actual identifier.

```
GET /v1/people/enterprise/E87234345

200 OK
{
  "names":[
    {
      "type":"official",
      "given":"Pat",
      "family":"Lee"
    }
  ],
  "emailAddresses":[
    {
      "address":"pat.lee@gmail.com",
      "type":"delivery"
    }
  ],
  "identifiers":[
    {
      "identifier":"pl53",
      "type":"network"
    },
    {
      "identifier":"E78967890",
      "type":"enterprise"
    }
  ],
  "sponsor":{
    "identifier":"12345",
    "type":"enterprise"
  },
  "roleEnds":"2014-08-04T12:00:00Z"
}
```

Search API

The Search API obtains a set of matching Registry People. ⚠ Need to align with [CIFER API Framework](#).

(HTML special characters not escaped here for readability.)

```
GET /v1/people

GET /v1/people?sor=hrms

GET /v1/people?name.given=j&name.family=smith

GET /v1/people?sponsor=E87234345

GET /v1/people?validThrough=lt.2014-02-14

200 OK
{
  "people":[
    "/v1/people/enterprise/E57413693",
    "/v1/people/enterprise/E78967890",
    "/v1/people/enterprise/E87234345",
  ]
}
```

Use POST for more complicated searches?

Open Questions

1. Can we come up with a better name?
2. Should attributes be conveyed with the change event?
3. How should *entity* be more formally defined?