OTTO Overview

Open Trust Taxonomy for Federation Operator

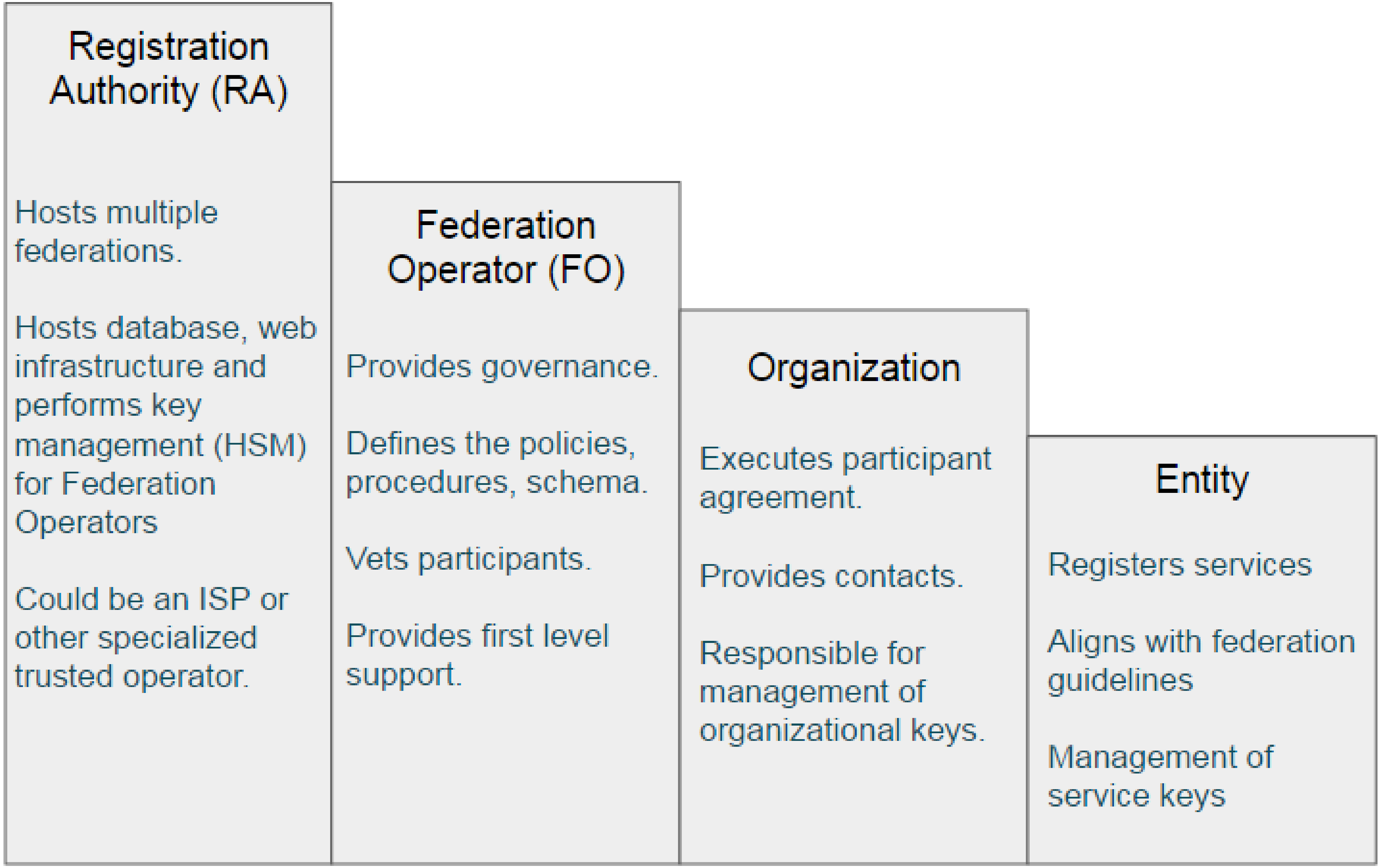
Kantara OTTO Working Group

Michael Schwartz, co-chair

Janusz Ulanowski, co-chair

Design Goals

- Facilitate publication of federation data for multiple technologies
  - SAML
  - OAuth,
  - PKI
  - Other services
- Define interoperable API’s that empower fine-grain discovery
  - Participants (request metadata from one or more participants)
  - Roles (search for specific entities, types of entities, or categories of entities)
  - Schema: user attributes, acr, amr
  - Security Policies and procedures: i.e. trust marks, software statements
- Create an architecture for inter-federation that avoids duplication of data



Registration Authority (RA)

Hosts multiple federations.

Hosts database, web infrastructure and performs key management (HSM) for Federation Operators

Could be an ISP or other specialized trusted operator.

Federation Operator (FO)

Provides governance.

Defines the policies, procedures, schema.

Vets participants.

Provides first level support.

Organization

Executes participant agreement.

Provides contacts.

Responsible for management of organizational keys.

Entity

Registers services

Aligns with federation guidelines

Management of service keys

JSON-LD data model

- Linked data model convenient for describing federation inter-relationships.
- Uses standard schema described in https://schema.org where possible; extend common schema at Kantara; provides for further extension by RA's or FO's.
- Can be converted to RDF and processed by standard tools.
- Developer friends—looks like JSON, and linked data features can be ignored by those who don't care.
- Consistent with new design being developed by W3C verifiable claims task group.
- See https://www.w3.org/TR/json-ld/#basic-concepts

Federation endpoint

POST /federation	Create federation
GET /federations	Get list of federations hosted by RA
GET /federations/{id}	Get a specific federation
GET /federations/{id}/jwks	Get federation key set
PUT /federations/{id}	Edit federation
POST /federations/{federationid}/{entityid}	Join existing entity to federation
POST /federations/{federationid}	Create entity and join federation
DELETE /federations/{id}	Remove federation
DELETE /federations/{id}/{entityid}	Remove entity from federation

Federation entity endpoint

POST /federation_entity	Create federation entity
GET /federation_entity	Get list of federation entities ids
GET /federation_entity/{id}	Get a specific federation entity
PUT /federation_entity/{id}	Edit federation entity
DELETE /federation_entity/{id}	Remove federation

Organization Endpoint

POST /organization	Create organization
GET /organization	Get list of organizations hosted by RA
GET /organization/{id}	Get a specific organization
PUT /organization/{id}	Edit organization
POST /organization/{oid}/federation/{federationid}	Add organization to federation
POST /organization/{oid}/federation_entity/{federation_entity_id}	Associate organization with entity
DELETE /organization/{id}	Remove organization

Discovery Endpoints

GET /otto/.well-known/otto-configuration	Federation endpoints, supported algs
DELETE /jwks	JSON key set for RA

Parameter: depth

- Specify which objects you want to return

`GET` /federations/1234?*depth=federations.organization*

Only return organization entities

Paramater: filter

- Using JSPath query syntax: https://github.com/dfilatov/jspath

`GET` /federations?*filter=.entities{.name=“MyWebsite”}*

Only return entity with this name…

Parameter: sign

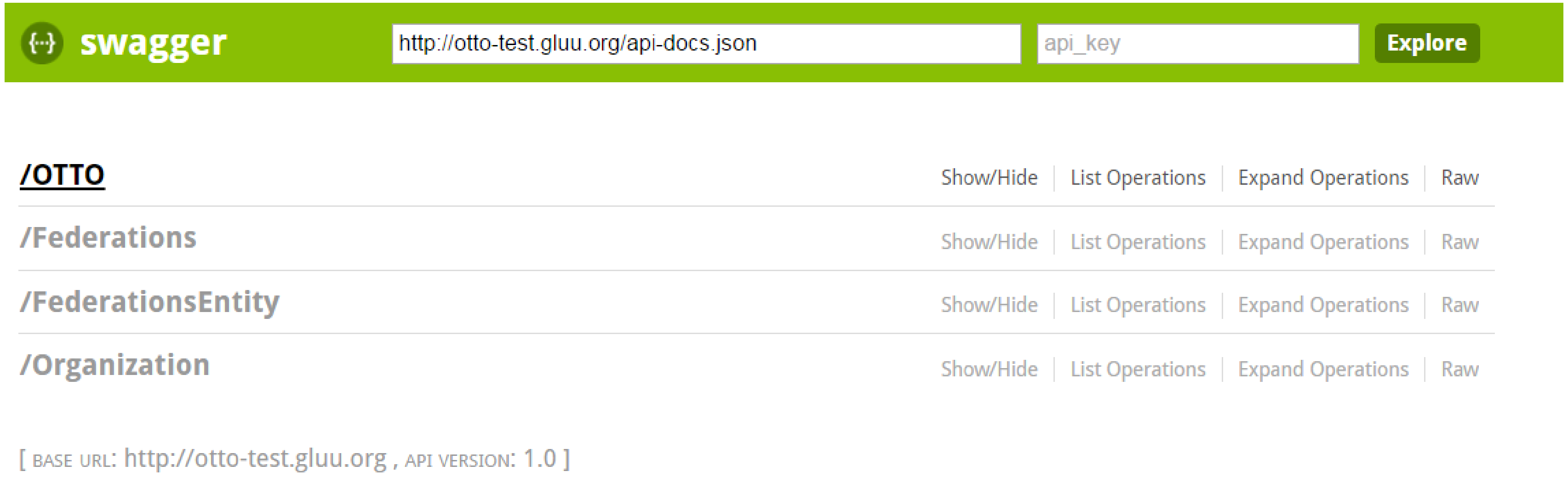
- Sign either a complete or partial result set

`GET` /federations/1234/*sign=true&alg=RS512*

Return signed JWT …

Test Implementation

- Server was written to demonstrate feasibility
  - MongoDB was used as the backend—loose schema
  - Performance was tested with 10,000 entries
    - Query and filter features seem to scale
  - MIT license



http://otto-test.gluu.org/api-docs.json

/OTTO

/Federations

/FederationsEntity

/Organization

[BASE URL: http://otto-test.gluu.org , API VERSION: 1.0]

Next steps

- Need to finalize schema – both keys and values
  - Organizations
  - SAML, OpenID, UMA Entities
  - Software statements
  - Trustmarks
  - User schema (eduperson)
  - ACR / AMRs
- Need to convert technical spec to English with good examples
  - Need writers!
- Need to pilot

What about OpenID Connect federation draft?

- Complimentary
  - OTTO would facilitate the creation of federations such as the kind imagined by the OIDC federation spec.
  - OIDC spec doesn’t say where|how the federation publishes its keys
  - It also doesn’t address issues beyond OpenID Connect
  - It has no automated discovery capabilities – for example how would the federation publish information about supported schema or trustmarks?

Links

- Join the WG: http://www.gluu.co/join-otto
- Github Project: https://github.com/KantaraInitiative/wg-otto
- Test code: https://github.com/GluuFederation/otto-node
- Swagger UI for test code: http://otto-test.gluu.org/swagger/
- OTTO test data generator: http://otto-test.gluu.org:8080/