DRS Auth Application

Assumptions:
Professional Users need access to applications based on the Organisation(s) / groups they belong to

Some Users in Organisations will have different access needs (higher or lower) than other Users in the same Organisation for a given Application.

Some Users in Organisations will not need to access a particular Application even if most of the Users in that Organisation do have access.

All Users should have the ability to log into an Application managed by DRS Auth
Users want a “portal” with links to all Applications that they have access to
Users want a central place to administer who can do what where (aka Authorization)

Structure:
User records hold both personal info + login info (assume all Users will need a login).

Group Users into Organisations.
All Users must belong to 1 or more Organisation(s).
Each User must fulfil one or more roles in an Organisation.
Roles for an Organisation must be named using language consistent with the domain that Organisation operates in. (different Organisation types have different roles)
Adding a User to an Organisation means that User gets a default role(s) which can be modified

Each Organisation has a type. This determines which roles are available to Users in that Organisation

Some roles should be available to all Organisations
e.g. admin - means Users with an admin role for an Organisation can manage other Users in that Organisation

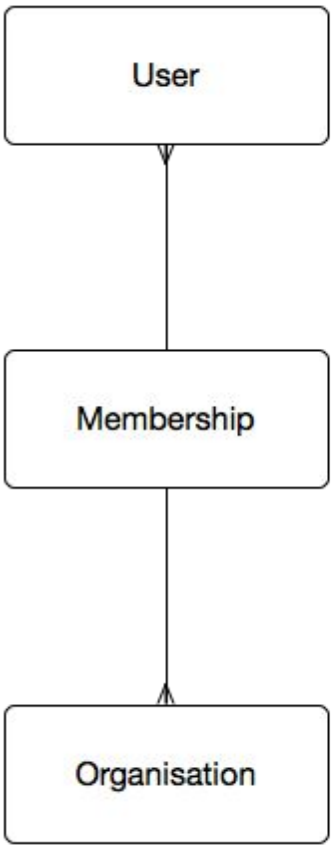
Access to Applications is determined by the type of Organisation(s) a User belongs to

Consuming Applications decide what each role means to them (if anything at all)
Consuming Applications by default deny access

Consuming Applications are free to hold additional role / permission information locally

Schema

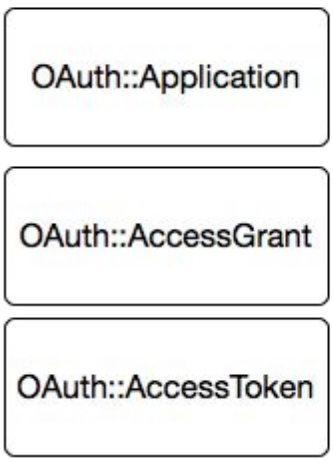
Permissions are stored in a Membership record. This links a User to an Organisation and records what roles that User performs in that Organisation.



User
- id (Integer)
- uid (uuid)
- email (string)
- password (string)
- name (string)
- address (string)
- postcode (string)
- tel (string)
-mobile (string)

Membership
- user_id (Integer)
- organisation_id (Integer)
- roles (Array)
- applications (Array)

Organisation
- id (integer)
- uid (uuid)
- type (string)
- name (string)
- address (string)
- postcode (string)
- tel (string)
- mobile (string)
- parent_organisation_uid (uuid)



OAuth::Application
- name (string)
- app_id (integer)
- app_secret (string)
- redirect_uri (string)

Configuration

Can be done using a static yaml file initially. Later on can be made more dynamic.

Organisation Types

Define what Applications and Roles a particular Organisation type has

Available roles are the roles that users in an Organisation
of that type can have.

Default roles are what users by default get when they are
added to an Organisation of that type

Example organisation_types.yaml for DRS:

```
default: &default
  - available_roles: [“admin”]
  - default_roles: []
  - applications: [“drs-auth”]

webops:
  - available_roles: [“support”]
  - default_roles: [“support”]
  - applications: [“*”]

custody_suite:
  <<: *default
  - available_roles: [“cso”]
  - default_roles: [“cso”]
  - applications: [“drs-service”]

drs_call_center:
  <<: *default
  - available_roles: [“manager”, “operator”]
  - default_roles: [“operator”]
  - applications: [“drs-service”, “drs-rota”]

law_firm:
  <<: *default
  - available_roles: [“solicitor”, “solicitor_admin”, “calendar_viewer”]
  - default_roles: [“solicitor”]
  - applications: [“drs-service”, “drs-rota”]
```

Use Cases

User created in Law Firm type Org
Automatically obtains “solicitor” role
Automatically obtains access to drs-service and drs-rota

User created in Webops type Org
Automatically obtains “support” role
Automatically obtains access to all applications (*)

User created in DRS call centre Org
Automatically obtains “operator” role
Also given “manager” role
Automatically obtains access to drs-service and drs-rota

User created in Law Firm type Org
Automatically obtains “solicitor” role
Automatically obtains access to drs-service and drs-rota
Has “solicitor” role remove
Has “calendar_view” given
Has “drs-service” access removed