Flat no more! Hierarchical multitenancy and projects acting as domains in OpenStack

Andrey Brito, Henrique Truta and Raildo Mascena
Universidade Federal de Campina Grande

Presenters

Andrey Brito

Professor - Universidade Federal de Campina Grande (Brazil)

Henrique Truta

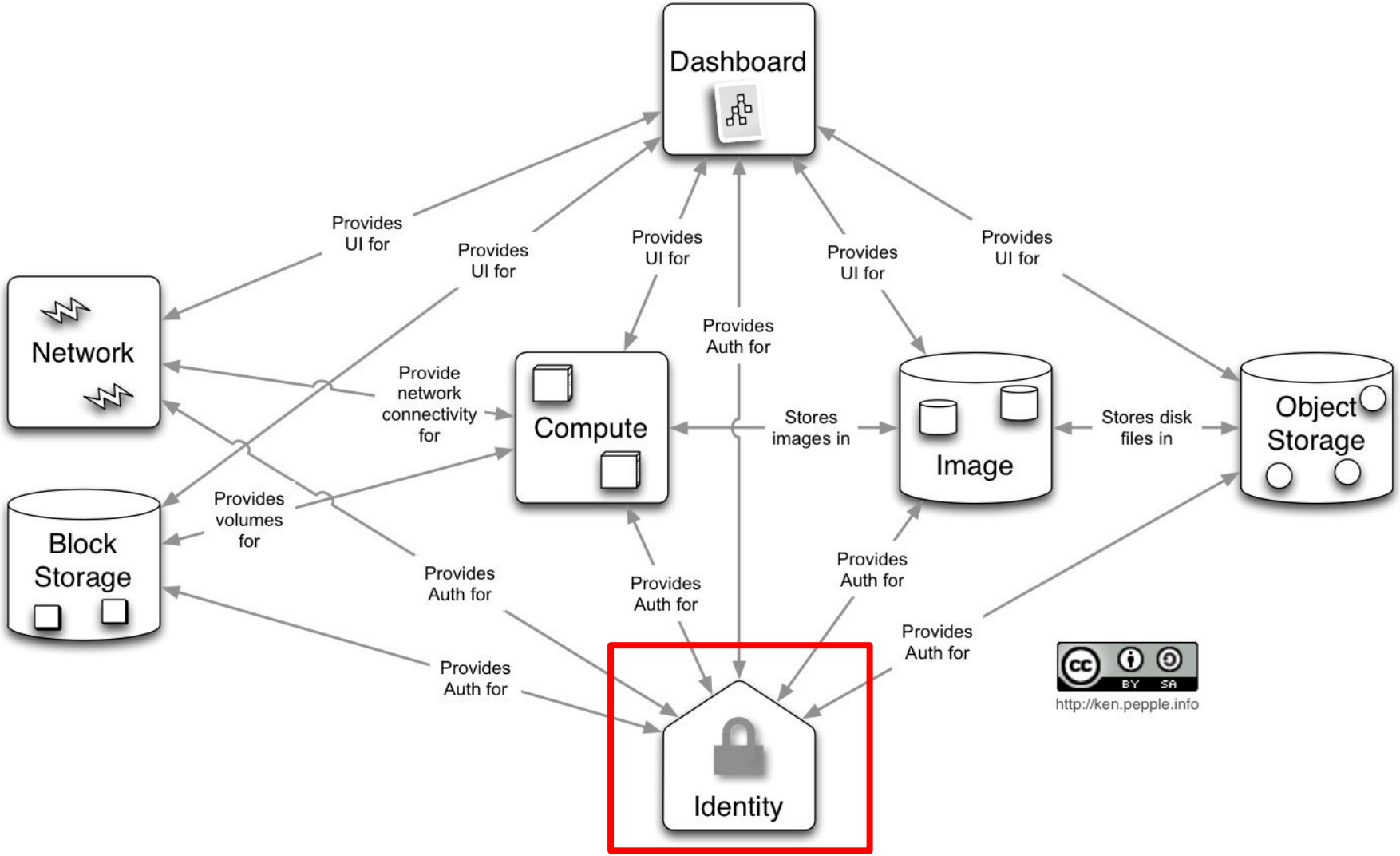
Lead Software Engineer - Universidade Federal de Campina Grande (Brazil)
OpenStack ATC

Raildo Mascena

Software Engineer - Universidade Federal de Campina Grande (Brazil)
OpenStack ATC

Agenda

- Introduction of OpenStack
- Introduction of Keystone
- Hierarchical Multitenancy
- Nested Quotas
- Projects acting as domains
- Next steps


Dashboard
Provides UI for
Provides UI for
Provides UI for
Provides UI for
Provides UI for
Network
Provide network connectivity for
Compute
Provides Auth for
Stores images in
Image
Stores disk files in
Object Storage
Block Storage
Provides volumes for
Provides Auth for
Provides Auth for
Provides Auth for
Provides Auth for
Provides Auth for
Identity
http://ken.pepple.info

Keystone

- The OpenStack component responsible for Identity management
  - Authorization
  - Authentication
  - Audit
- Supports multiple Identity providers
  - Federation
- Support for auth backends and frameworks such as LDAP and OAuth
- Enables Multitenancy

Multitenancy:

“A single instance of software that runs on a server and serves multiple tenants. A tenant is a group of users who share a common access with specific privileges to the software instance”

A bit of history

Multitenancy in OpenStack

- From Austin to Cactus:
  - One user → One tenant
  - A user could not belong to more than one tenant
  - Nova handled the authentication
- From Diablo to Folsom:
  - Keystone released in Diablo with API 2.0
  - Kept the “one user → one tenant” model
  - Simple RBAC existed: Hardcoded to admin and member operations

Multitenancy in OpenStack

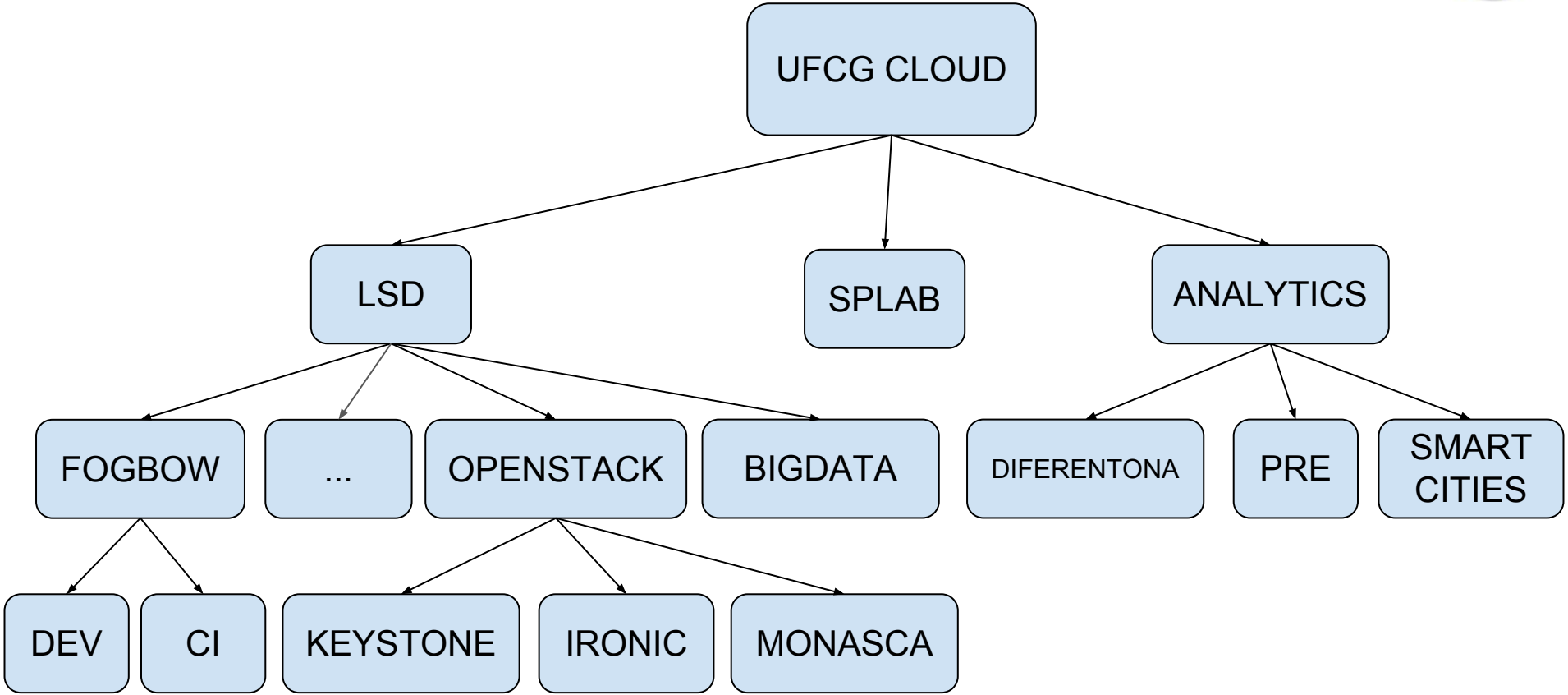
- Grizzly and Havana:
  - v3 release
    - Domains introduced: Container of projects
    - Tenants became projects
    - Users no longer belong to the tenant, but to the domain
  - One user → Many projects
  - RBAC via policy file introduced
  - “admin” role is global

Multitenancy in OpenStack

- Icehouse and Juno:
  - First efforts to eliminate global admin
  - Improvements on domain usage
    - Domain specific backends
    - Possibility of domain policy enforcement
- And Kilo...

Hierarchical Multitenancy

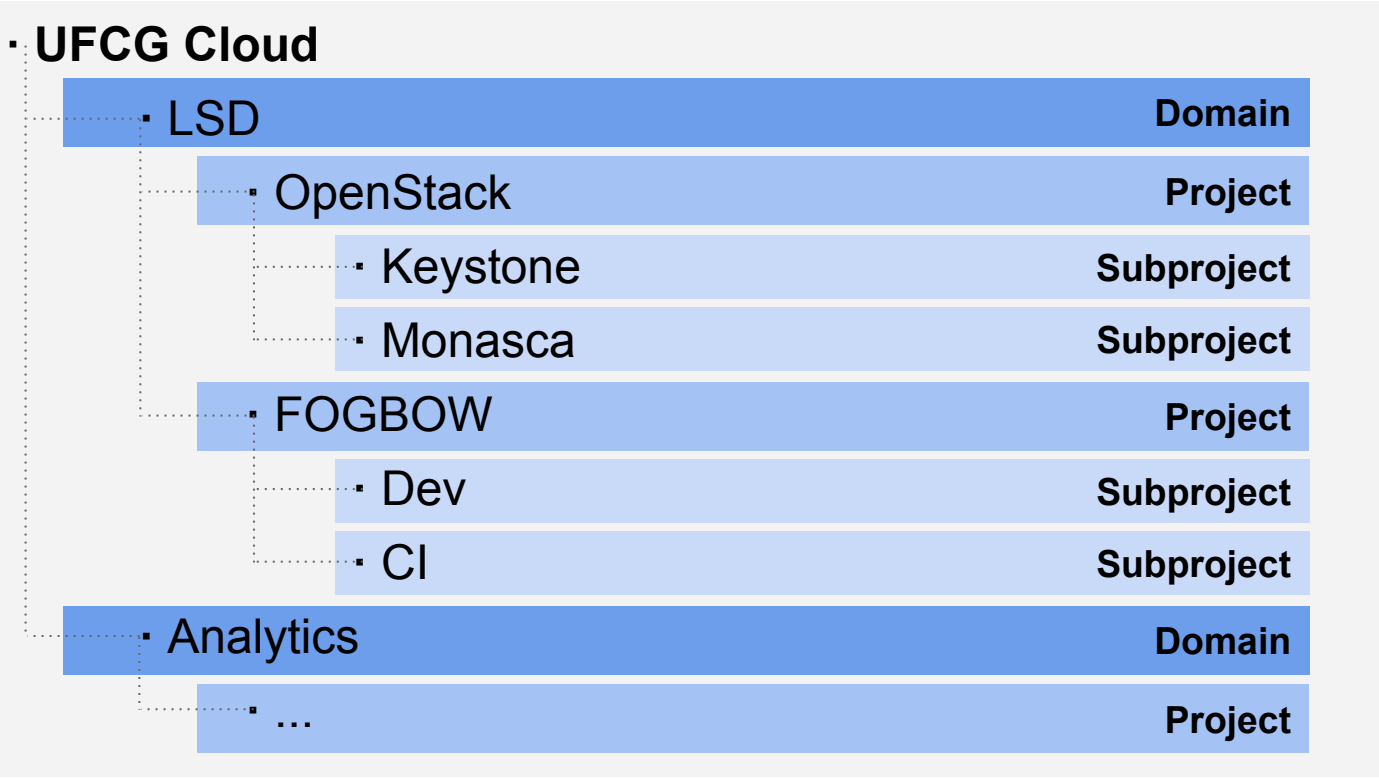
How to better represent that?



Workaround in a flat way

- **UFCG Cloud**
  - LSD — **Domain**
    - LSD_Fogbow — **Project**
    - LSD_BigData — **Project**
    - LSD_OpenStack_Keystone — **Project**
    - LSD_OpenStack_Ironic — **Project**
    - LSD_OpenStack_Monasca — **Project**
  - Analytics — **Domain**
    - Analytics_Research — **Project**

Hierarchical Multitenancy



Hierarchical Multitenancy (HMT)

Basic operations

Operation	OpenStack Call
Create	project create <p-name> [**--parent <parent_name>**]
Read	project show <p-name> [**--parents**][**--subtree**]
Update	project set <p-name> --description <p-description>
Delete	project delete <p-name>

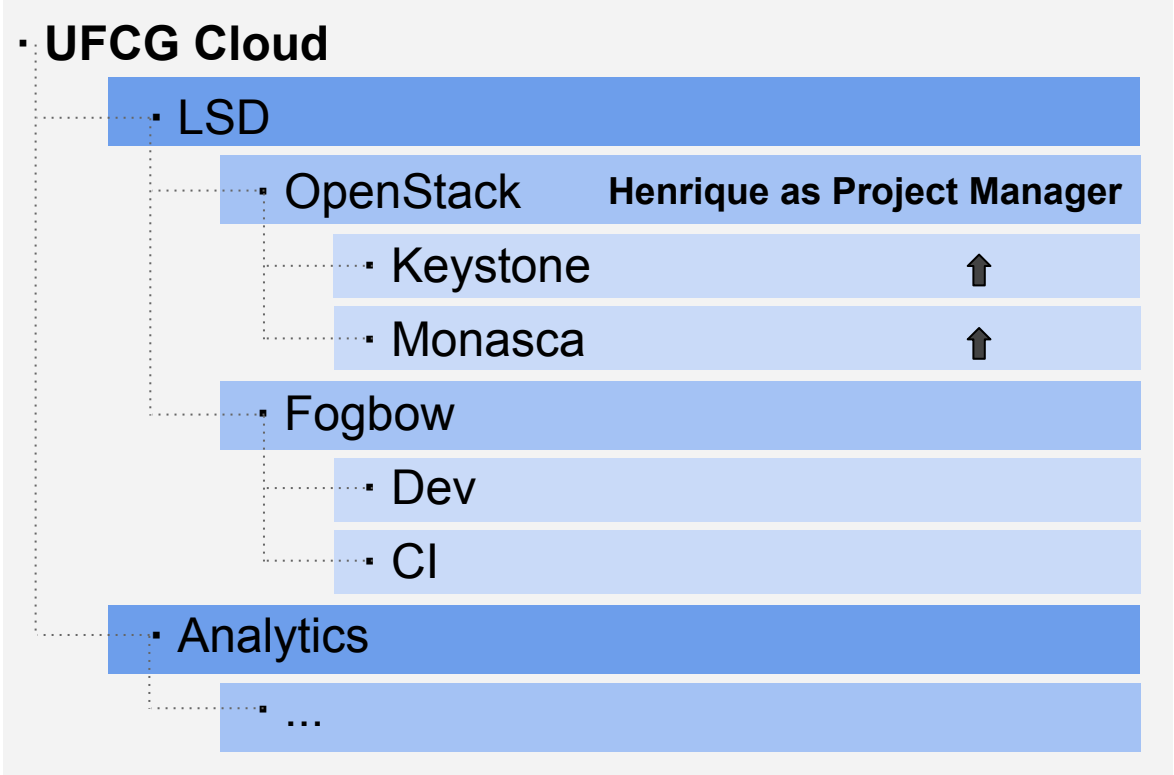
How can we improve the access control?

Usual role assignments

- **UFCG Cloud**
  - LSD
    - OpenStack — Henrique as Project Manager
      - Keystone — Henrique as PM
      - Monasca — Henrique as PM
    - Fogbow
      - Dev
      - CI
  - Analytics
    - ...

How to grant a role to a user on a project subtree?

Inherited role assignments



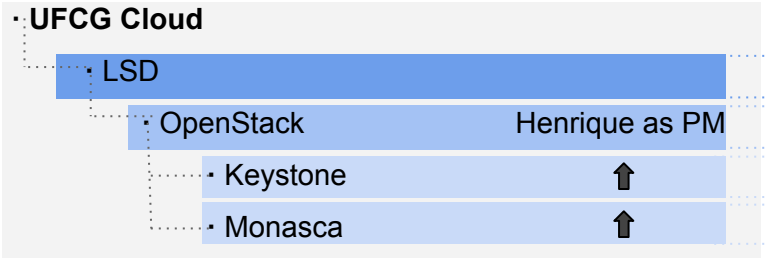
Keystone Inherited Roles Assignment Concept

Role assignment operations

Operation	OpenStack Call
Create	role add <r-name> --user <u-name> --project <p-name> [**--inherited**]
Read	role assignment list --user <u-name> [**--inherited** | **--effective**]
Delete	role remove <r-name> --user <u-name> --project <p-name> [**--inherited**]

Set Up

· Projects Hierarchy




```
domain create lsd
project create openstack  --domain lsd
project create keystone   --domain lsd  --parent openstack
project create monasca    --domain  lsd  --parent openstack
```

· User & Grant

- `user create` **`henrique`**`--domain` **`lsd`** `--password` **`tough_password`**
- `role add` **`project_manager`** `--user` **`henrique`** `--project` **`openstack`** **`--inherited`**

Enforcing quota

The current quota implementation

- The existing driver is useful to enforce quotas when projects are independent, but...
  - A quota for a subproject can exceed its parent's quota
  - The project manager cannot control the subprojects' quotas
- Others services do not support domains
  - Consequently, there are no quotas for domains
  - If you want project admins to handle their own users (i.e., give them domains), then you cannot control their quotas

Current Quota Driver

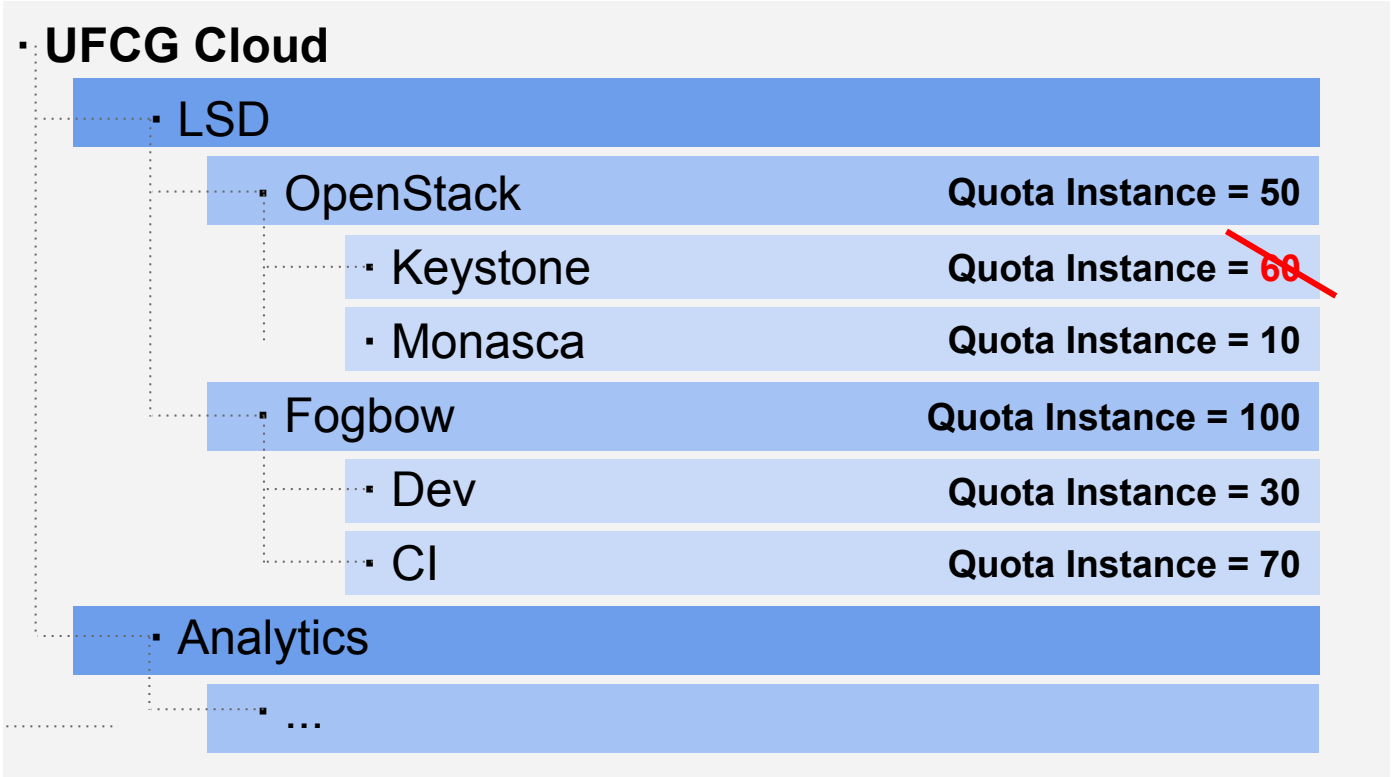
- **UFCG Cloud**
  - LSD
    - OpenStack — **Quota Instance = 50**
      - Keystone — **Quota Instance = 60**
      - Monasca — **Quota Instance = 10**
    - Fogbow — **Quota Instance = 100**
      - Dev — **Quota Instance = 30**
      - CI — **Quota Instance = 70**
  - Analytics
    - ...

Nested Quota

Nested Quota

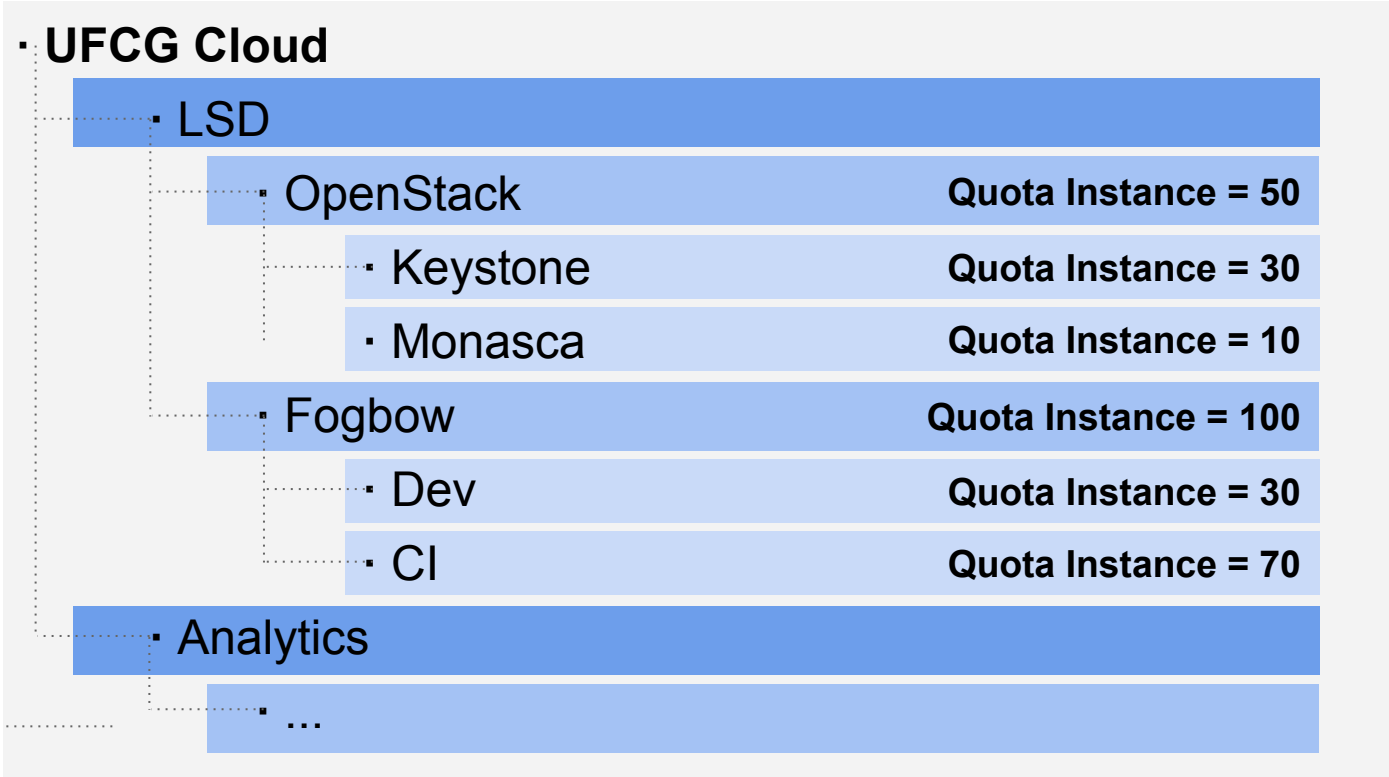
- New driver that enforces quotas in nested projects
- Allocate part of the parents' quota to their subtree
  - The project manager shares his quota: split his resources among his subprojects
  - Quota for a subproject will always be lower than its parent project

Current Quota Driver



Nested Quota

Nested Quota



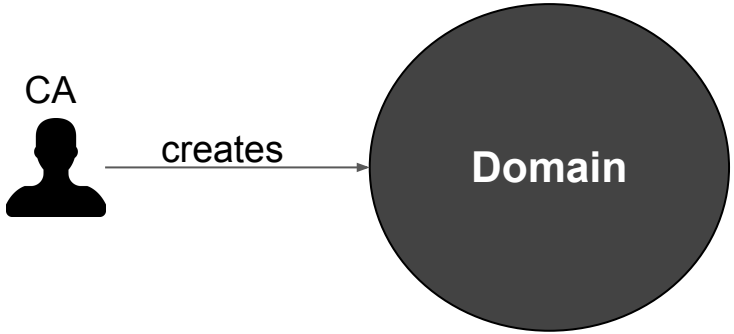
Nested Quota

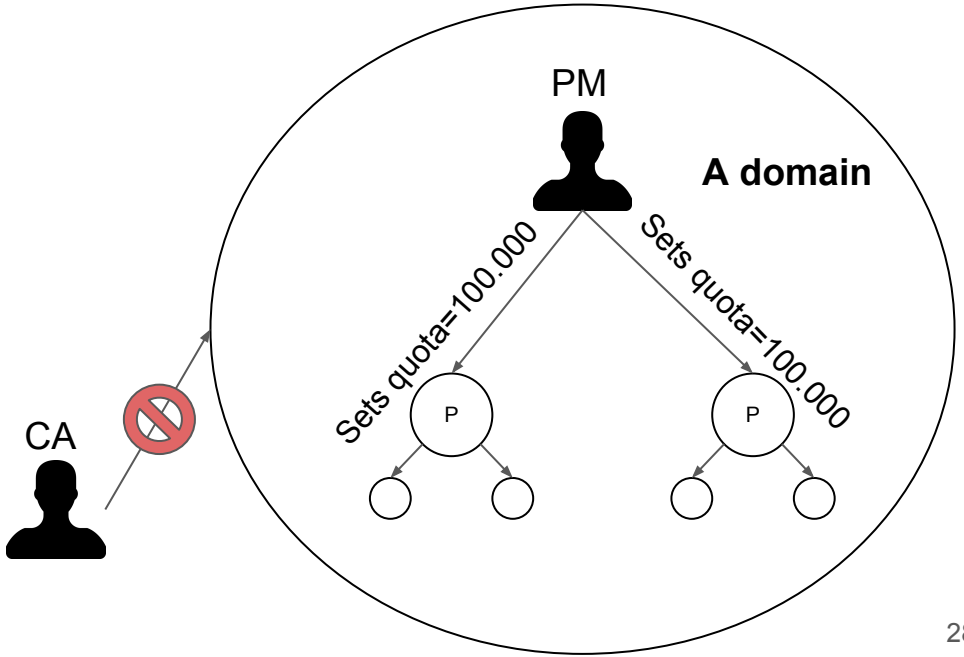
What is the expected visibility of the cloud admin?

Cloud admin delegating control

Cloud admin creates domain and gives to the PM

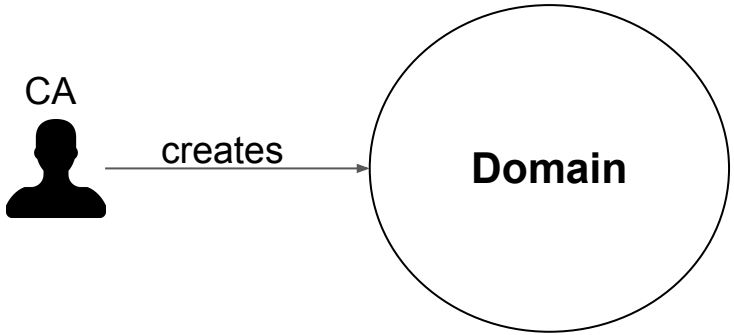
PM creates users, hierarchies and is able to set unlimited quotas



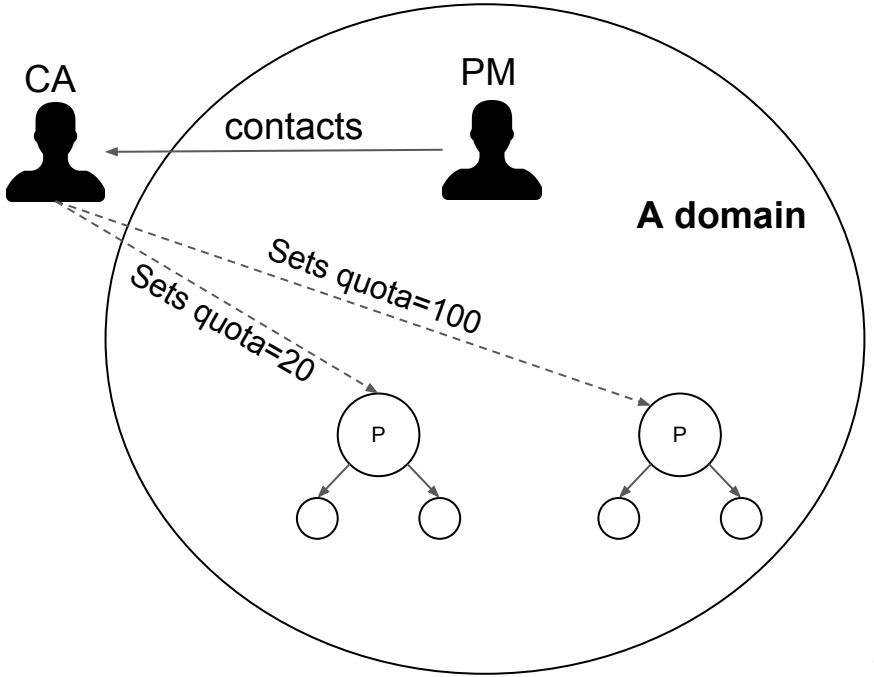


Cloud admin controlling

Cloud admin creates domain, but it will not be a black box anymore



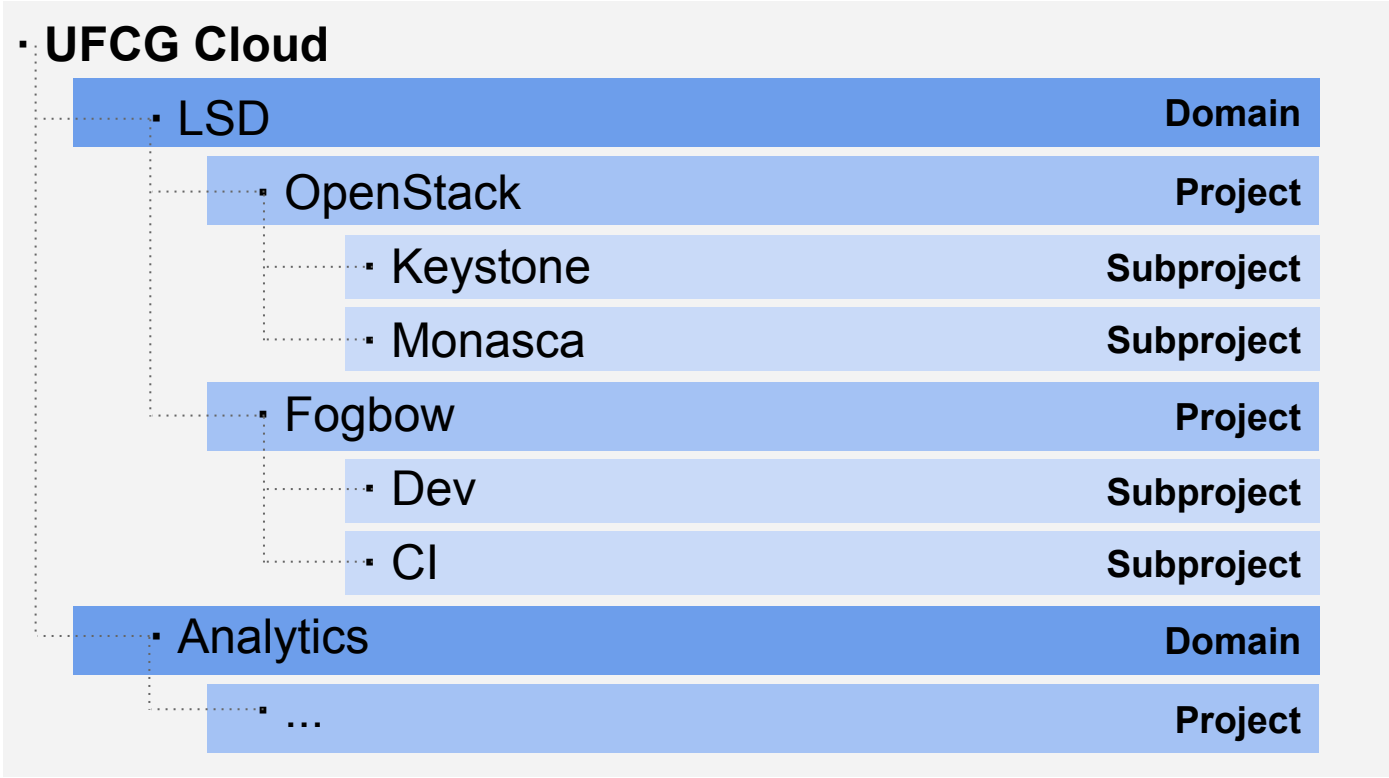
PM contacts CA when needs an operation like project creation and quota sets



Challenge:
Give project managers the control of their resources, without giving them all resources of the cloud

Projects acting as domains

Representing with HMT



HMT

New Representation with PAAD

- **UFCG Cloud**
  - LSD — **Project that acts as Domain**
    - OpenStack — **Project**
      - Keystone — **Subproject**
      - Monasca — **Subproject**
    - Fogbow — **Project**
      - Dev — **Subproject**
      - CI — **Subproject**
  - Analytics — **Project that acts as a Domain**
    - … — **Project**

Projects Acting as Domains (PAAD)

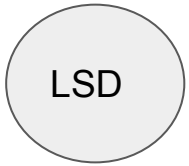
What has changed

- “LSD” is now a project (but also a domain)
  - Internally represented by the flag “is_domain”
  - It is still the container of users and projects
- But now the cloud admin is able to set its quota
- And the project managers can distribute their quota across the tree, as they creates subprojects

Creating a new hierarchy

Step 1: Cloud admin creates the project that acts as a domain

is_domain=True
parent_id=None
domain_id=None

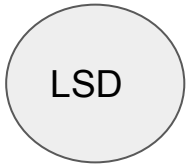


POST /v3/projects

Body:
{
“name”: “LSD”,
“description”: “My root project that acts as a domain”,
“is_domain”: true
}

Alternative step 1: Using the domain API

is_domain=True
parent_id=None
domain_id=None



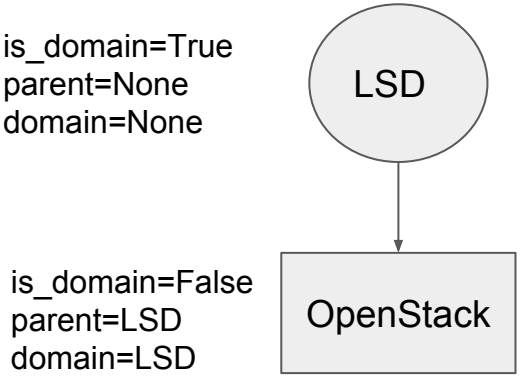
POST /v3/domains

Body:
{
“name”: “LSD”,
“description”: “My root project that acts as a domain”
}

In the CLI:

```
domain create lsd --description “My root project that acts as a domain”
```

Step 2: User creates a regular project



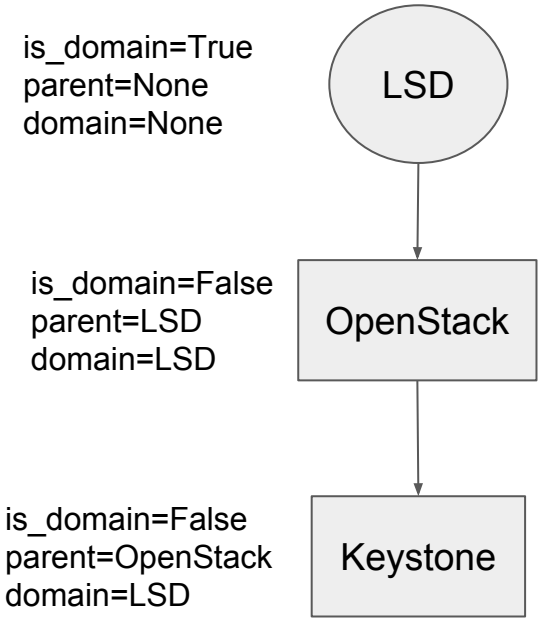
POST /v3/projects

Body:
{
“name”: “OpenStack”,
“description”: “Project of OpenStack group”,
“is_domain”: false,
“parent_id”: <lsd id>
}

In the CLI:

```
project create openstack --parent
lsd --domain lsd
```

Step 3: User creates subprojects



POST /v3/projects

Body:
{
“name”: “National marketing”,
“description”: “Project of keystone team”,
“is_domain”: false,
“parent_id”: <openstack id>
}

In the CLI:

```
project create keystone --parent
openstack --domain lsd
```

Summary & Next steps

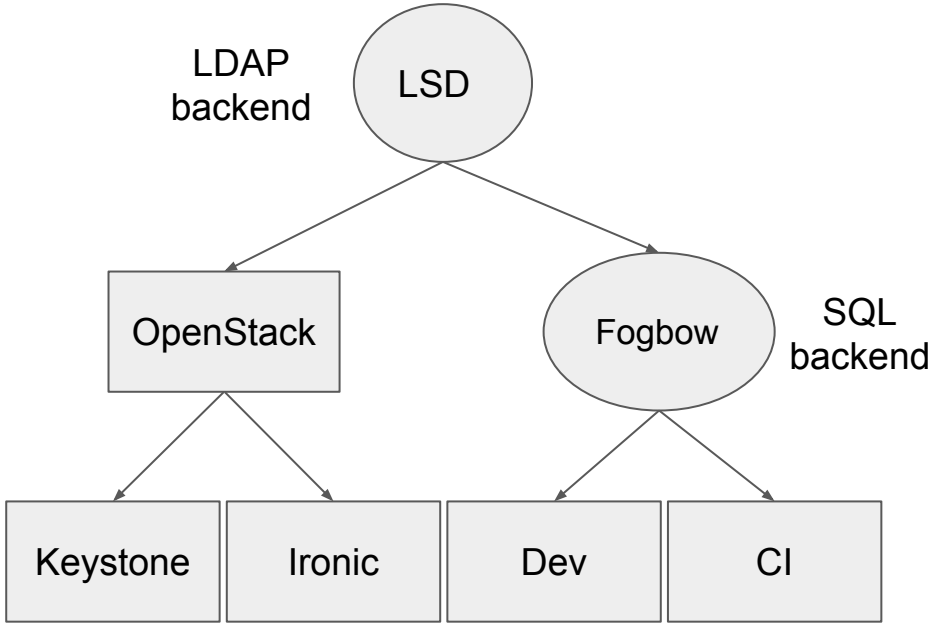
Summary

- Hierarchical Multitenancy enables better management of resource
- It is a combination of several features
  - Creating hierarchical relationships between projects
  - Assigning roles to users in projects using these hierarchical
  - Managing resource limits hierarchically
  - Delegating control of a subtree

Next steps: Reseller

- Subprojects may want to manage their users from the parent
- Delegating user management across the tree
  - Subprojects manage their own users, acting also as domains
  - Resource usage controlled by the parent PM
  - Different user backends (LDAP, SQL, AD)
- Enables reselling part of the resources

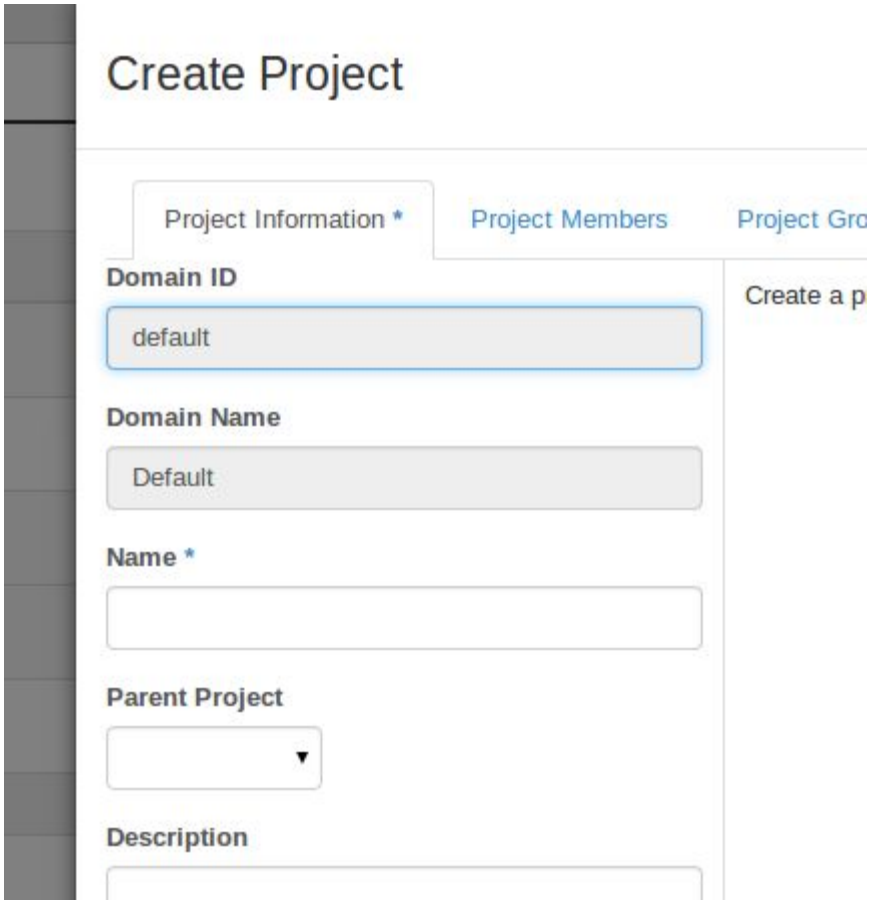
Next steps: Reseller



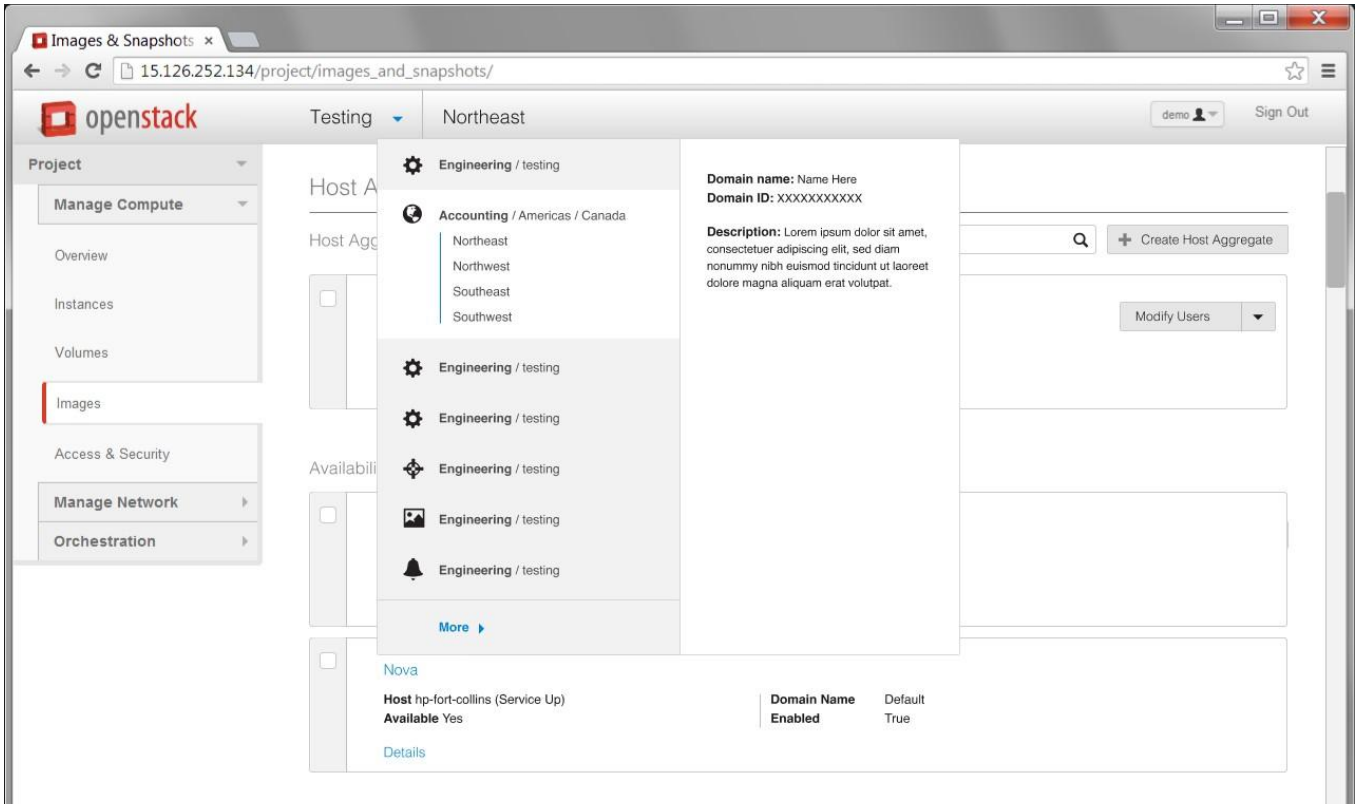
Next steps: Nested Quota

- Already works on Cinder
- Under review in Nova
  - But there is work to do: a common quota library for all services
  - Contributors are welcome
    - Contribute code
    - Share use cases
    - Review!
    - Channel: #openstack-quota

Next steps: UX



Next steps: UX



Besides us, a couple more users believe this is pretty relevant

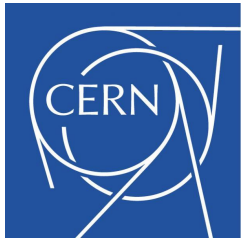

redhat


Hewlett Packard
Enterprise


YAHOO!


IBM


UFCG
SUPRA OMNES LUX LUCES


CERN

Thank you!

Flat no more!
Hierarchical multitenancy and projects acting as domains in OpenStack

{andrey, henrique, raildo}@lsd.ufcg.edu.br
IRC: abrito, htruta, raildo