Mix & Match: Resource Federation

Kristi Nikolla
Massachusetts Open Cloud

The Massachusetts Open Cloud

- Multiple Landlords: BU, MIT, Northeastern, Harvard, UMass
  - Universities want to administer their own hardware
  - Each university has their own auth framework, and will not trust a centralized Keystone
  - So they will want to set up OpenStack themselves
- Open Cloud eXchange
  - Competing service providers standing up services in their own OpenStack deployments
  - Users can combine resources from different service providers: “mix and match”

Resource Federation

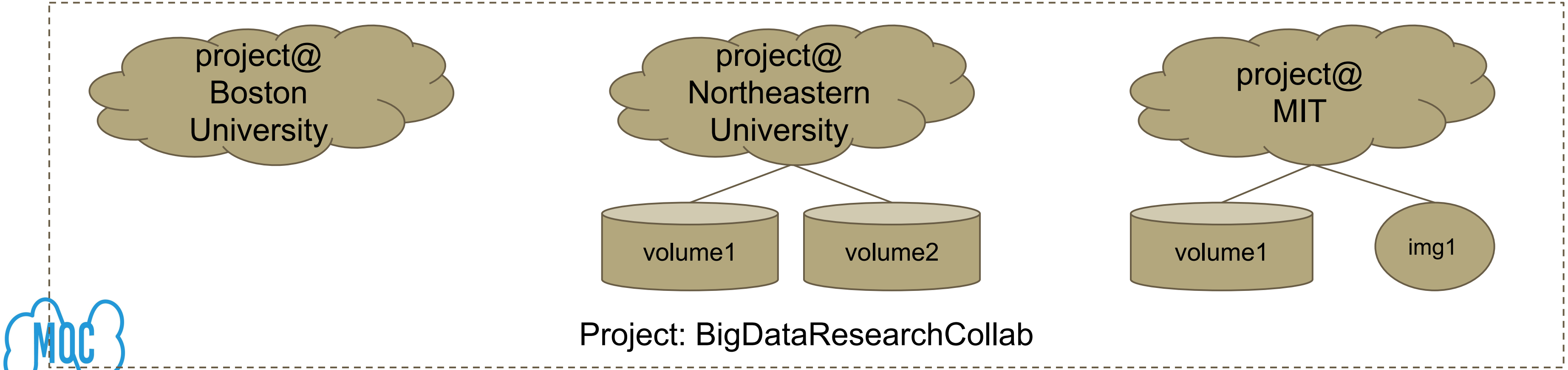
- Allow OpenStack services to consume resources from services in other OpenStack deployments
  a. Resources are volumes, images, snapshots, etc.
- **Resource Federation is the first step towards OCX**

Challenges

- Preserving API and user experience
  - Combine information from multiple providers
  - Uniquely qualifying resources
- Authentication and authorization
- Security
- Scalability
- Performance

Combining information: Meta-Projects

- Every resource is owned by a project
- Projects are mapped with each other to form a meta-project
- User is presented with **combined view** of all resources in meta-project

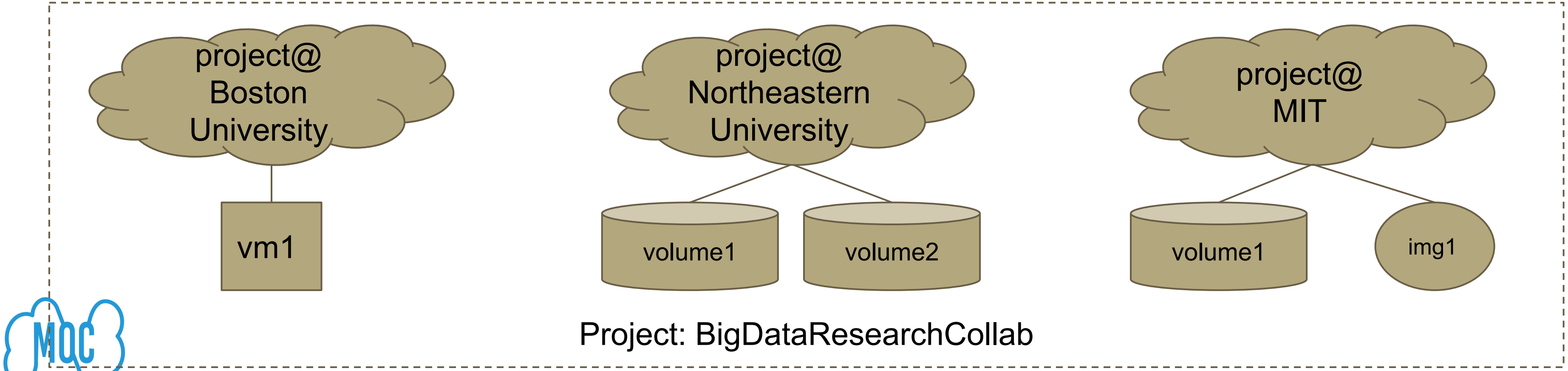

project@ Boston University
project@ Northeastern University
volume1
volume2
project@ MIT
volume1
img1
Project: BigDataResearchCollab

Uniquely Qualifying Resources

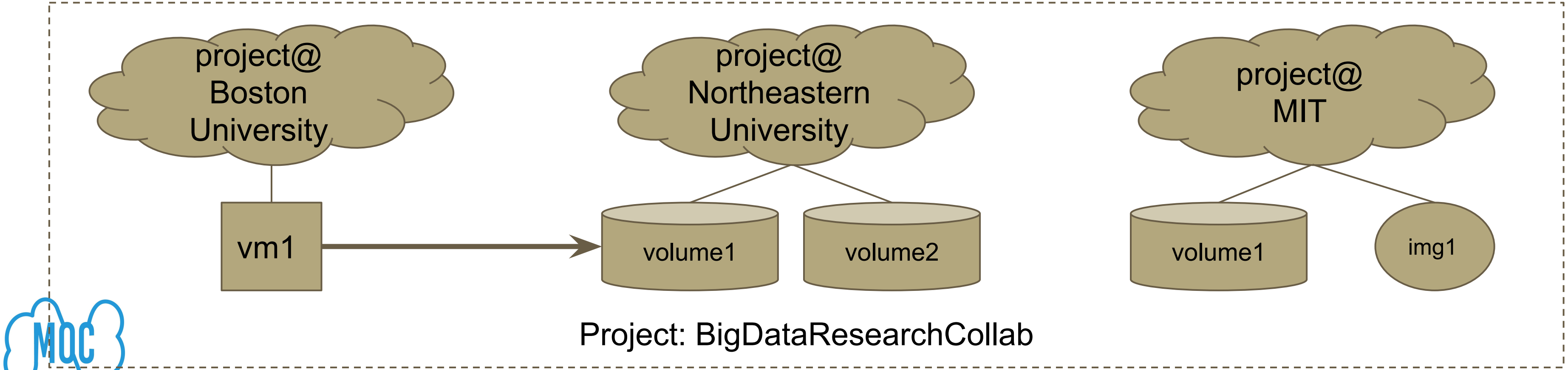
- Everything in OpenStack is identifying by a UUID
- UUIDs are unique, even across multiple service providers
    - We didn’t need to change the API to uniquely qualify the **target resource**
    - We can **combine** without naming conflicts

```
$ openstack volume list
```

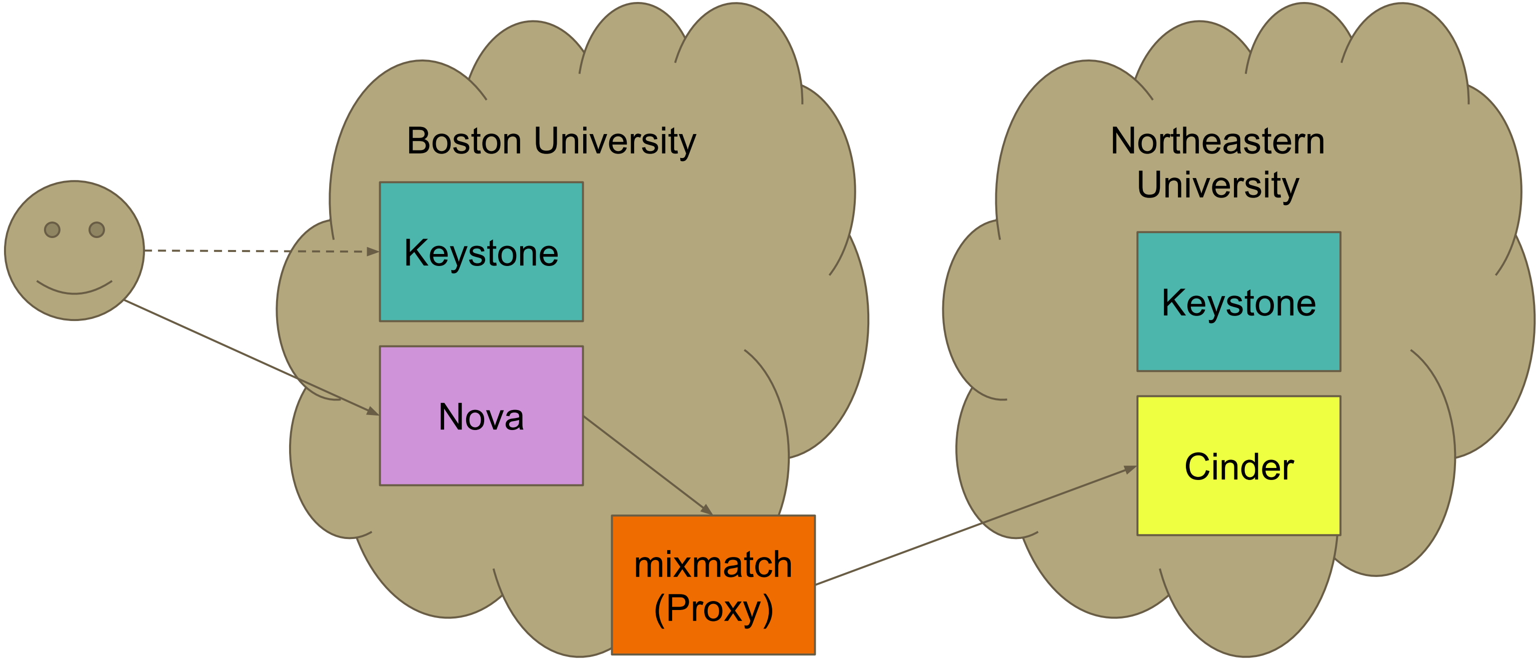
ID	Volume Name	Service Provider
3294C96D...831DBCCB1F73	volume1	Northeastern University
AFB5236E...768B8BF5801C	volume2	Northeastern University
890DD196...C017D93E1AA3	volume1	MIT


project@ Boston University
vm1
project@ Northeastern University
volume1
volume2
project@ MIT
volume1
img1
Project: BigDataResearchCollab

```
$ openstack server add volume vm1 3294C96D...831DBCCB1F73
```

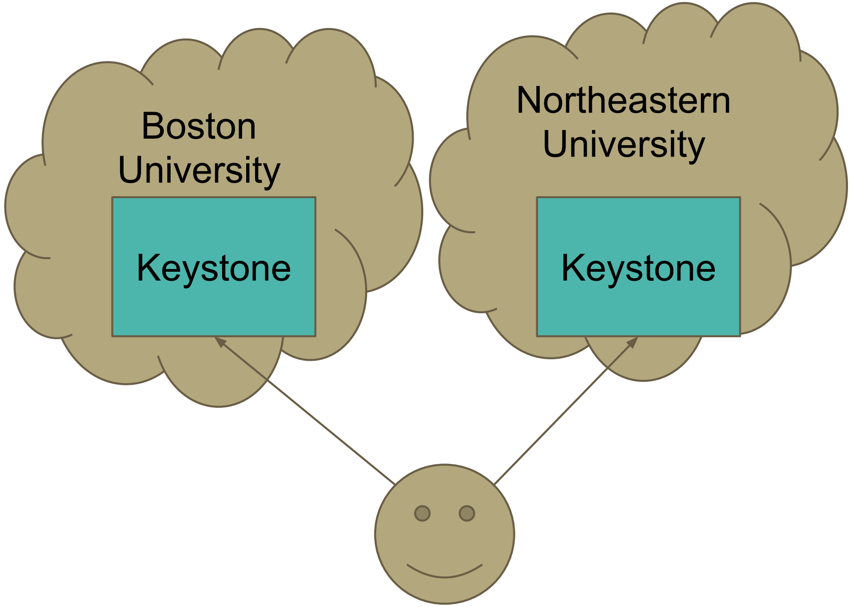


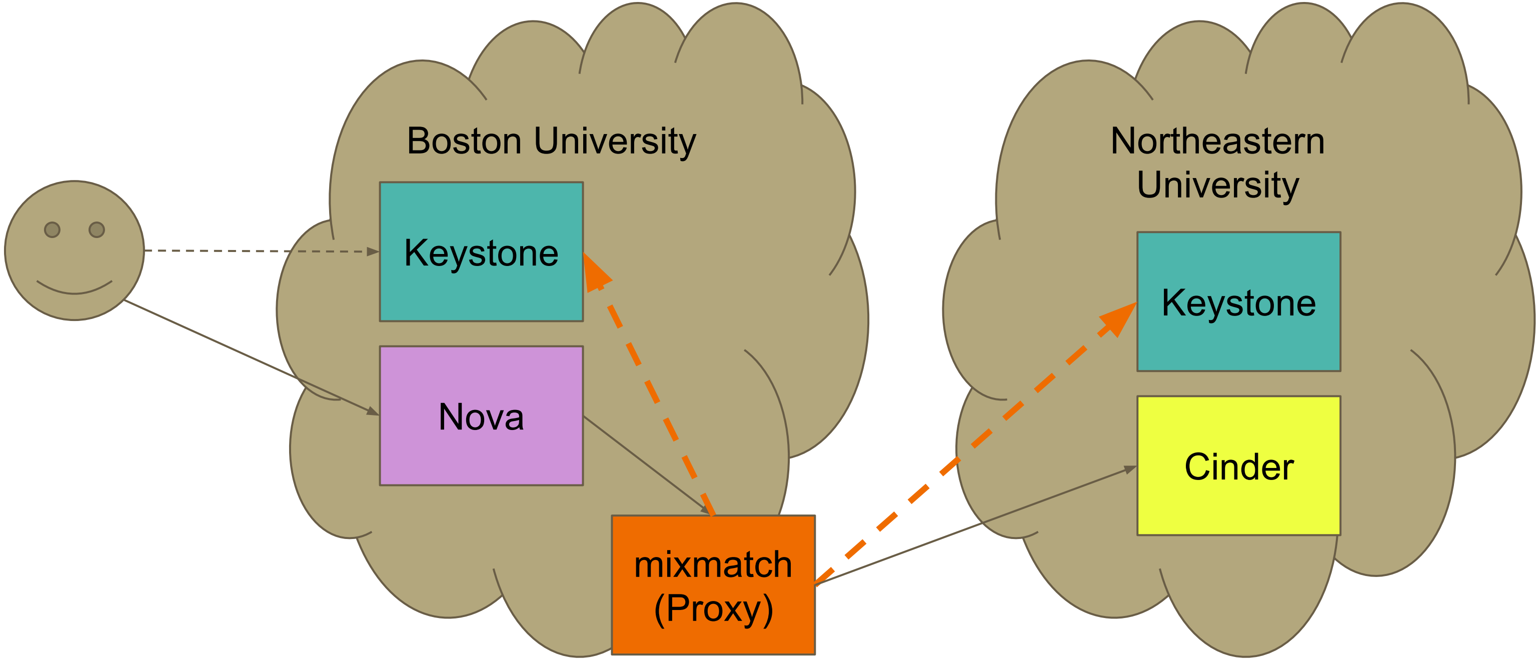
Crossing boundaries

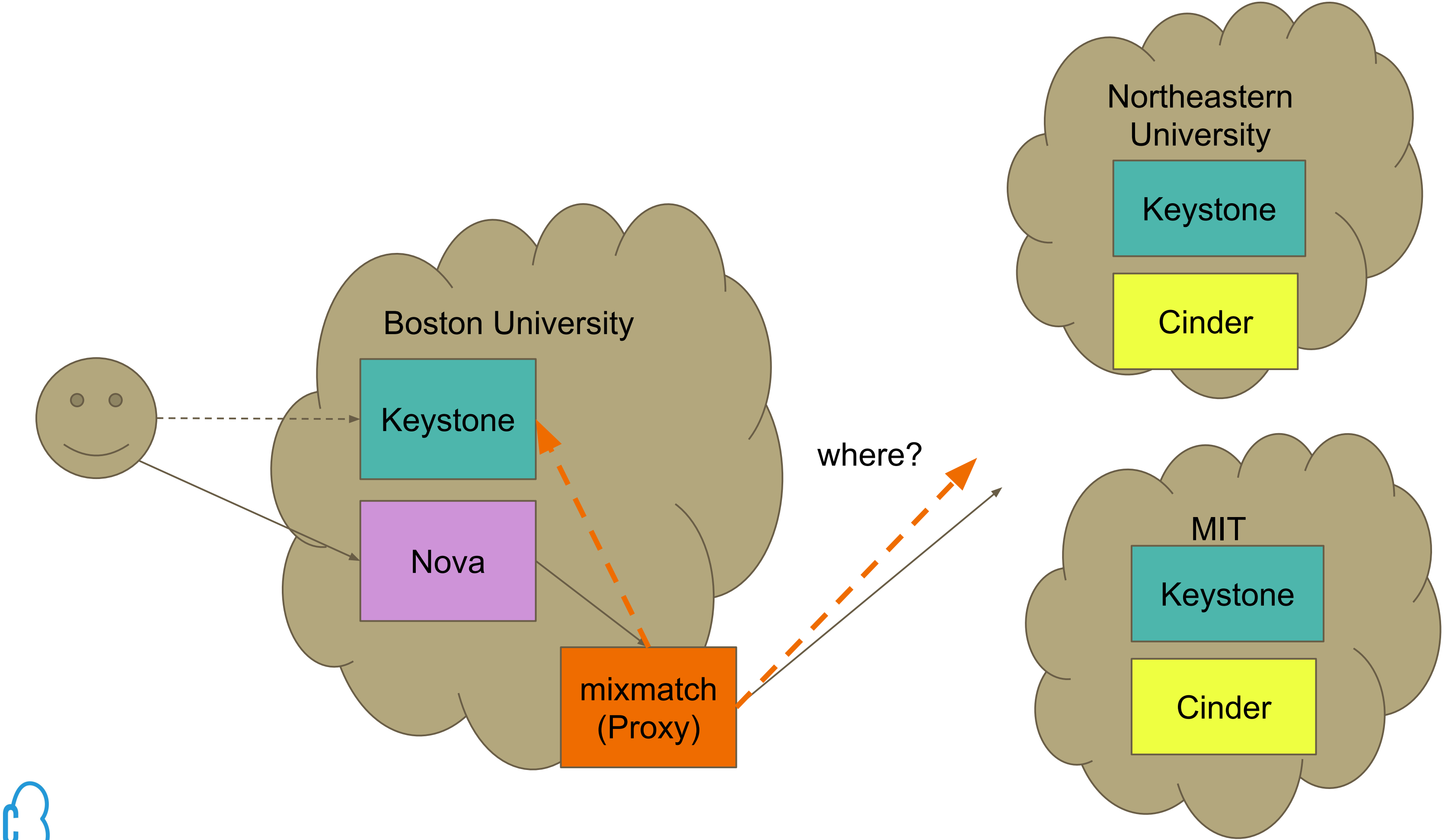


Authentication and Authorization

- Keystone-to-Keystone federation
- SAML2 assertion contains user attributes
  - Keystone maps roles on projects based on those attributes
  - **We exploit this to implement the meta-project**




Boston University
Keystone
Nova
mixmatch
(Proxy)
Northeastern
University
Keystone
Cinder


Northeastern University
Keystone
Cinder
Boston University
Keystone
Nova
mixmatch (Proxy)
where?
MIT
Keystone
Cinder

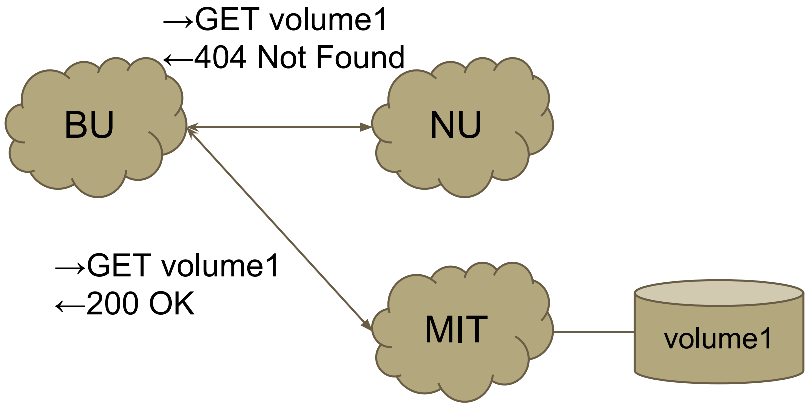
How It Works

- Every request in OpenStack is done through the REST API
  - Resource UUID are a **predictably located** part of the URL
  - Proxy analyzes URL for UUID

Call	Action
GET w/o UUID	Aggregate
GET w UUID	**Find resource**
PUT/PATH w UUID	**Find resource**
DELETE w UUID	**Find resource**
POST	Be more explicit? **Header API** to the proxy from the client

Finding Resources

- Search by broadcasting
  - Proxy will query service providers until it finds the resource with the requested ID.
  - Does not scale to many SPs



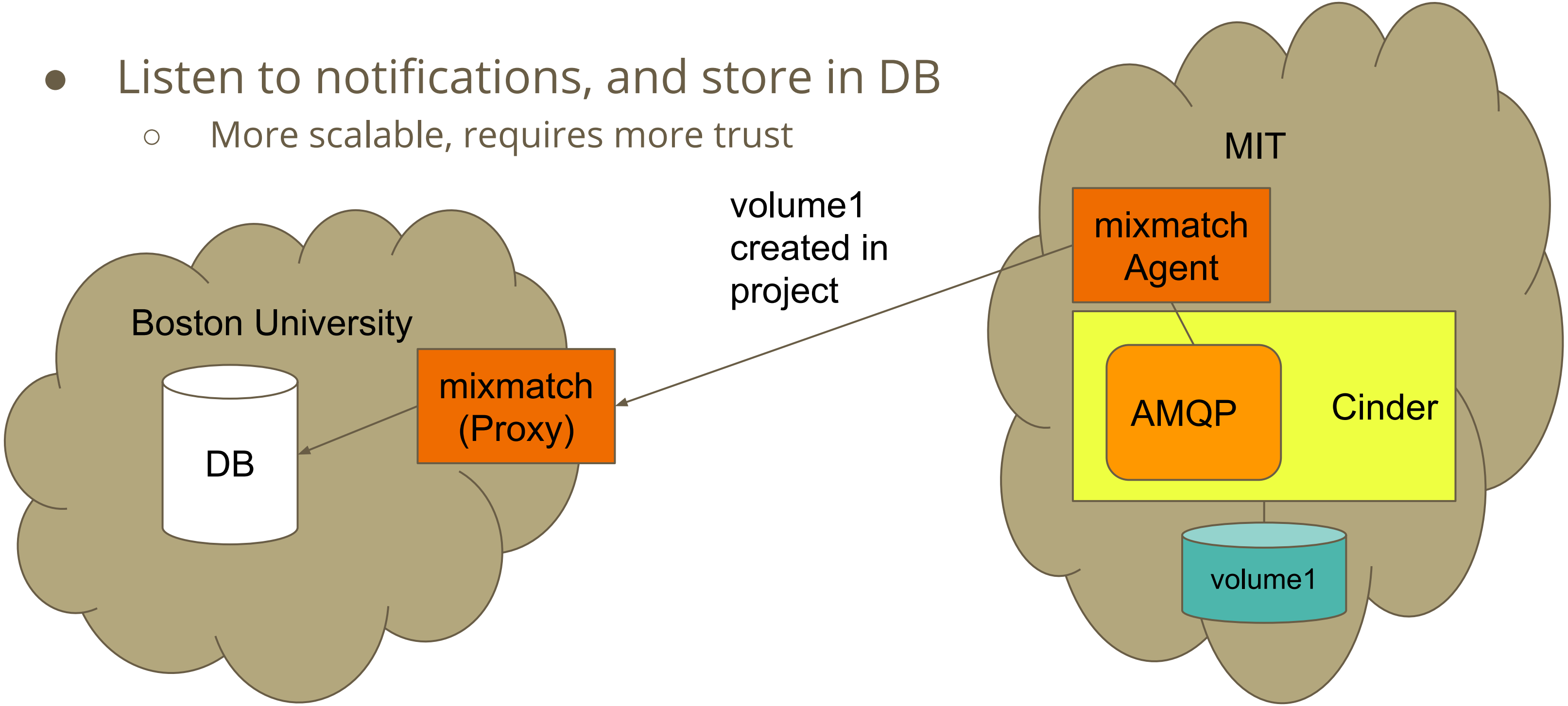
Performance Improvements

- Cache Tokens
  - Local Token → Service Provider, Project ID, Remote Project
- Cache Resource Mappings in DB after finding resources

Ideally, proxy should already know the location...

Finding Resources (part 2)

- Listen to notifications, and store in DB
  - More scalable, requires more trust



Data plane

- No performance degradation in data plane
- iSCSI
  - Just works™
  - Credentials for the volume are passed in API calls, so no more access is granted than needed.
- Ceph/RBD
  - Works, however...
  - All compute nodes must have all Ceph authentication keys
  - This requires a high amount of trust between service providers
  - We’re working with the Ceph developers to address these issues

Beyond Open Cloud eXchange

- Adding experimental services to a production cloud
- Partial upgrade of cloud services—standing up multiple versions at once
- Defense in depth—limiting scope of a security breach

Future Work

- Deploying in production
- Security
  - More granular permission model for Ceph/RBD
  - Limit information exposed from proxy agent
- Federation of networks across service providers
- Testing cross-attach with other Cinder backends
- Benchmarking the API overhead
- Becoming an official OpenStack project

Check us out!

http://info.massopencloud.org/blog/mixmatch-federation

https://github.com/openstack/mixmatch