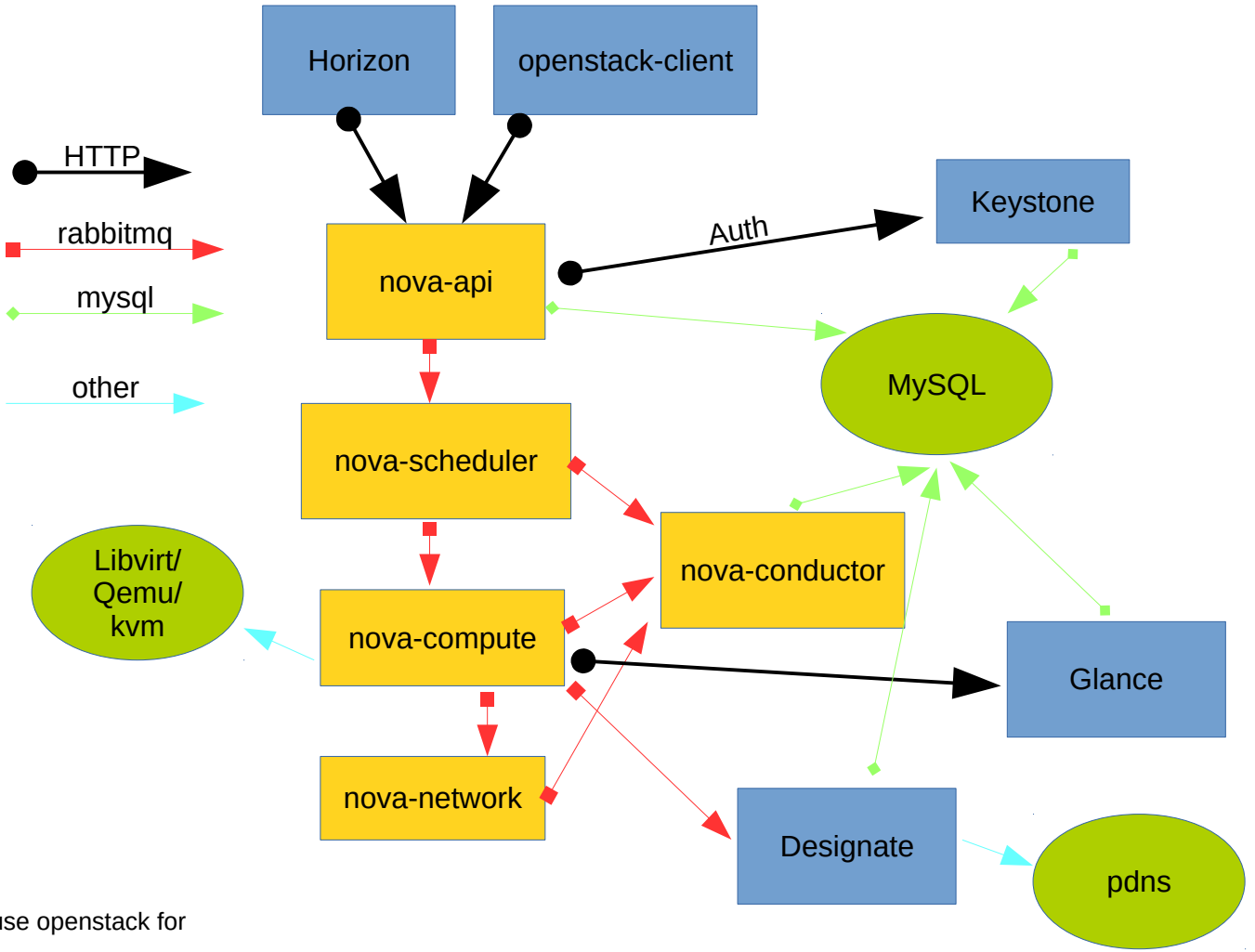
OpenStack at WMCS



- What is openstack
- What does WMCS use openstack for
- Details about specific projects and services

What is OpenStack

- APIs for managing Infrastructure as a Service
- A Python implementation of those APIs
- Bindings, libraries, UIs for accessing the APIs

OpenStack Manages Things

- VMs (aka 'instances')
- Projects (aka 'tenants')
- Images (used to create new instances)
- Users
- DNS Zones and Records
- Relationships among all of the above

OpenStack Includes Many Projects!

- Currently 21 different projects listed on https://en.wikipedia.org/wiki/OpenStack
- Quite a few of those projects exist only to configure the other projects.
- OpenStack has an open-door policy so a lot of their projects are half-baked.

We use...

- Nova (for virtualization)
- Keystone (for identity, tokens, roles, etc.)
- Glance (for image management)
- Designate (DNS as a service)
- Neutron (For networking – coming soon!)
- Horizon (Web UI for all of the above)

We don't use...

- Any of the configuration projects – everything is configured using home-made puppet classes.
- Any of OpenStack's K8s management projects

- Swift (it's used /on/ Vms and elsewhere in at the WMF but not coupled to any other services)
- Ceph (not an OpenStack project, but part of every OpenStack deploy but ours)

OpenStack uses...

- Python
- MySql, hosted on misc db cluster
- RabbitMQ, hosted on labcontrol1001

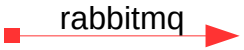
OpenStack doesn't do much

- OpenStack is interfaces for other miscellaneous backends.

- Actual virtualization is qemu/kvm
- Actual DNS is pdns
- Etc.

- The good news is that failure or restart of an OpenStack service doesn't break anything that's already running.

RabbitMQ

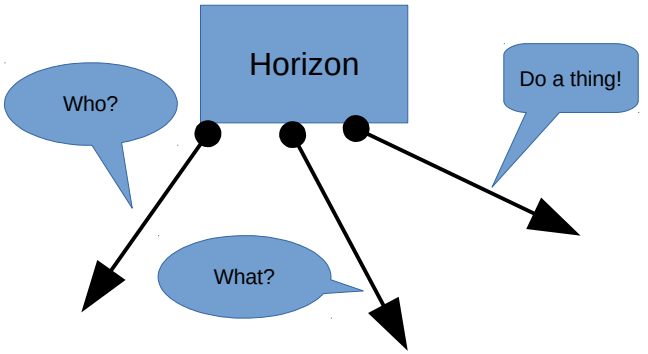


- All backend communication within a given OpenStack project happens on RabbitMQ (e.g. between nova-api and nova-compute)
- Some inter-project communication also happens on RabbitMQ via pub/sub (e.g. between Nova and Designate)

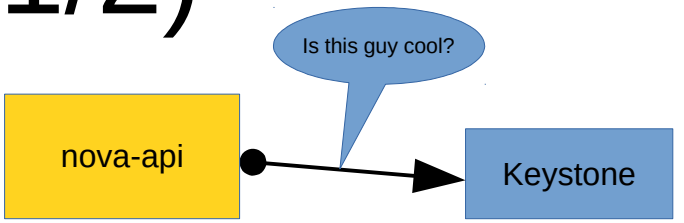
- Runs on labcontrol1001
- Historically fragile and leaky but seems stable lately
- Restarting will cause lots of other services to freak out

Horizon

- A collection of dashboards for talking to OpenStack APIs
- APIs are stable so we don't need to keep the Horizon version in sync with the other project versions.
- Heavily customized for our use
- Includes some in-house dashboards, e.g. puppet and proxy management
- Historically written in Python/Django but some parts are now rewritten in AngularJS
- Runs on labweb1001 and labweb1002

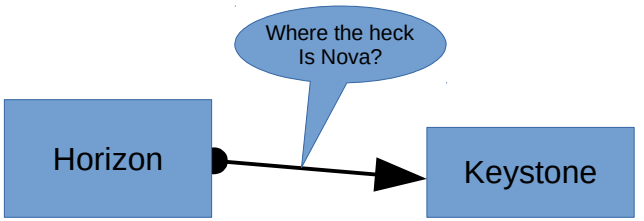


Keystone (1/2)



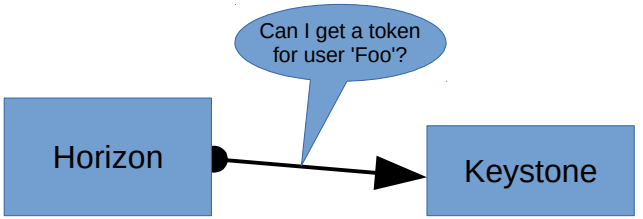
- Aka 'Identity'
- Manages Projects, Users, and Roles
- Provides discovery and auth for all other services

- Somewhat crippled by early design decisions, e.g. no way to set global roles



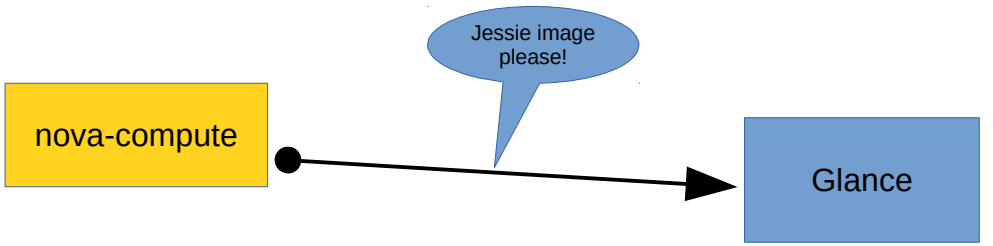
Keystone (2/2)

- Just one service, 'keystone-all'
- Runs on labcontrol1001
- Logs to /var/log/keystone/keystone.log
- Largely harmless to restart



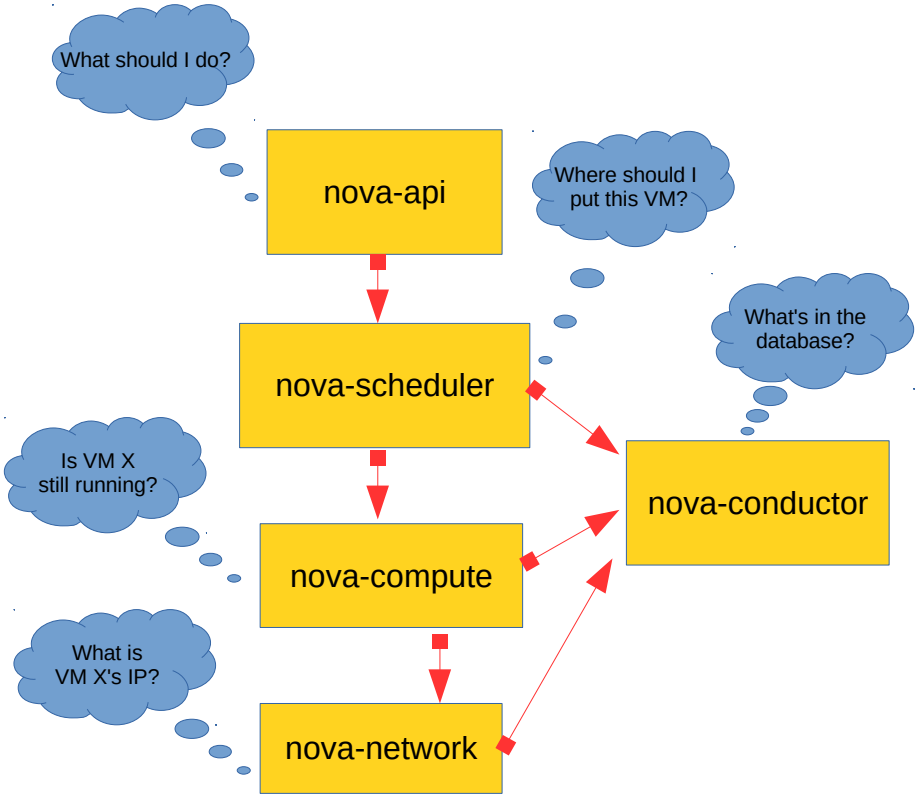
Glance

- Manages and serves base images for VMs
- Runs on labcontrol1001
- Two services: glance-registry and glance-api
- Logs to /var/log/glance/api.log and /var/log/glance/registry.log
- Could use swift or ceph, but we just use local files plus a cronjob to keep things backed up with labcontrol1002
- Simple, stable, hasn't changed noticeably in years



Nova

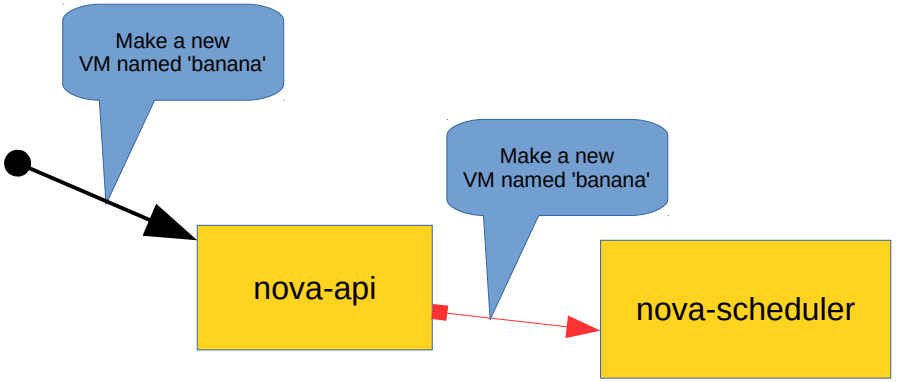
- Virtualization
- Lots of services:
  - Api
  - scheduler
  - compute
  - network
  - Conductor



- Log to /var/log/nova/nova-*.log
- All services configured via unified /etc/nova/nova.conf

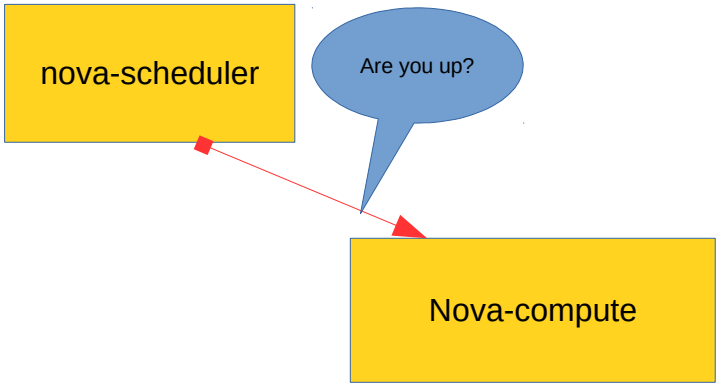
Nova-api

- The REST api for all nova services
- Authenticates with keystone
- Runs on labnet1001
- Uses its own database (on the misc cluster)
- Log is a good place to monitor nova activity. /var/log/nova/nova-api.log



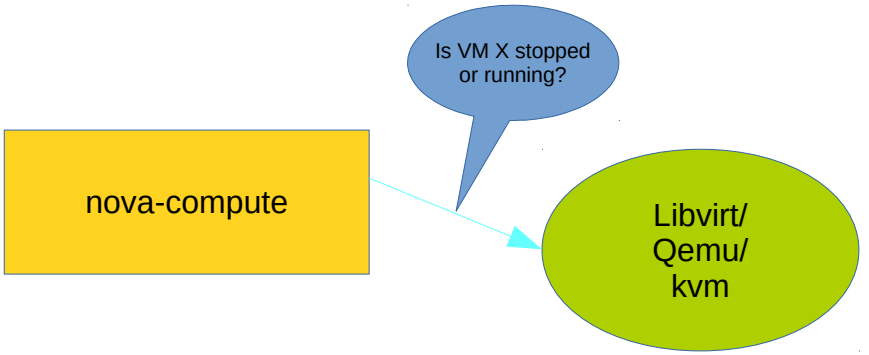
Nova-scheduler

- Monitors the state of each compute node
- Chooses compute hosts for new VMs based on availability and lots of custom filters (disk space, CPU load, etc.)
- When instance creation is failing, this is the first place to look! /var/log/nova/nova-scheduler.log
- Log is easy to follow and often helpful
- Runs on labcontrol1001
- Perfectly safe to restart



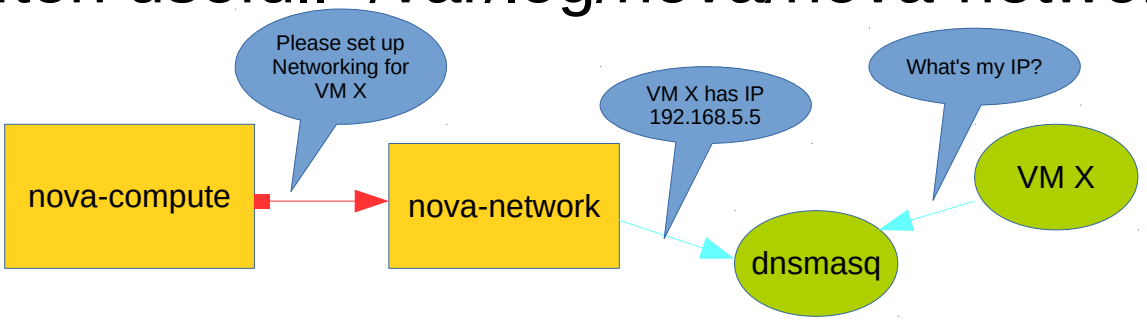
Nova-compute

- Creates, monitors, destroys instances.
- Talks to the scheduler on one end and libvirt on the other
- Does not actually host the VMs. If it crashes, VMs are unaffected
- One nova-compute service running on each labvirt host
- Lots to /var/log/nova/nova-compute.log
- But, the really interesting stuff is in the per-instance logs in /var/log/libvirt



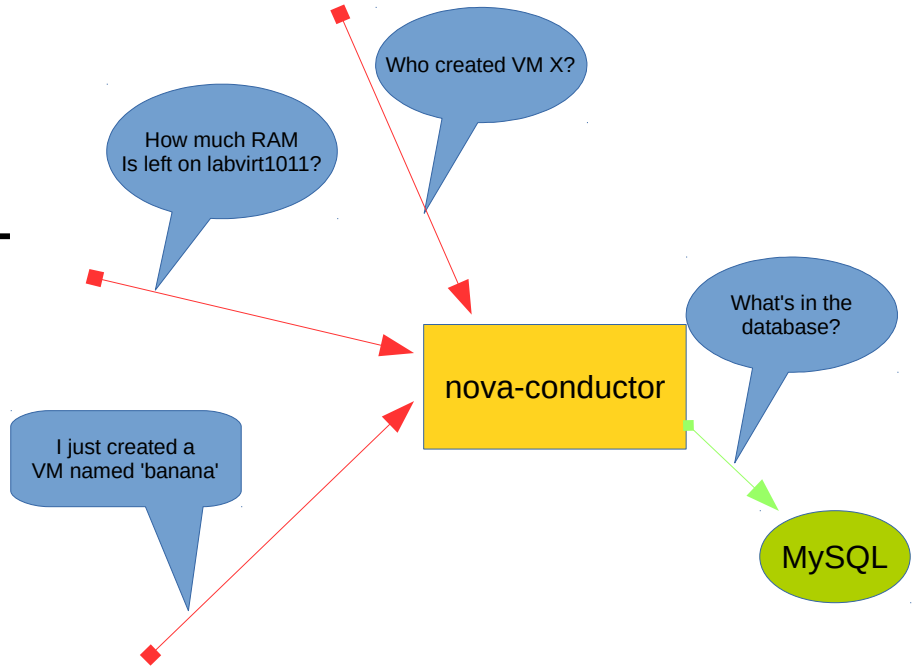
Nova-network

- Configures bridges and a dhcp server for all VM network connections.
- Soon to be replaced by much-more-complicated OpenStack Neutron.
- Has a pet dnsmasq service that handles dhcp for new Vms.
- Runs on labnet1001
- Restarting nova-network does not interrupt network service for existing Vms
- Log is often useful. /var/log/nova/nova-network.log



Nova-conductor

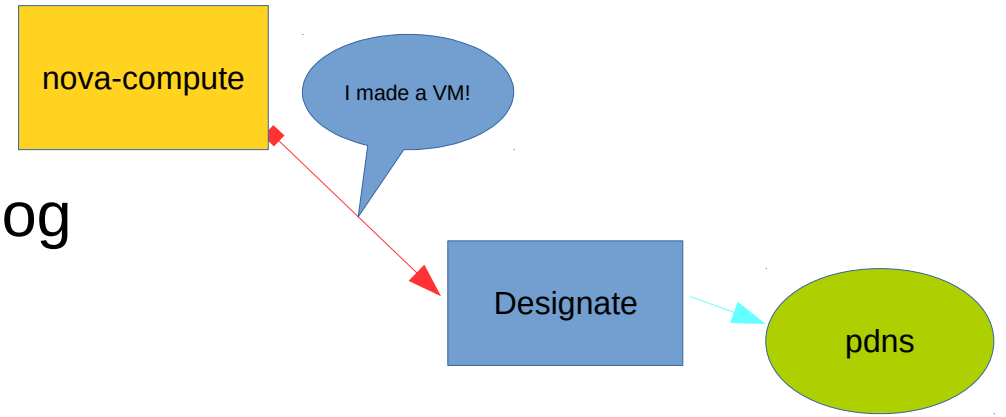
- Marshals all database activity
- Services requested db calls via rabbitmq
- Only conductor ever talks directly to mysql (except for nova-api which has its own database)
- Runs on labcontrol1001



Designate (1/2)

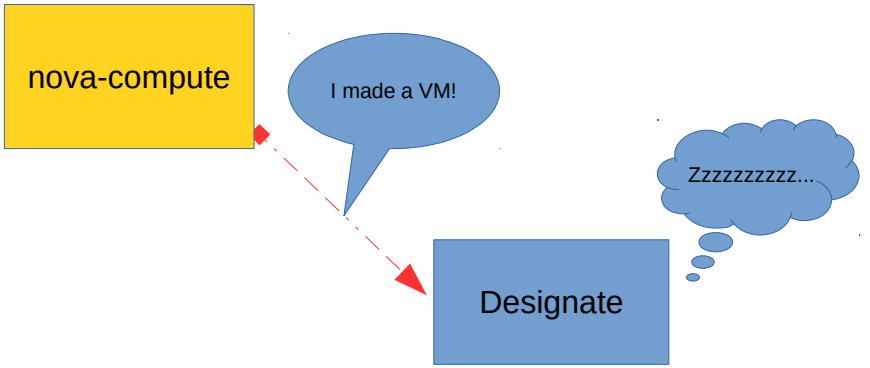
- Lots of services!
- Designate-api, for configuring DNS entries (with to Horizon)
- Designate-sink, creating new DNS entries for new Vms (listens for instance creation notification on RabbitMQ)
- Designate-mdns, a little DNS server that initiates XFR synchronization with the real PDNS backend
- And others :(

- Runs on labservices1001
- /var/log/designate/designate-*.log



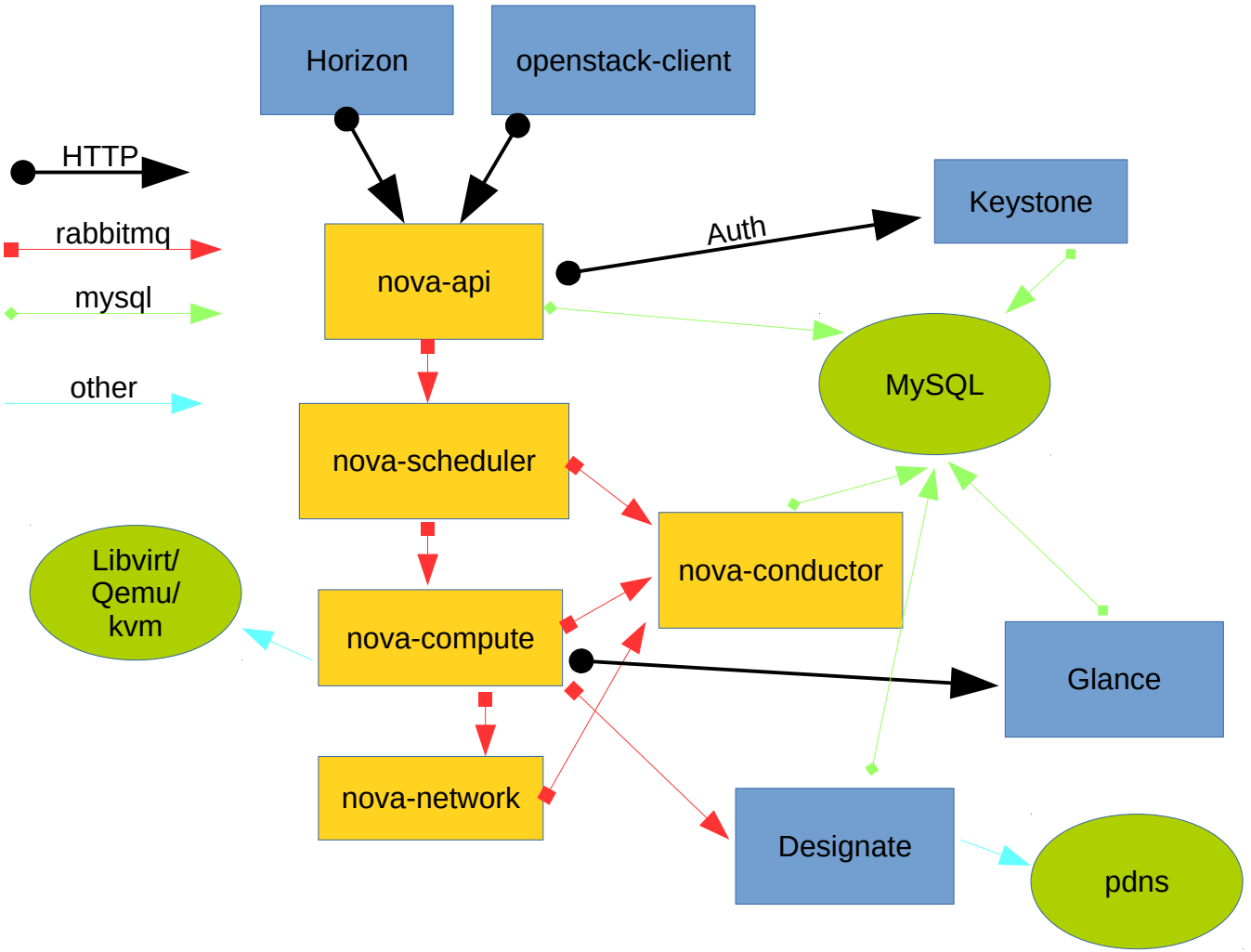
Designate (2/2)

- Not tightly-coupled with instance creation, can leak entries or fail to notice new VM creation.
- If Designate is broken, new VMs don't get DNS entries and never will



- Project still feels very young, subject to constant redesign

That's not everything

- Instance auth (ssh, sudo, etc)
- Instance NFS
- Instance Puppet
- Lots of fiddly DNS bits
- Dynamic instance proxies
- Things I'm forgetting

OpenStack at WMCS