MIRANTIS

OpenStack Architecture

BUILT FOR openstack™ CLOUD SOFTWARE

Polina Petriuk
Sr. Technical Trainer
Mirantis Inc.

Release H2.0.3

OpenStack Architecture

Provision a VM Request Flow

VM Provisioning

- Is the most common and complex process in OpenStack
- Involves interaction of most of OpenStack components

Initial State

Cloud Operator, DevOp, etc.

Assumes Project is created, provisioning quota is available, user has an access to Horizon/CLI

UI: Horizon or CLI

Nova

Nova API

Scheduler

Conductor

Queue

Nova DB

Compute Node

nova-compute

VM

Hypervisor

Network

Keystone

KeystoneAPI

Keystone DB

Glance

Glance API

Glance DB

Glance Registry

Cinder

Queue

Cinder DB

Cinder Vol

Cinder API

Scheduler

Cinder Backup

Neutron

Neutron API

Scheduler

Plugin/Agent

Queue

Neutron DB

Swift

Proxy Server

Object Store

Block Storage

Storage

Network Node

DHCP/IPAM

Router/GW

Ceilometer

Ceilometer API

Agent

Collector

MIRANTIS

Step 1: Request Provisioning – From UI

- Login to Horizon
- Specify parameters of VM
  - VM Name
  - Image (OS type)
  - Flavor (specifies CPU, Memory, Disk)
  - Network (required for Neutron)
    - Optional (SSH Keys, Persistent volumes, comments, etc.)
- Select "Create" button

MIRANTIS

The OpenStack Dashboard (Horizon)

Horizon provides a

baseline user interface

for managing OpenStack services.

MIRANTIS

Horizon

- Is “stateless” — doesn’t require a database
- Delegates error handling to the back-end
- Doesn’t support all the API functions
- Can use memcached or database to store sessions
- Gets updated via API polling

Step 1: Request VM Provisioning via UI/CLI

Cloud Operator, DevOp, etc.

User logs in to UI
Specifies VM params: name, flavor, keys, etc. and hits "Create" button

UI: Horizon or CLI

Nova

- Nova API
- Scheduler
- Conductor
- Queue
- Nova DB

Compute Node

- nova-compute
- VM
- Hypervisor
- Network

Keystone

- KeystoneAPI
- Keystone DB

Glance

- Glance API
- Glance DB
- Glance Registry

Cinder

- Queue
- Cinder DB
- Cinder Vol
- Cinder API
- Scheduler
- Cinder Backup

Neutron

- Neutron API
- Scheduler
- Plugin/Agent
- Queue
- Neutron DB

Swift

- Proxy Server
- Object Store

Block Storage

- Storage

Network Node

- DHCP/IPAM
- Router/GW

Ceilometer

- Ceilometer API
- Agent
- Collector

MIRANTIS

Step 1: Request Provisioning – Under the Hood

- Form parameters are converted to POST data
- "Create" request initiates HTTP POST request to back-end
  - To Keystone if auth token is not cached – step 2

The OpenStack Identity Service (Keystone)

Keystone provides

Identity, Token, Catalog and Policy services

for use specifically by projects in the OpenStack family.

MIRANTIS

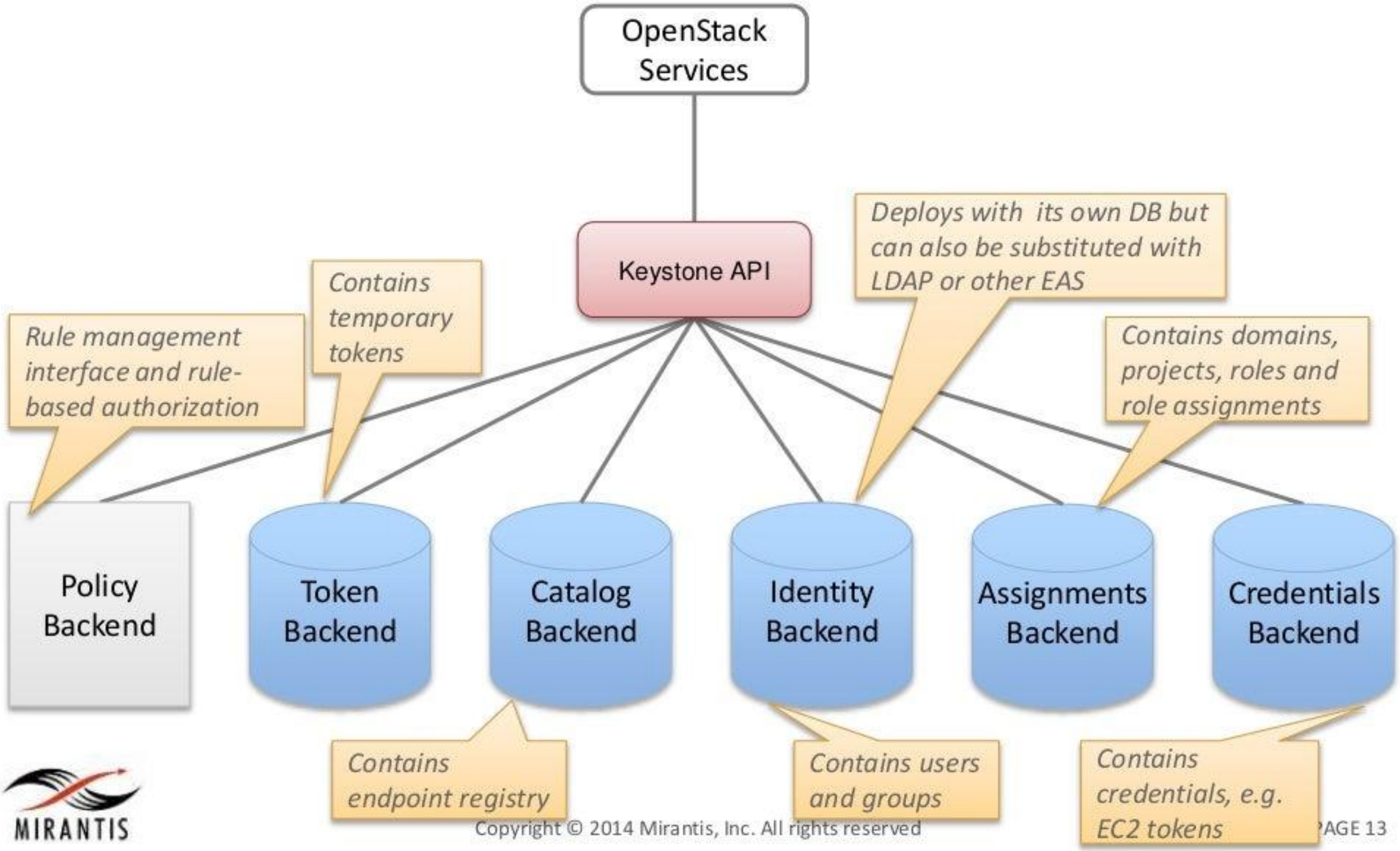
Keystone: Identity Management

- User
- Credentials
- Token
  - Associated with a user, an arbitrary bit of text that is used to access resources
- Group of users
- Project
  - Synonym to tenant
- Role
  - Assigned to sers or groups for projects
- Domain
  - Higher level of hierarchy – users and projects belong to domains

Keystone: Service Catalog

- Service
  - An OpenStack service, such as Compute (Nova), Object Storage (Swift), or Image Service (Glance).
- Endpoint
  - A network-accessible address, usually described by URL, from where you access an OpenStack service
- Rule
  - A set of requirements for performing an action over the endpoint.

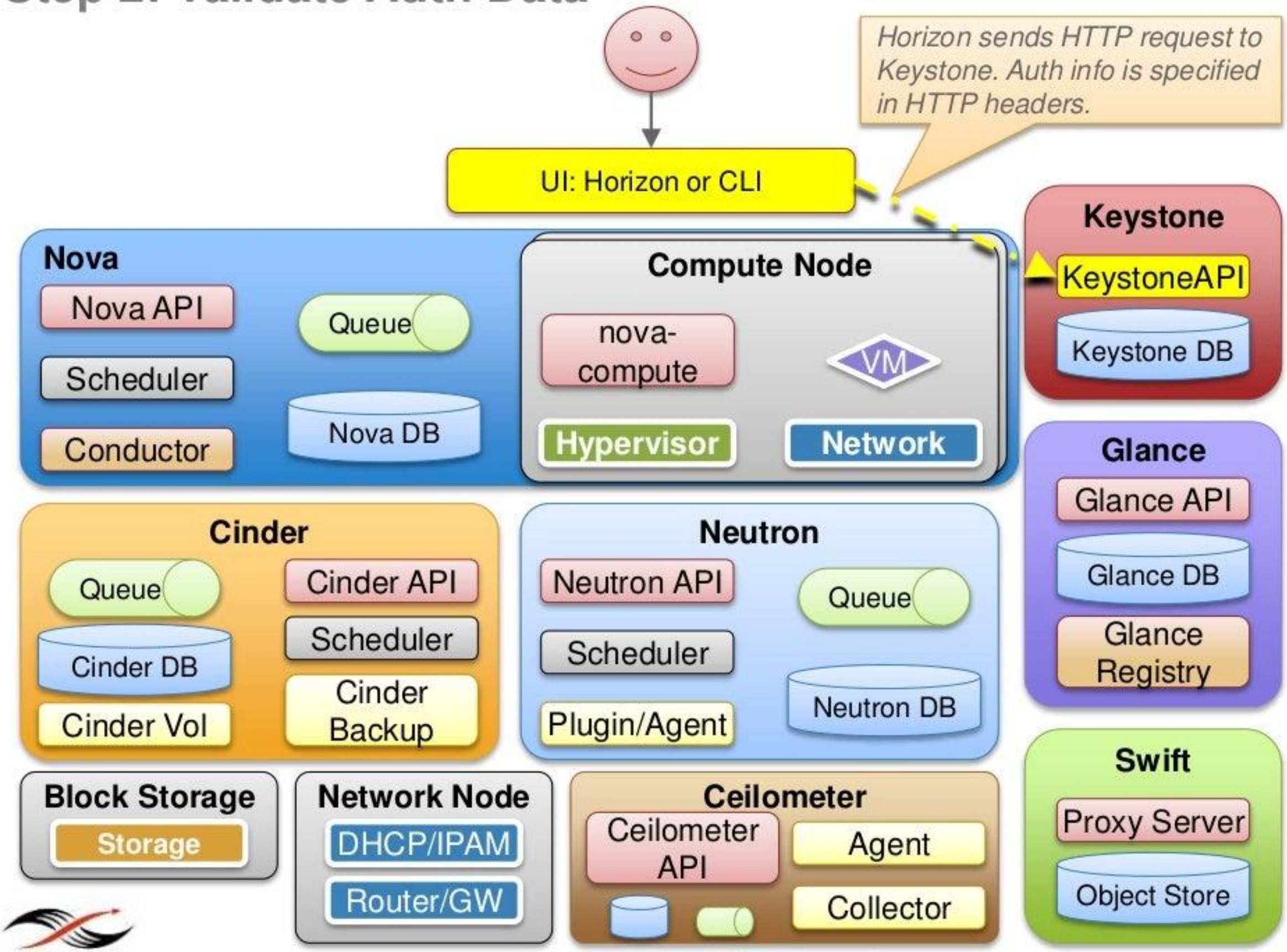
Keystone Architecture



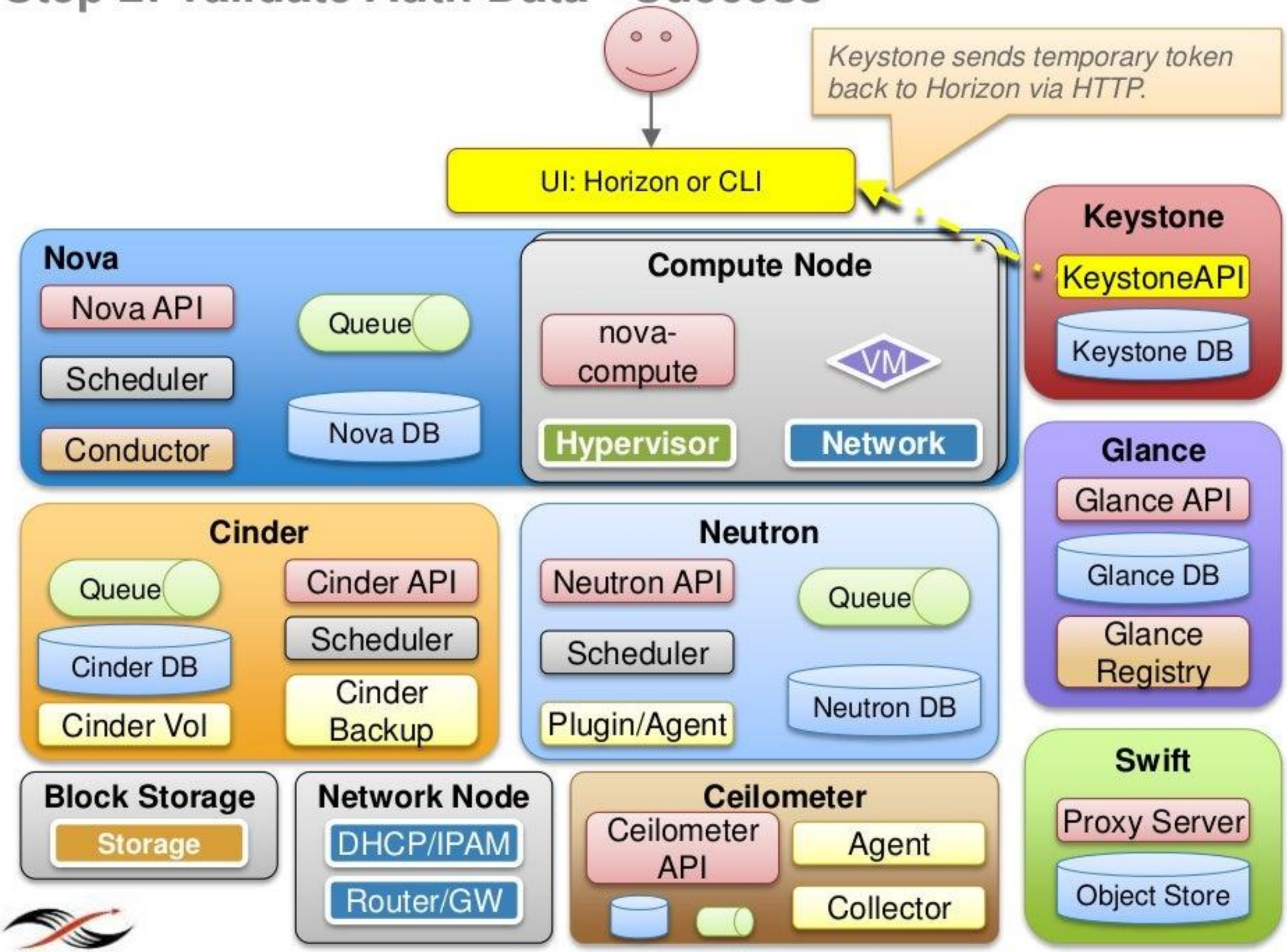
Keystone:Role Based Access Control (RBAC)

1. User gets Token from Keystone.
2. Token includes the list of user Projects and Roles in them.
3. User calls the Service specifying the Token.
4. Service interprets the Roles:
   - Service consults its policy.json file.
   - Policy.json specifies the list of available rules.
     - "admin_required": [["role:admin"], ["is_admin:1"]],
     - “owner" : [["project_id:%(project_id)s"]],
     - "admin_or_owner": [["rule:admin_required"], ["rule:owner"]],
   - Policy.json specifies which rules are enforced for operations and resources.
     - "volume:create": [["rule:admin_or_owner"]],

Step 2: Validate Auth Data

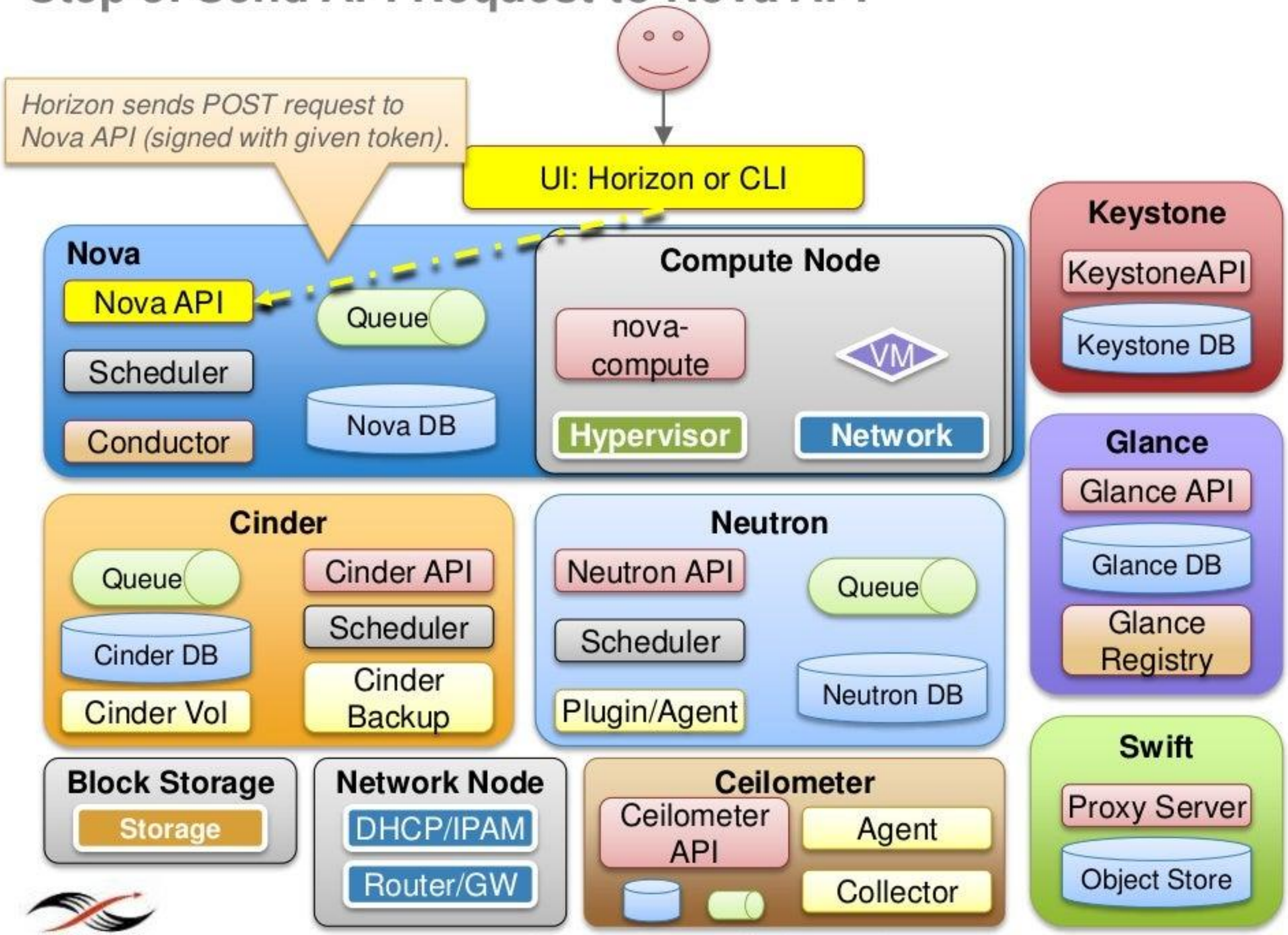


Step 2: Validate Auth Data - Success



MIRANTIS

Step 3: Send API Request to Nova API



The OpenStack Compute API (Nova API)

Nova API is a

RESTful API web service

which is used to interact with Nova.

Nova API

- Exposes REST API via HTTP
- Provides system for managing multiple APIs on different sub-domains:
  - EC2-compatible—starting to be deprecated
  - Compute API—all innovation happens here
- Is the only "allowed" way to interact with Nova
- Is “stateless”

Step 4: Validate API Token