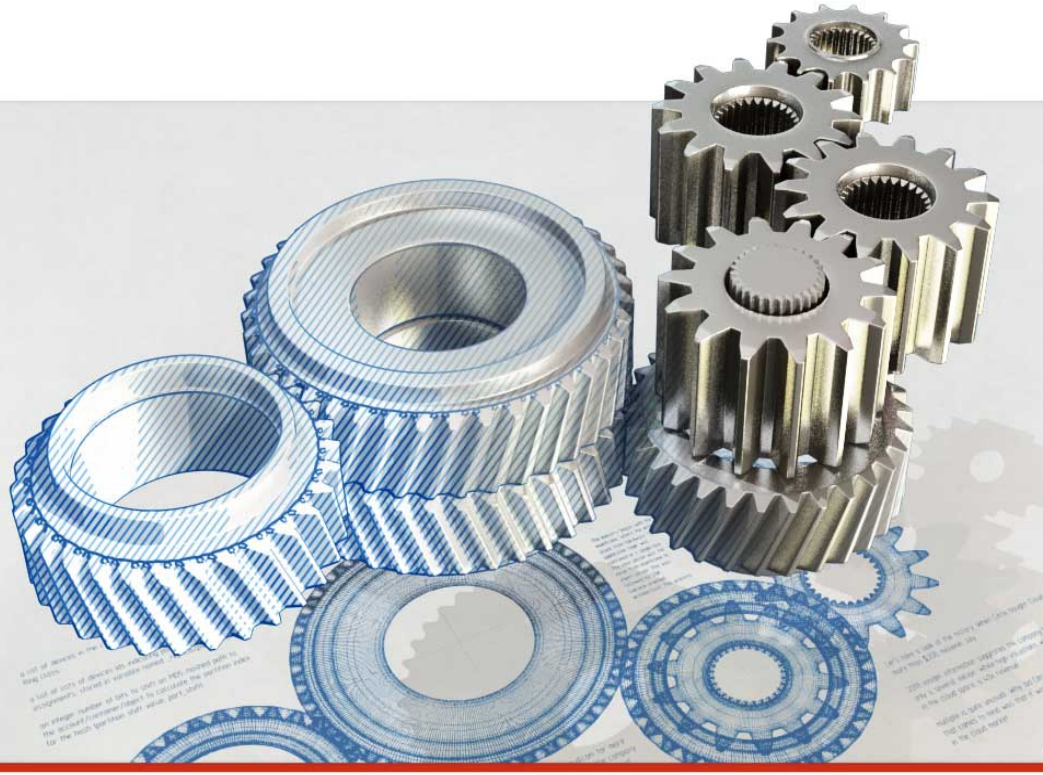
MIRANTIS

OpenStack in Action

Architecture and Request Flow



Mirantis

September 2012

What is OpenStack?

"Open source software for building private and public clouds"

MIRANTIS

© Mirantis, Inc, 2012. All rights reserved.

OpenStack capabilities

- VMs on demand
  - provisioning
  - snapshotting
- Volumes
- Multi-tenancy
  - quotas for different users
  - user can be associated with multiple tenants
- Object storage for VM images and arbitrary files

MIRANTIS

© Mirantis, Inc, 2012. All rights reserved.

OpenStack History

- July 2010 - Initial announcement
- October 2010 - Austin Release
- February 2011 - Bexar Release
- April 2011 - Cactus Release
- October 2011 - Diablo Release
- **April 2012 - Essex Release**
- October 2012 - Folsom Release

MIRANTIS

© Mirantis, Inc, 2012. All rights reserved.

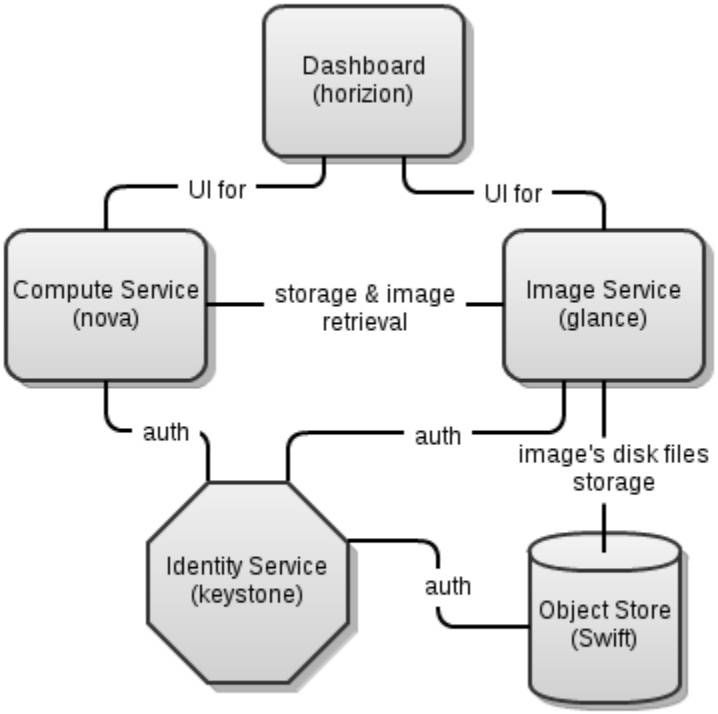
OpenStack Projects

- Nova (Compute)
- Glance (Image Service)
- Swift (Object Store)
- Keystone (auth)
- Horizon (Dashboard)

MIRANTIS

© Mirantis, Inc, 2012. All rights reserved.

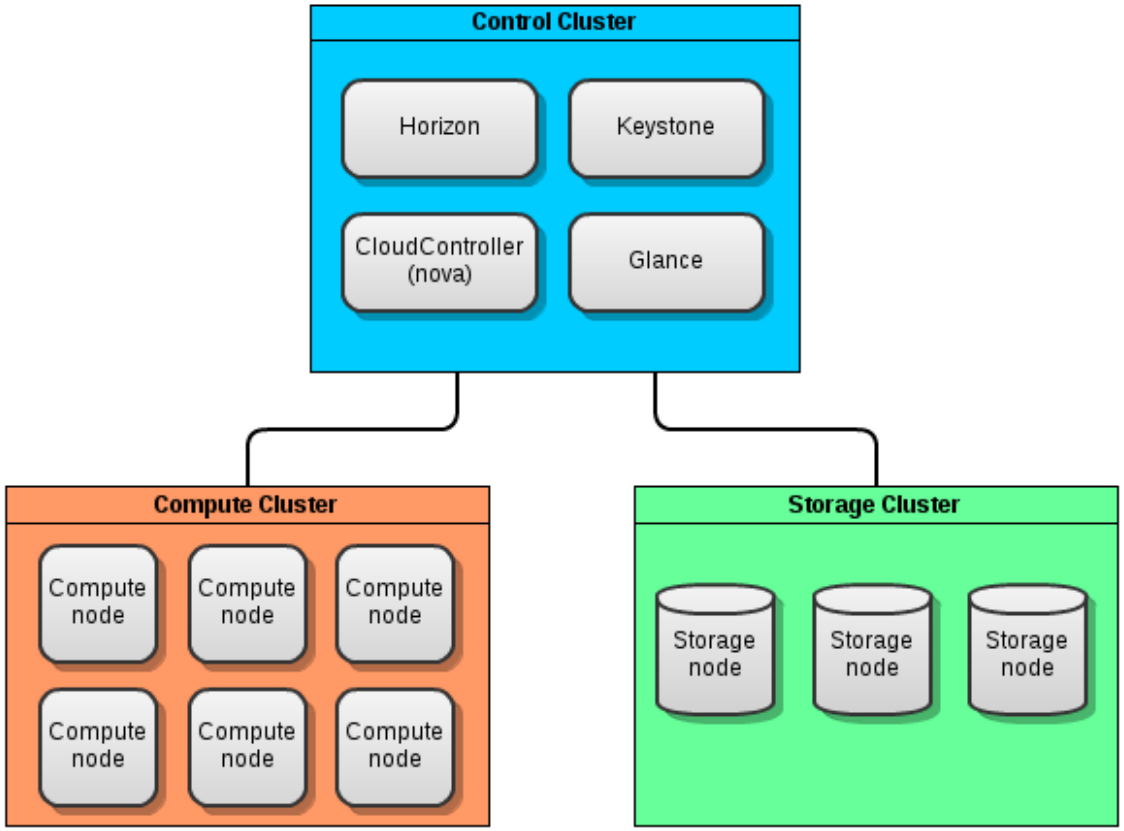
OpenStack Projects: Relationship



MIRANTIS

© Mirantis, Inc, 2012. All rights reserved.

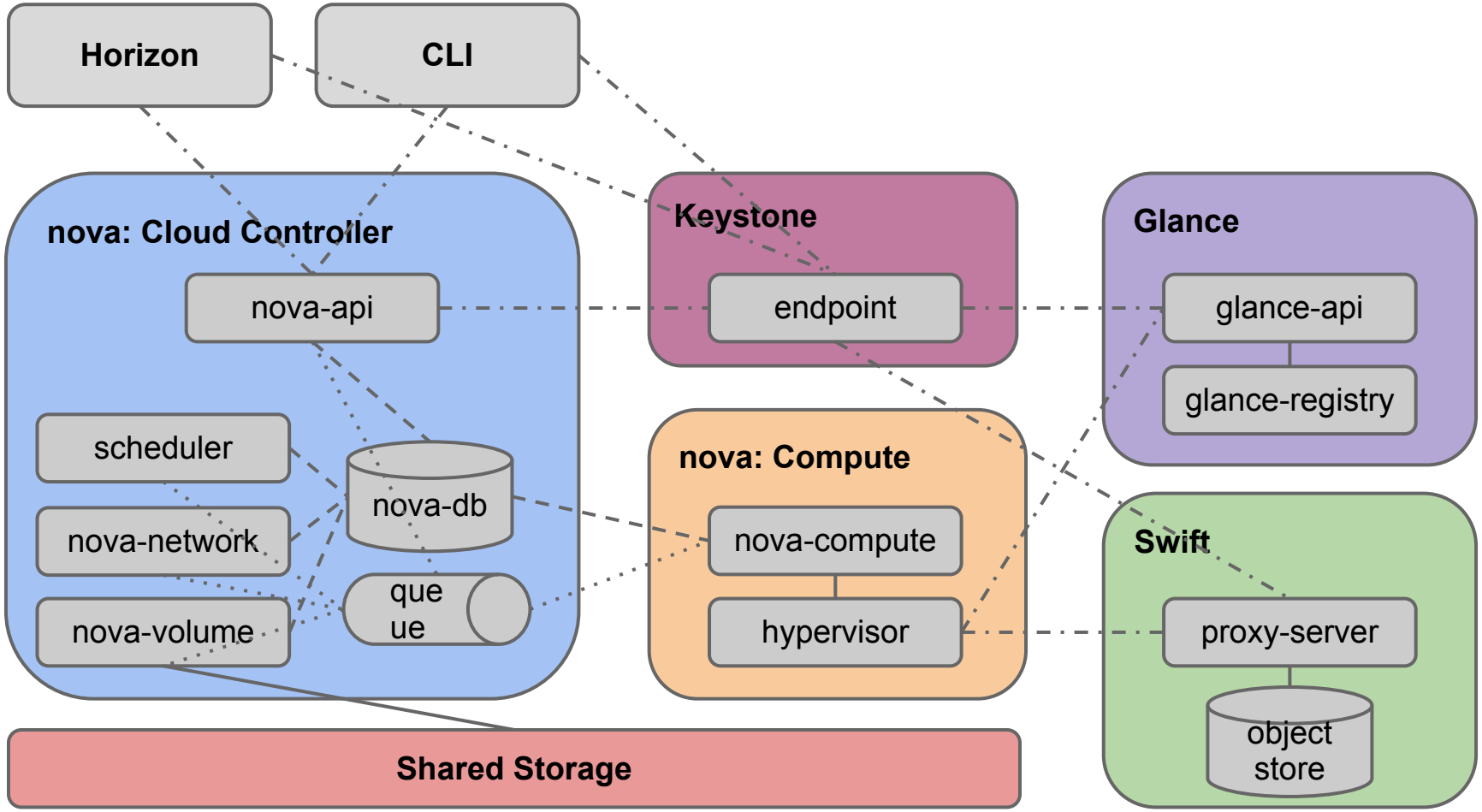
OpenStack: Deployment Topology



MIRANTIS

© Mirantis, Inc, 2012. All rights reserved.

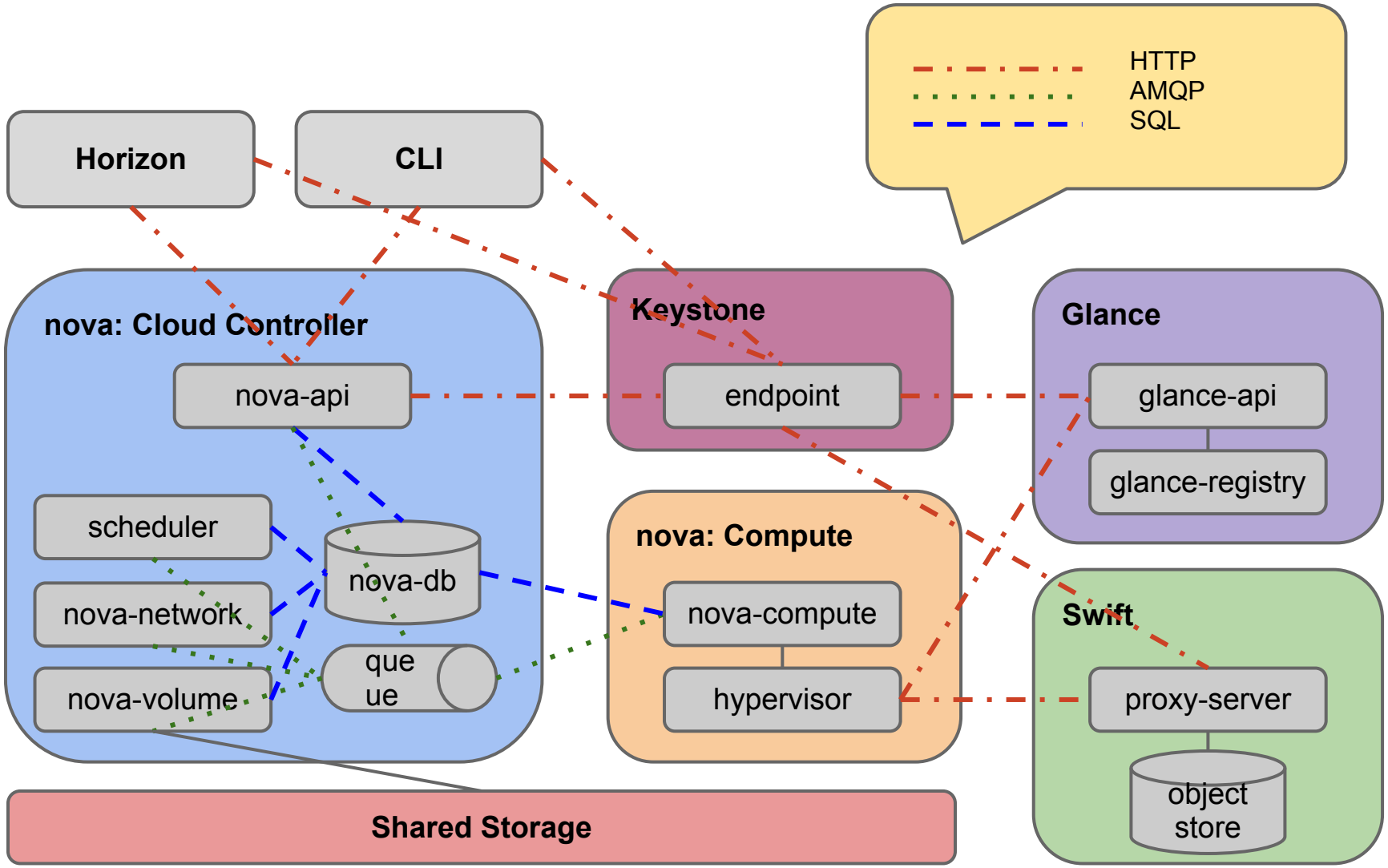
OpenStack Projects: Detailed View



MIRANTIS

© Mirantis, Inc, 2012. All rights reserved.

OpenStack Projects: Communication Types



MIRANTIS

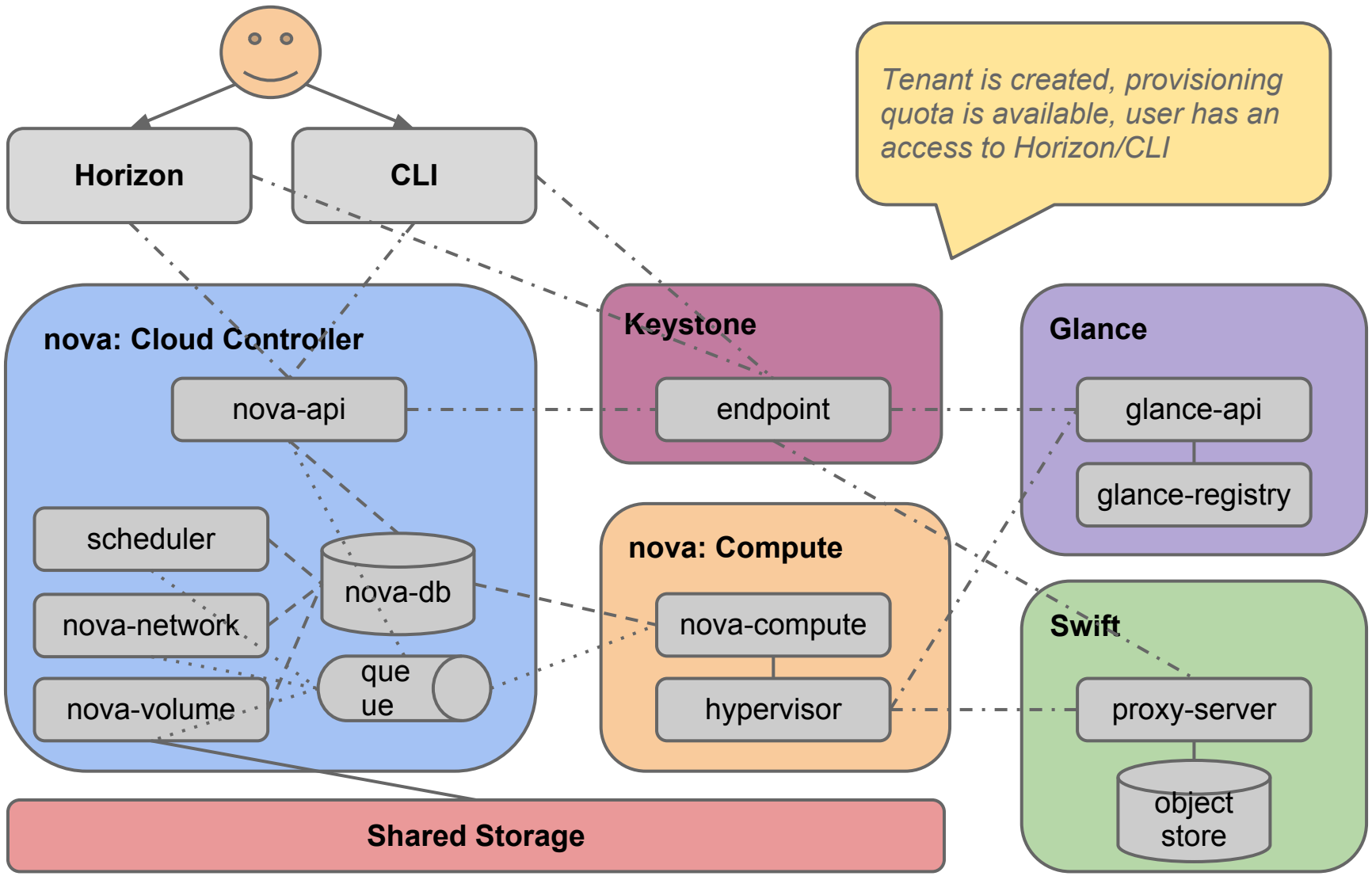
© Mirantis, Inc, 2012. All rights reserved.

Use Case: provision a VM

MIRANTIS

© Mirantis, Inc, 2012. All rights reserved.

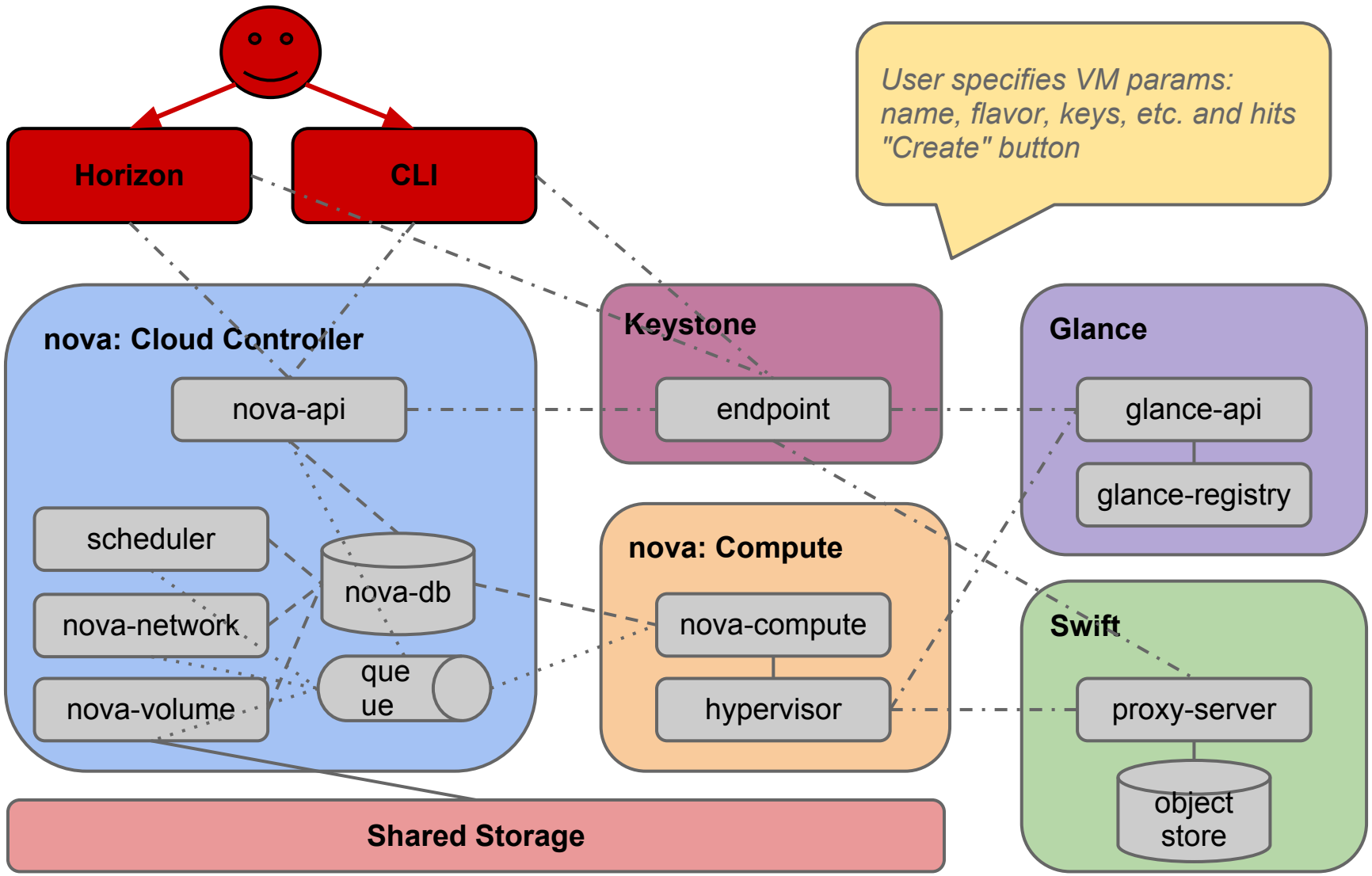
Initial State



MIRANTIS

© Mirantis, Inc, 2012. All rights reserved.

Step 1: Request VM Provisioning via UI/CLI



MIRANTIS

© Mirantis, Inc, 2012. All rights reserved.

Step 1: Request Provisioning via UI/CLI

- From UI
  a. Login to Horizon
  b. Specify params of VM in "Create VM" form
  c. Hit "Create" button

MIRANTIS

© Mirantis, Inc, 2012. All rights reserved.

Step 1: Request Provisioning via UI/CLI

- From UI
  a. Login to Horizon
  b. Specify params of VM in "Create VM" form
  c. Hit "Create" button

- Under the hood
  a. Form params are converted to POST data
  b. "Create" request initiate HTTP POST request to back-end
     - To **Keystone** if auth token is not cached
     - To **nova-api** if auth token hasn't expired yet

MIRANTIS

© Mirantis, Inc, 2012. All rights reserved.

Horizon

"The OpenStack Dashboard (Horizon) provides a baseline user interface for managing OpenStack services."

MIRANTIS

© Mirantis, Inc, 2012. All rights reserved.

Horizon notes

- "Stateless"

MIRANTIS

© Mirantis, Inc, 2012. All rights reserved.

Horizon notes

- "Stateless"
- Error handling is delegated to back-end

MIRANTIS

© Mirantis, Inc, 2012. All rights reserved.

Horizon notes

- "Stateless"
- Error handling is delegated to back-end
- Doesn't support all API functions

MIRANTIS

© Mirantis, Inc, 2012. All rights reserved.

Horizon notes

- "Stateless"
- Error handling is delegated to back-end
- Doesn't support all API functions
- Can use memcached or database to store sessions

MIRANTIS

© Mirantis, Inc, 2012. All rights reserved.

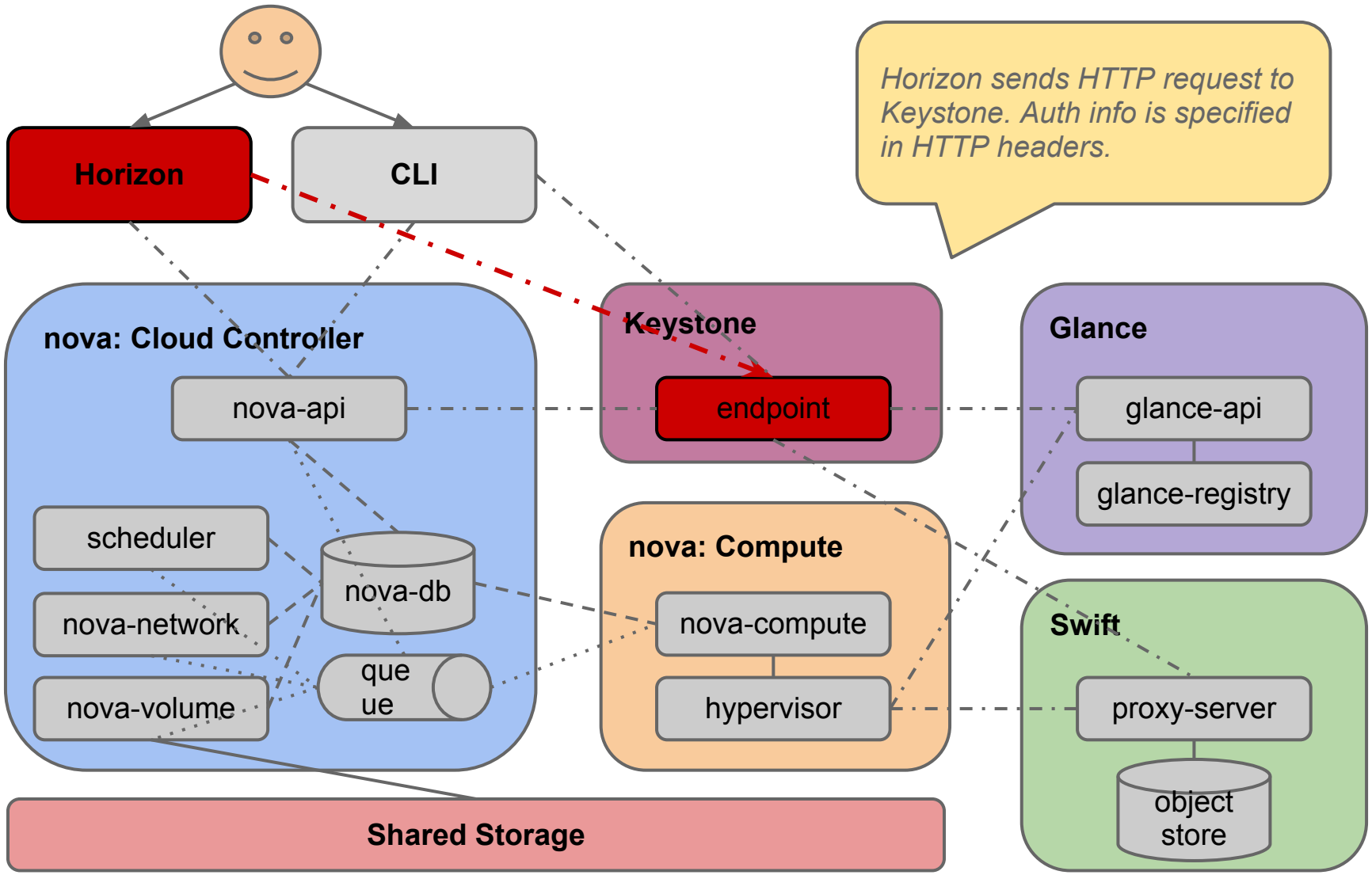
Horizon notes

- "Stateless"
- Error handling is delegated to back-end
- Doesn't support all API functions
- Can use memcached or database to store sessions
- Gets updated via nova-api polling

MIRANTIS

© Mirantis, Inc, 2012. All rights reserved.

Step 2: Validate Auth Data



MIRANTIS

© Mirantis, Inc, 2012. All rights reserved.

Step 2: Validate Auth Data

- From UI
  a. Horizon sends HTTP request to Keystone
  b. Keystone parses HTTP header info and verifies that
     - The credentials are valid
     - User-Tenant-Role mapping is valid
     - The requested action is available for this user

MIRANTIS

© Mirantis, Inc, 2012. All rights reserved.

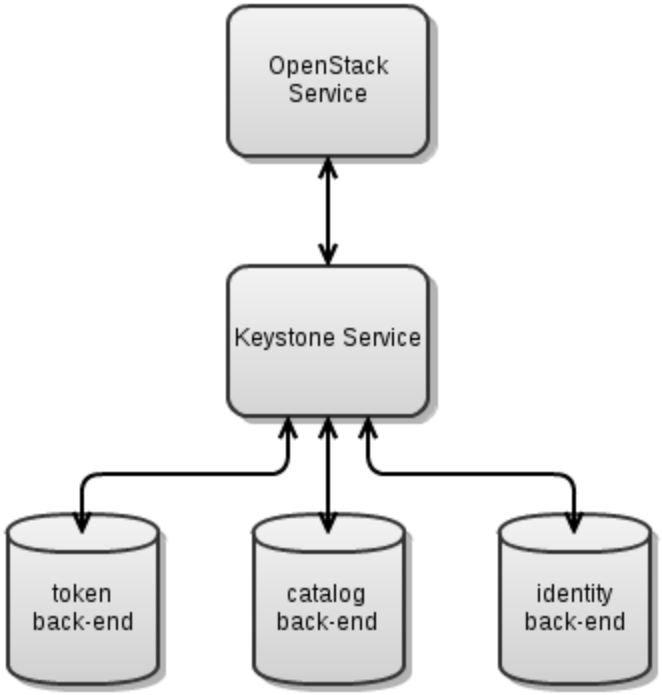
Keystone

"Keystone is an OpenStack project that provides Identity, Token, Catalog and Policy services for use specifically by projects in the OpenStack family."

MIRANTIS

© Mirantis, Inc, 2012. All rights reserved.

Keystone Architecture



MIRANTIS

© Mirantis, Inc, 2012. All rights reserved.

Keystone data model

- **User**: has account credentials, is associated with one or more tenants
- **Tenant**: unit of ownership in OpenStack, contains one or more users
- **Role**: a first-class piece of metadata associated with many user-tenant pairs.
- **Token**: identifying credential associated with a user or user and tenant
- **Extras**: bucket of key-value metadata associated with a user-tenant pair.
- **Rule**: describes a set of requirements for performing an action.

MIRANTIS

© Mirantis, Inc, 2012. All rights reserved.

Step 2: Validate Auth Data

- From UI
  a. Horizon sends HTTP request to Keystone
  b. Keystone parses HTTP header info and verifies that
     - The credentials are valid
     - User-Tenant-Role mapping is valid
     - **The requested action is available for this user**

MIRANTIS

© Mirantis, Inc, 2012. All rights reserved.

Step 2: Validate Auth Data

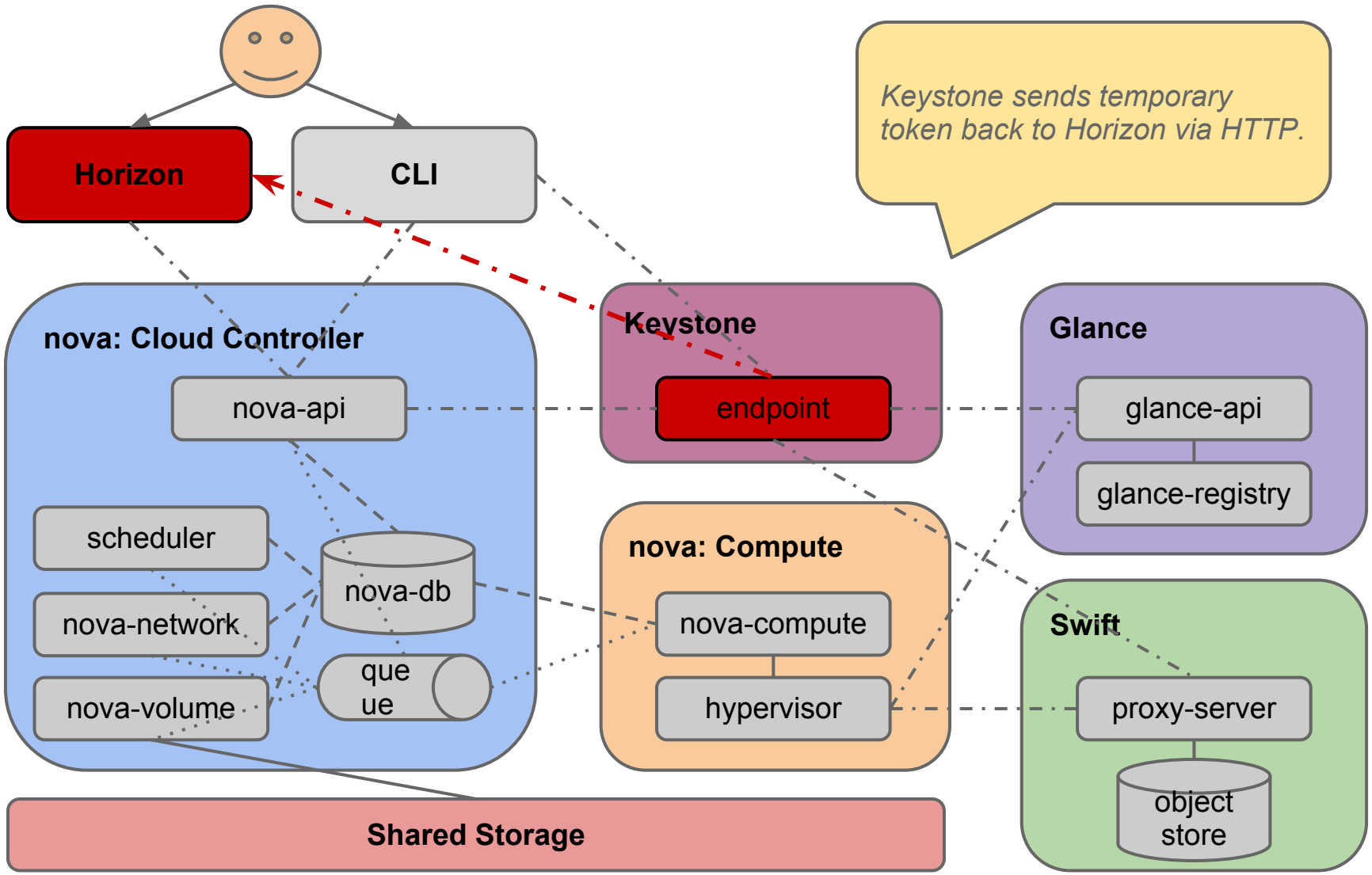
- From UI
  a. Horizon sends HTTP request to Keystone
  b. Keystone parses HTTP header info and verifies that
     - The credentials are valid
     - User-Tenant-Role mapping is valid
     - The requested action is available for this user

- If operation is allowed - send token back
- If operation is not allowed - send error code

MIRANTIS

© Mirantis, Inc, 2012. All rights reserved.

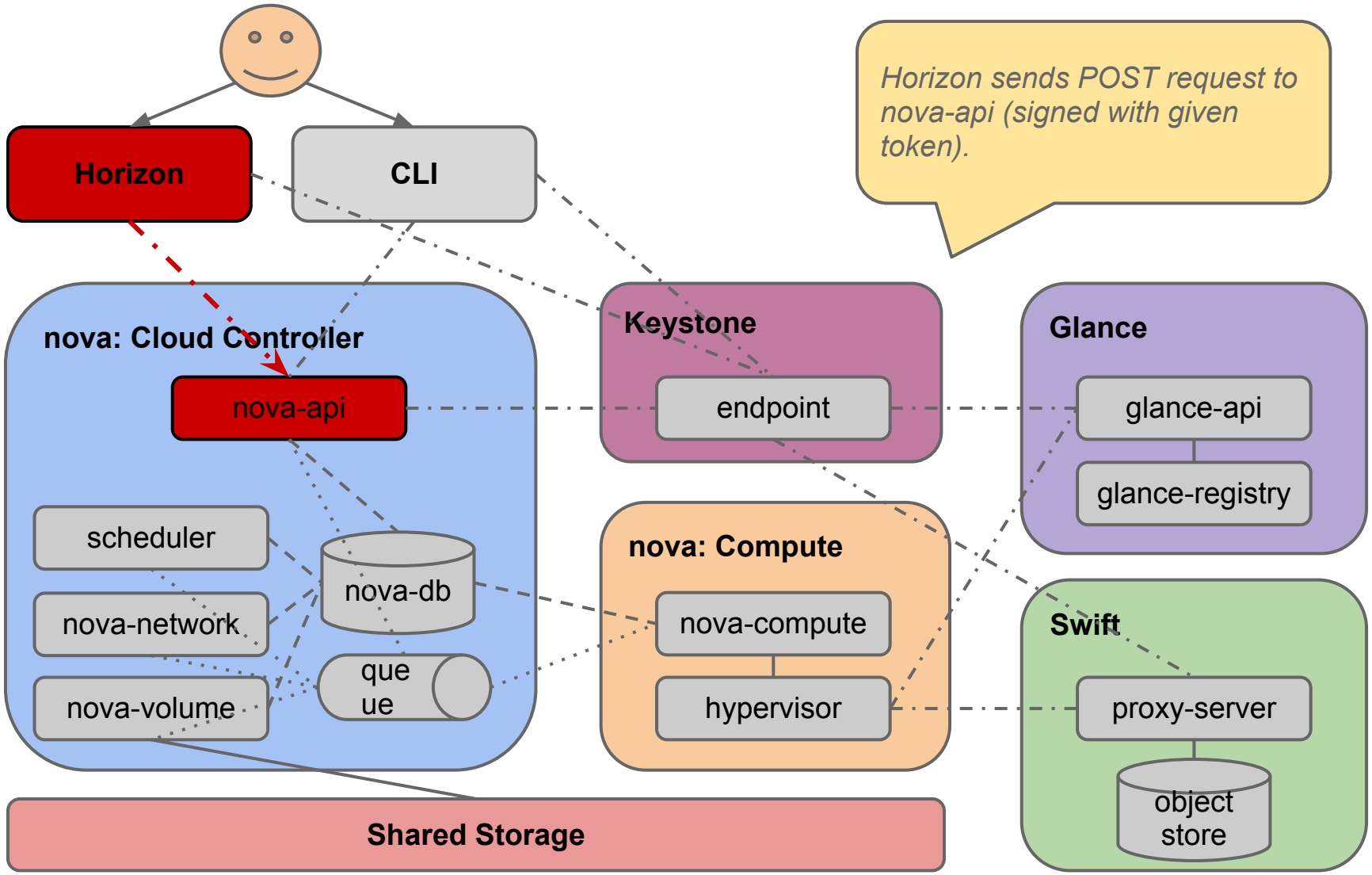
Step 2: Validate Auth Data



MIRANTIS

© Mirantis, Inc, 2012. All rights reserved.

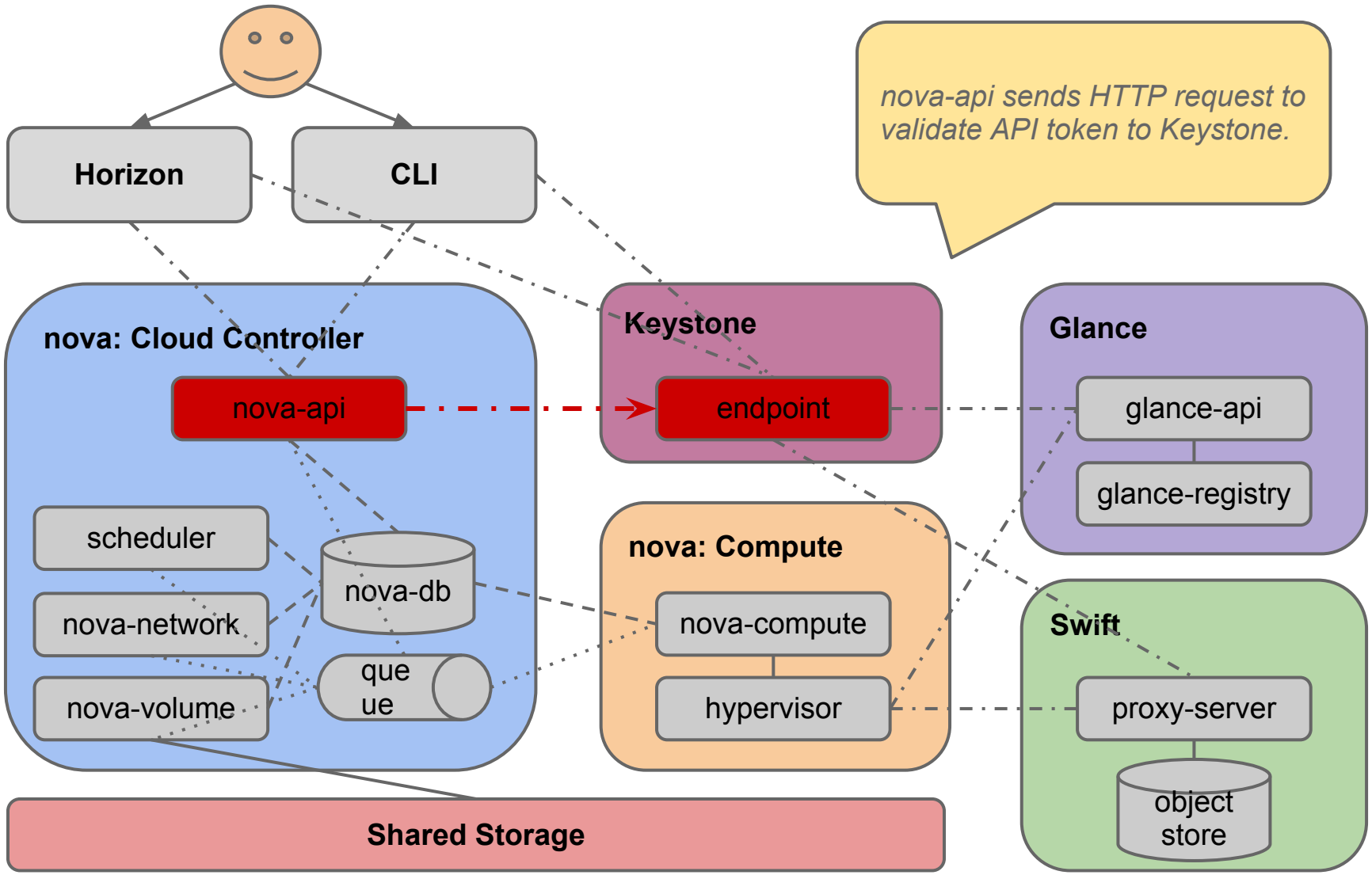
Step 3: Send API request to nova-api



MIRANTIS

© Mirantis, Inc, 2012. All rights reserved.

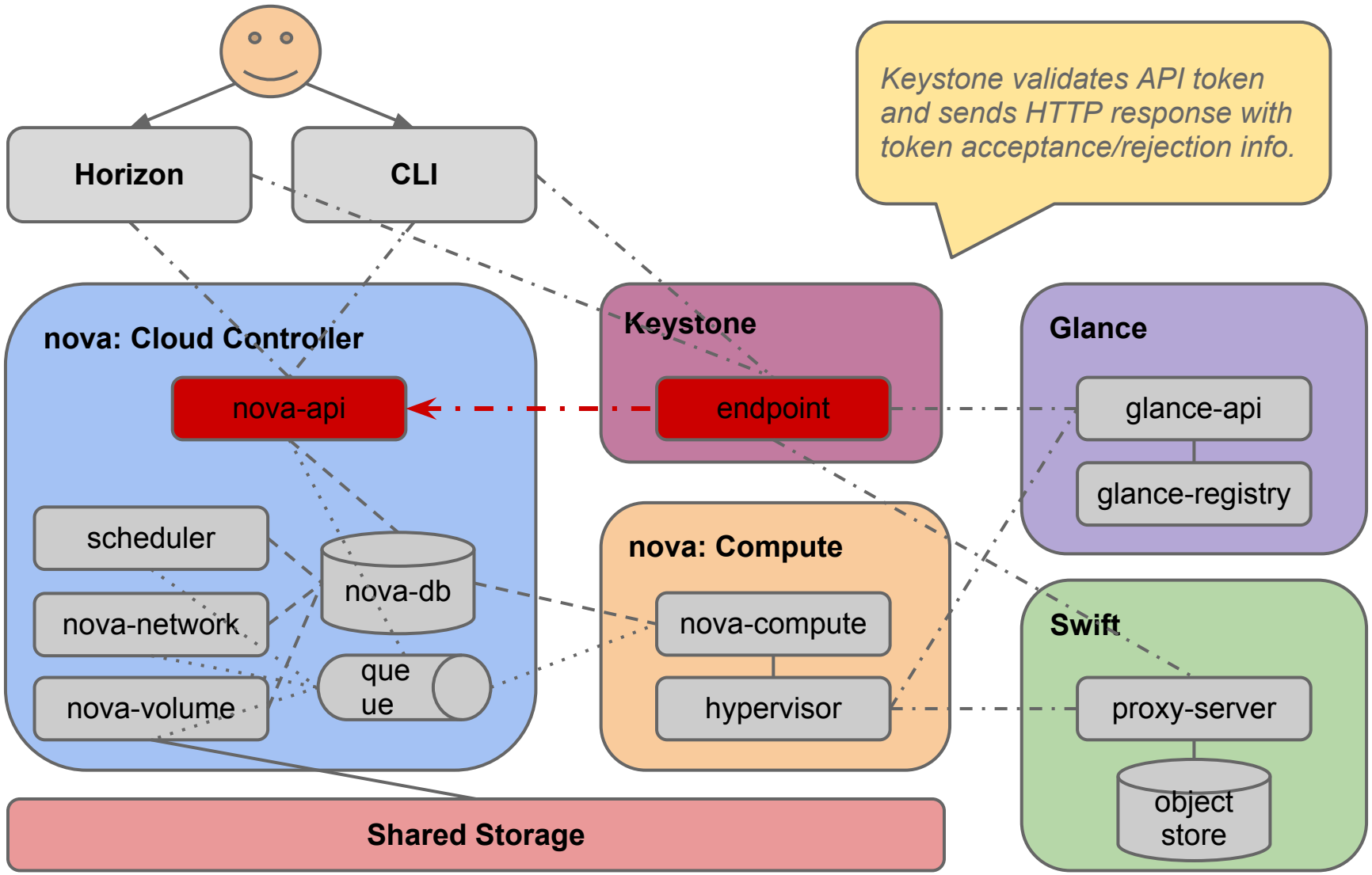
Step 4: Validate API Token



MIRANTIS

© Mirantis, Inc, 2012. All rights reserved.

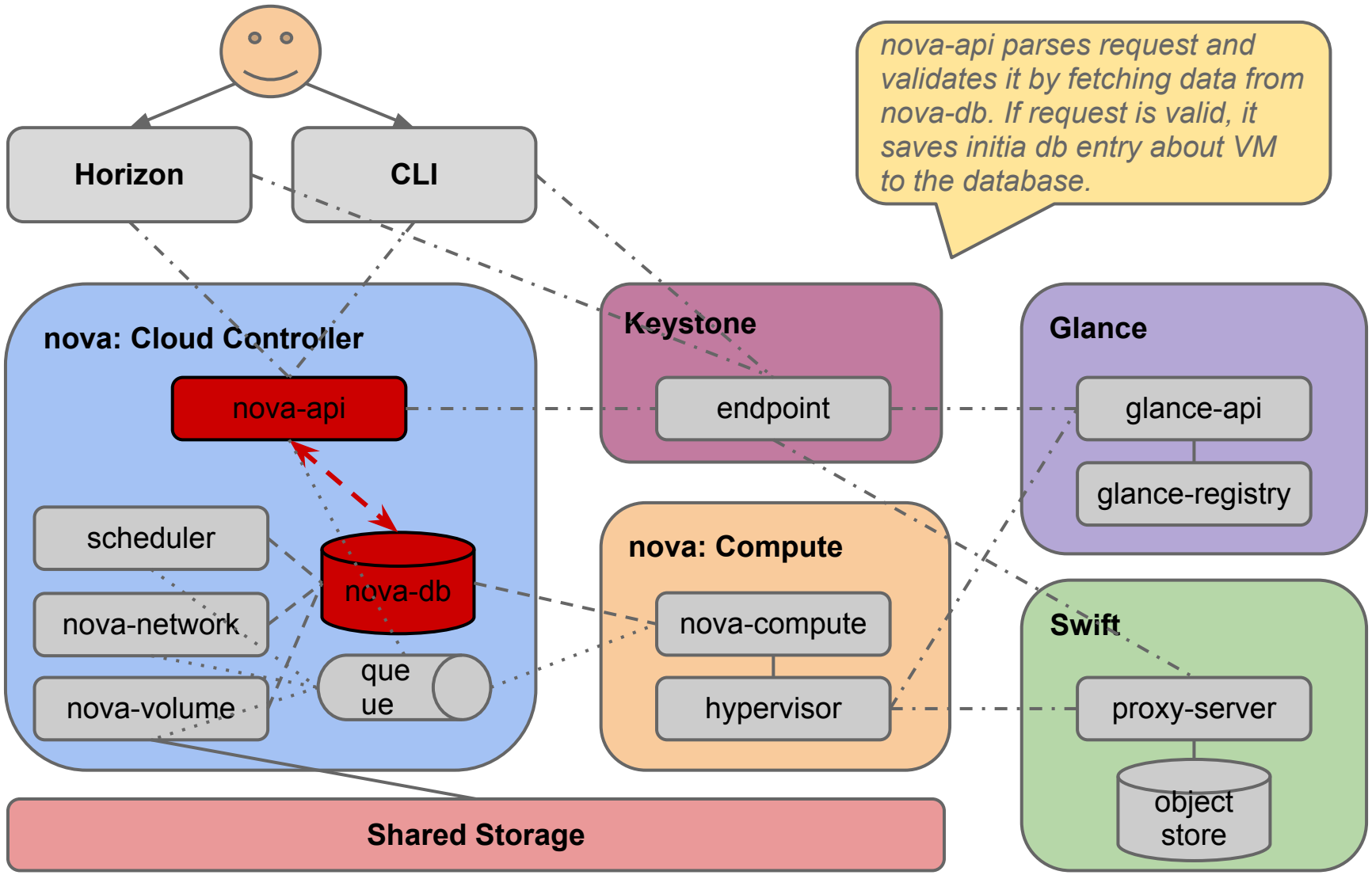
Step 4: Validate API Token



MIRANTIS

© Mirantis, Inc, 2012. All rights reserved.

Step 5: Process API request



MIRANTIS

© Mirantis, Inc, 2012. All rights reserved.

Step 5: Process API request

- Validate request params
  - Type errors are verified on code level
  - Cloud-related params are validated via DB requests
- If request cannot be processed then throw an exception
- If request can be processed
  - Save initial state to the database
  - Send message with next actions to MQ

MIRANTIS

© Mirantis, Inc, 2012. All rights reserved.

nova-api

"nova-api is a RESTful API web service which is used to interact with nova"

MIRANTIS

© Mirantis, Inc, 2012. All rights reserved.

nova-api characteristics

- Exposes REST API

MIRANTIS

© Mirantis, Inc, 2012. All rights reserved.

nova-api characteristics

- Exposes REST API
- Provides system for managing multiple APIs on different sub-domains
  - EC2-compatible - will be deprecated
  - OpenStack Compute API - all innovation happens here
- The only "allowed" way to interact with nova

MIRANTIS

© Mirantis, Inc, 2012. All rights reserved.

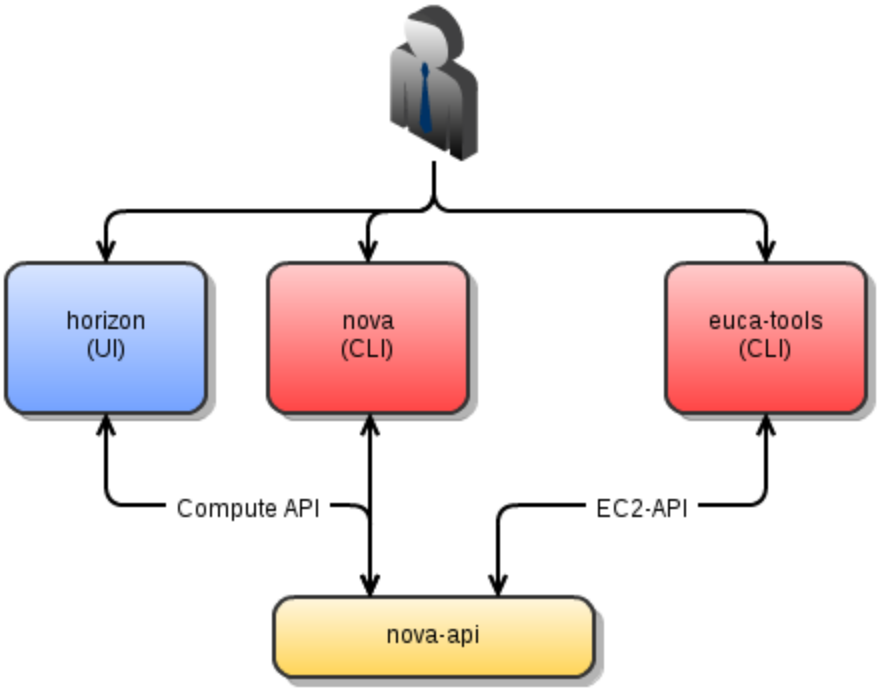
nova-api characteristics

- Exposes REST API
- Provides system for managing multiple APIs on different sub-domains
  - EC2-compatible - will be deprecated
  - OpenStack Compute API - all innovation happens here
- The only "allowed" way to interact with nova
- Stateless - HA-ready

MIRANTIS

© Mirantis, Inc, 2012. All rights reserved.

nova-api clients



MIRANTIS

© Mirantis, Inc, 2012. All rights reserved.

nova database

"nova database stores current state of all objects in compute cluster."

MIRANTIS

© Mirantis, Inc, 2012. All rights reserved.

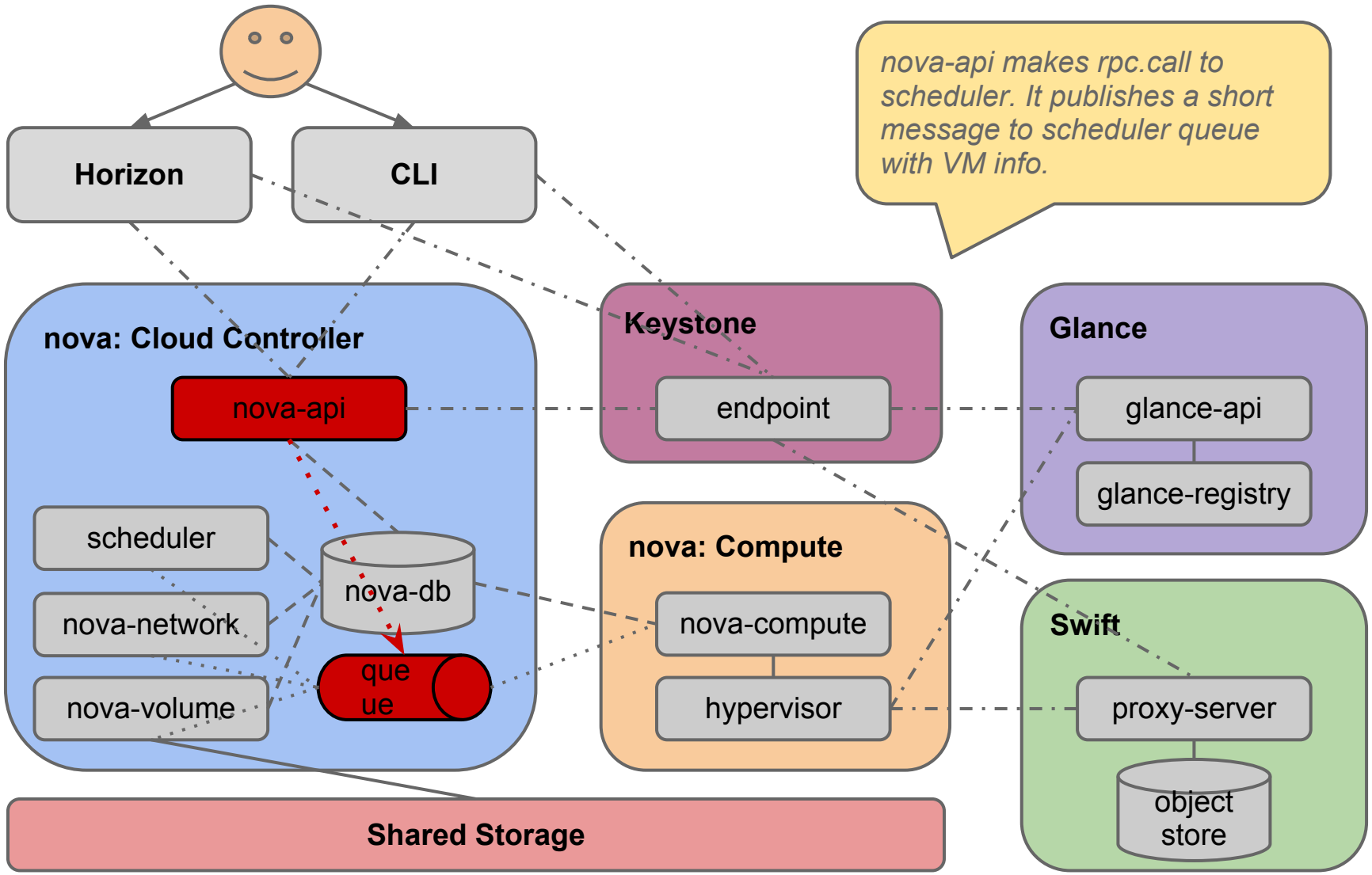
nova database

- Can be any relational database
- nova-api talks to DB via SQLAlchemy (python ORM)
- Most of the deployments are done with MySQL or PostgreSQL
- DB HA should be done via external tools (like MMM for MySQL)

MIRANTIS

© Mirantis, Inc, 2012. All rights reserved.

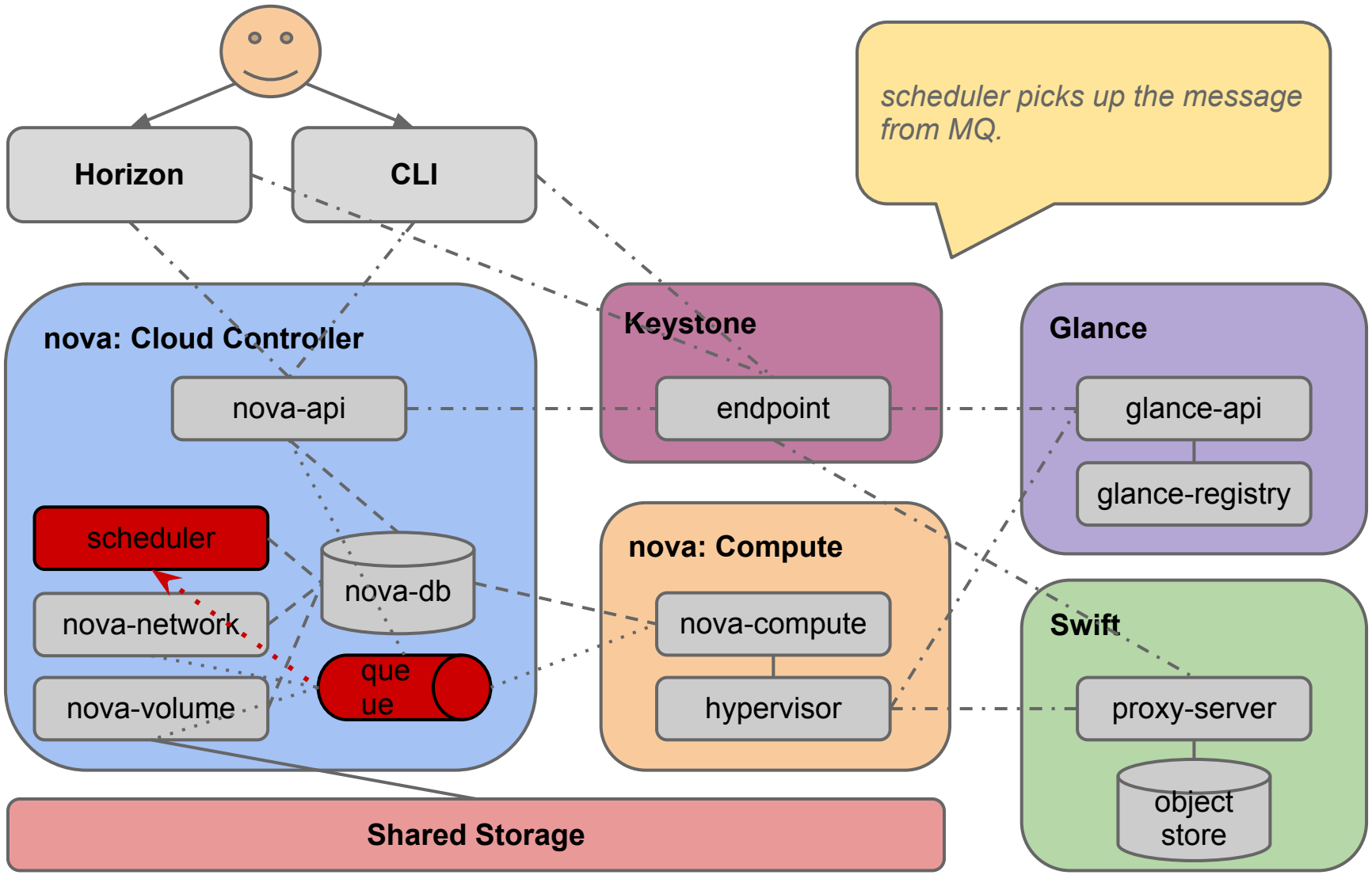
Step 6: Publish provisioning request to queue



MIRANTIS

© Mirantis, Inc, 2012. All rights reserved.

Step 6: Pick up provisioning request



MIRANTIS

© Mirantis, Inc, 2012. All rights reserved.

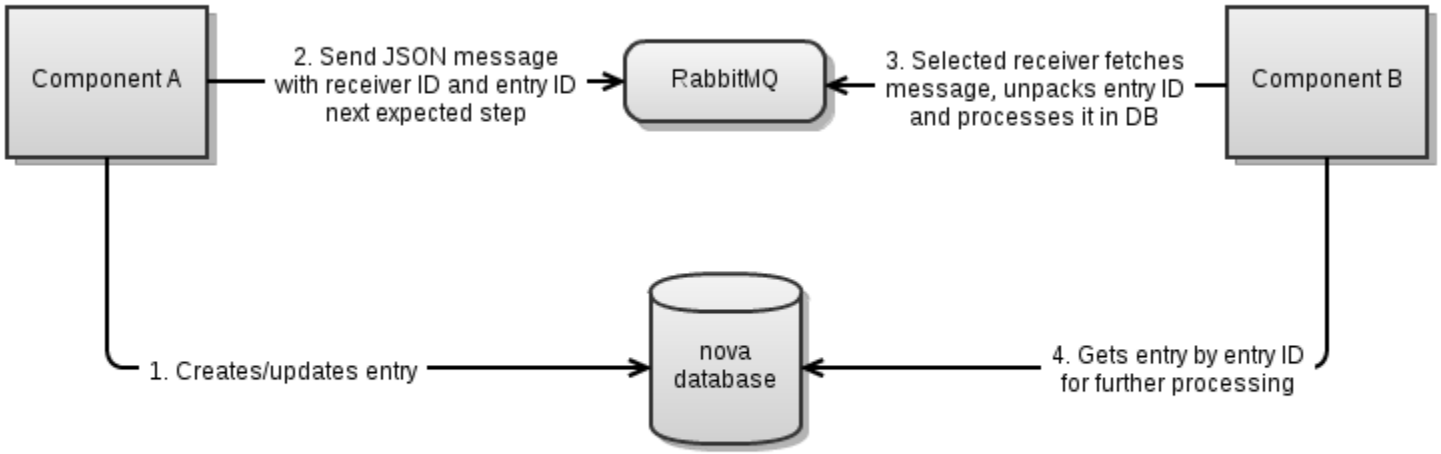
Message queue

"Message queue is a unified way for collaboration between nova components."

MIRANTIS

© Mirantis, Inc, 2012. All rights reserved.

OpenStack messaging



2 modes:

- rpc.cast - don't wait for result
- rpc.call - wait for result (when there is something to return)

MIRANTIS

© Mirantis, Inc, 2012. All rights reserved.

Messaging: notes

- OpenStack uses multiple queues within single RabbitMQ instance
  - 1 Queue per service
  - 1 "fanout" queue

MIRANTIS

© Mirantis, Inc, 2012. All rights reserved.

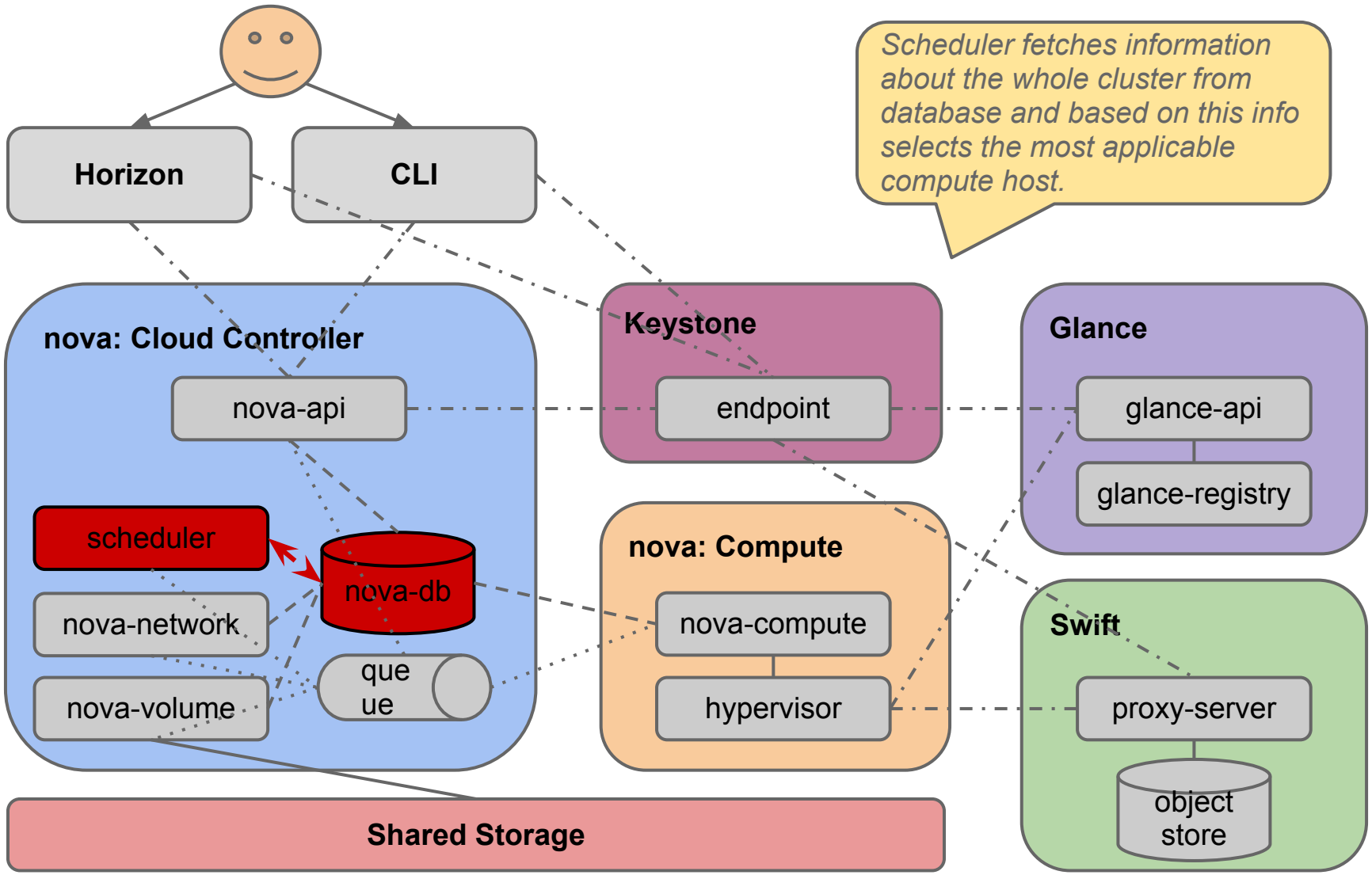
Messaging: notes

- OpenStack uses multiple queues within single RabbitMQ instance
  - 1 Queue per service
  - 1 "fanout" queue
- OpenStack messages traffic is not intensive

MIRANTIS

© Mirantis, Inc, 2012. All rights reserved.

Step 7: Schedule provisioning



MIRANTIS

© Mirantis, Inc, 2012. All rights reserved.

nova-scheduler

"nova-scheduler is a daemon, which determines, on which compute host the request should run."

MIRANTIS

© Mirantis, Inc, 2012. All rights reserved.

nova-scheduler: available schedulers

Scheduler	Description
Chance	Picks a host that is up at random
Simple	Picks a host that is up and has the fewest running instances
Filter	Picks the best-suited host which satisfies selected filter
Multi	A scheduler that holds multiple sub-schedulers

MIRANTIS

© Mirantis, Inc, 2012. All rights reserved.

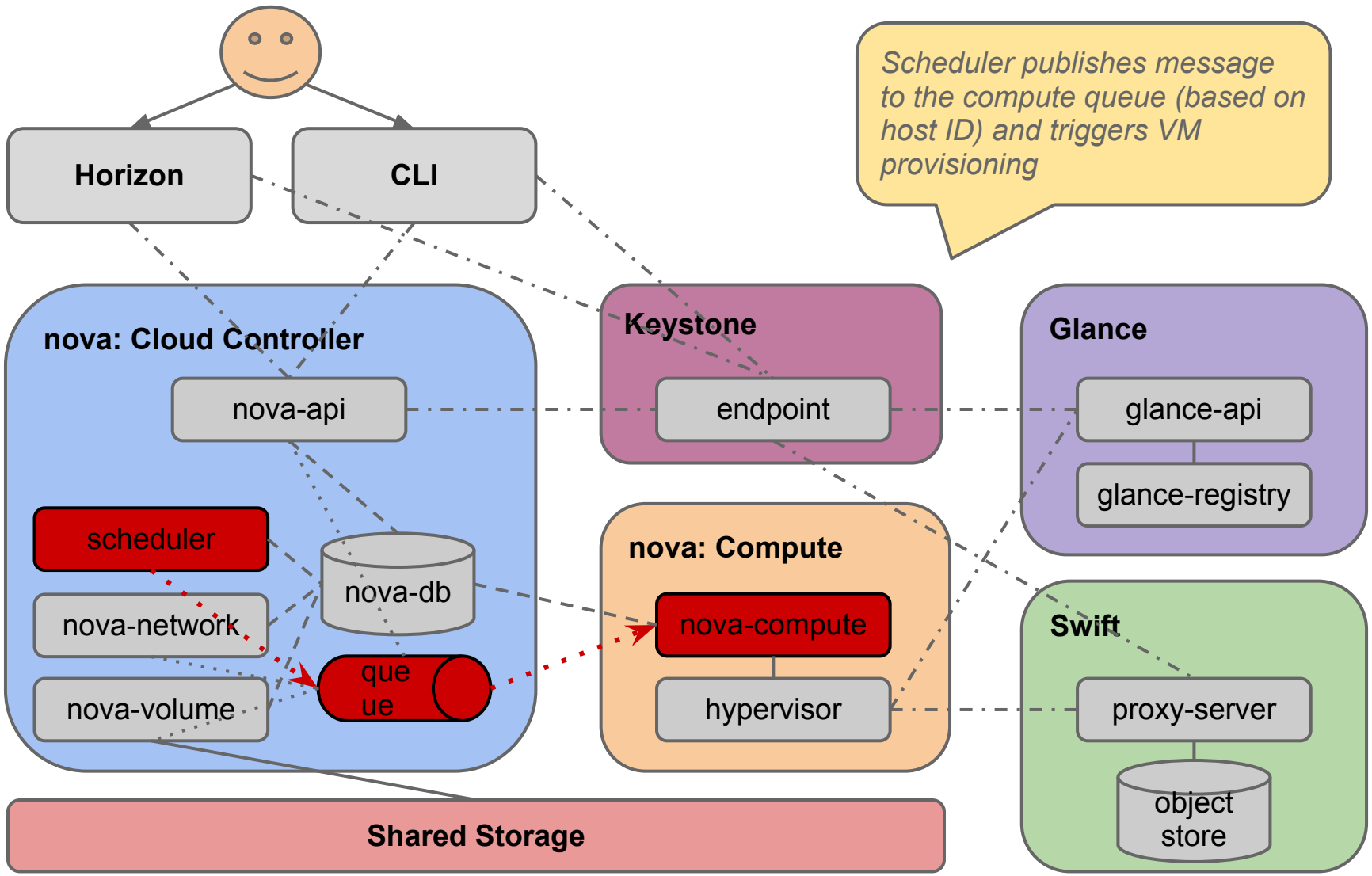
nova-scheduler: filters

Filter	Description
affinity	Same host or different host
availability zone	Least cost inside selected availability zone
core	Least CPU core utilization
ram	Only return hosts with sufficient RAM
json	Allows simple JSON based grammar. Can be used to build custom schedulers.

MIRANTIS

© Mirantis, Inc, 2012. All rights reserved.

Step 8: Start VM provisioning on compute node



MIRANTIS

© Mirantis, Inc, 2012. All rights reserved.

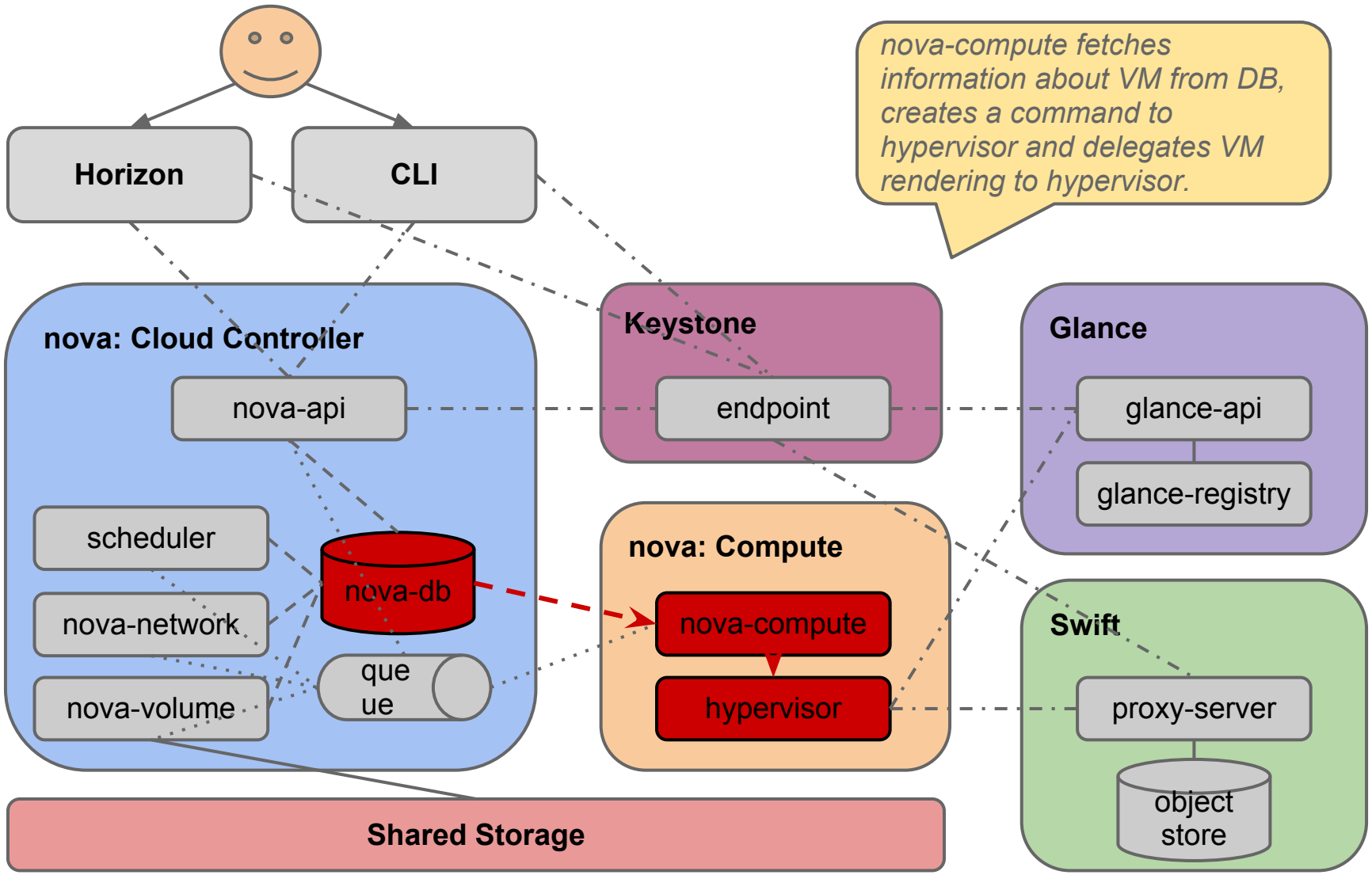
VM Provisioning Algorithm

1. nova-compute gets message from MQ
2. nova-compute fetches required instance information from DB
3. nova-compute starts instance via hypervisor driver
   a. driver requests image from glance by image_id
   b. if image is available, driver downloads image (depends on implementation)
   c. when download is complete, driver renders image
4. nova-compute calls nova-network to configure networking for created instance

MIRANTIS

© Mirantis, Inc, 2012. All rights reserved.

Step 9: Start VM rendering via hypervisor



MIRANTIS

© Mirantis, Inc, 2012. All rights reserved.

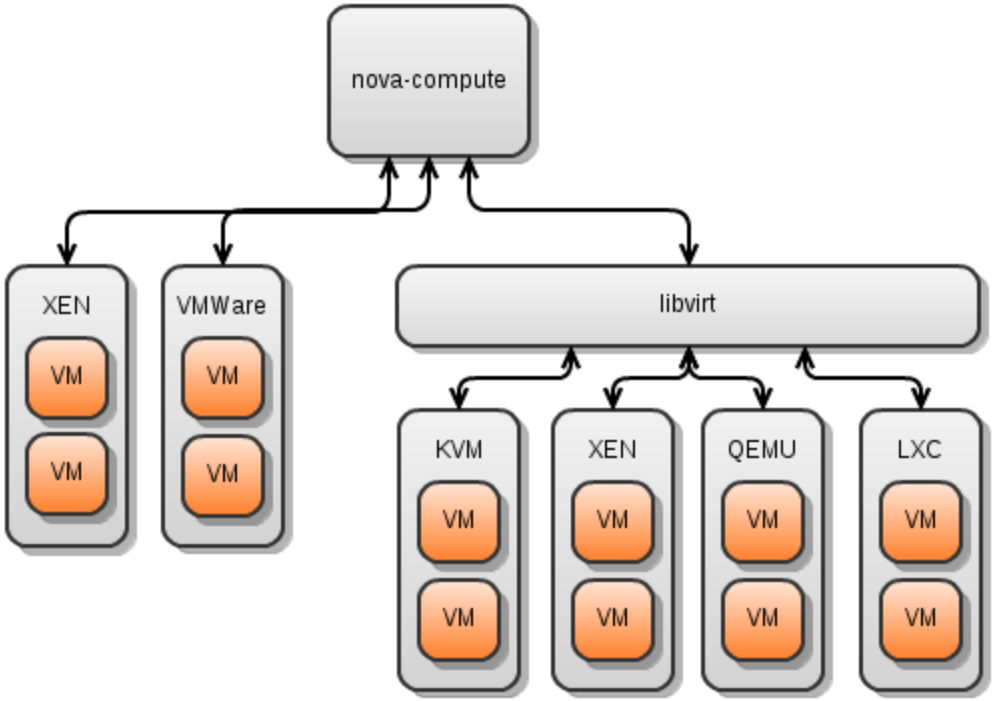
nova-compute

"nova-compute is a worker daemon, which primarily creates and terminates VMs via hypervisor API."

MIRANTIS

© Mirantis, Inc, 2012. All rights reserved.

nova-compute



MIRANTIS

© Mirantis, Inc, 2012. All rights reserved.

nova-compute: drivers

- Functionality is not 100% similar
- Exact "run_instance" flow depends on driver implementation
- Most of the features are tested on KVM

MIRANTIS

© Mirantis, Inc, 2012. All rights reserved.

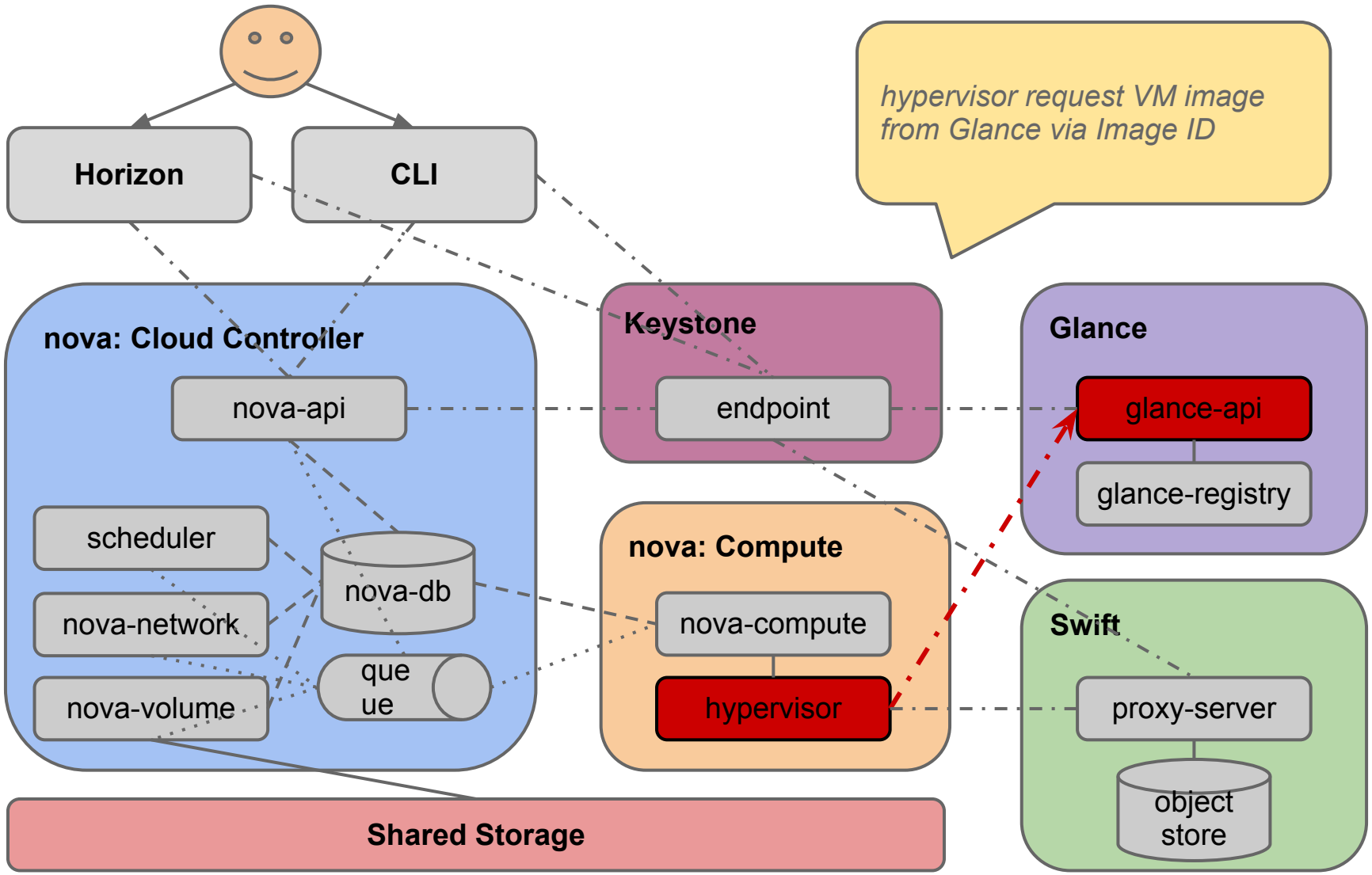
nova-compute config

- **`--libvirt_type`**
  Hypervisor being used. In this deployment 'kvm' is specified.
- **`--libvirt_uri`**
  URI to use for connection to hypervisor. In this deployment 'qemu+tcp:///system' is specified.
- **`--sql_connection`**
  Database connection string in SQLAlchemy format. This is used for connecting to state database.
- **`--rabbit_host`**
  IP address for RabbitMQ host. Non-standard port also can be specified
- **`--glance_host`**
  IP address and port of Glance Image Service host. This is needed for streaming virtual boot images.
- **`--glance_api_server`**
  IP address and port of Glance API server. This is needed for getting virtual boot images meta-data.

MIRANTIS

© Mirantis, Inc, 2012. All rights reserved.

Step 10: Request VM Image from Glance



MIRANTIS

© Mirantis, Inc, 2012. All rights reserved.

Glance

"The Glance project provides services for discovering, registering, and retrieving virtual machine images."

MIRANTIS

© Mirantis, Inc, 2012. All rights reserved.

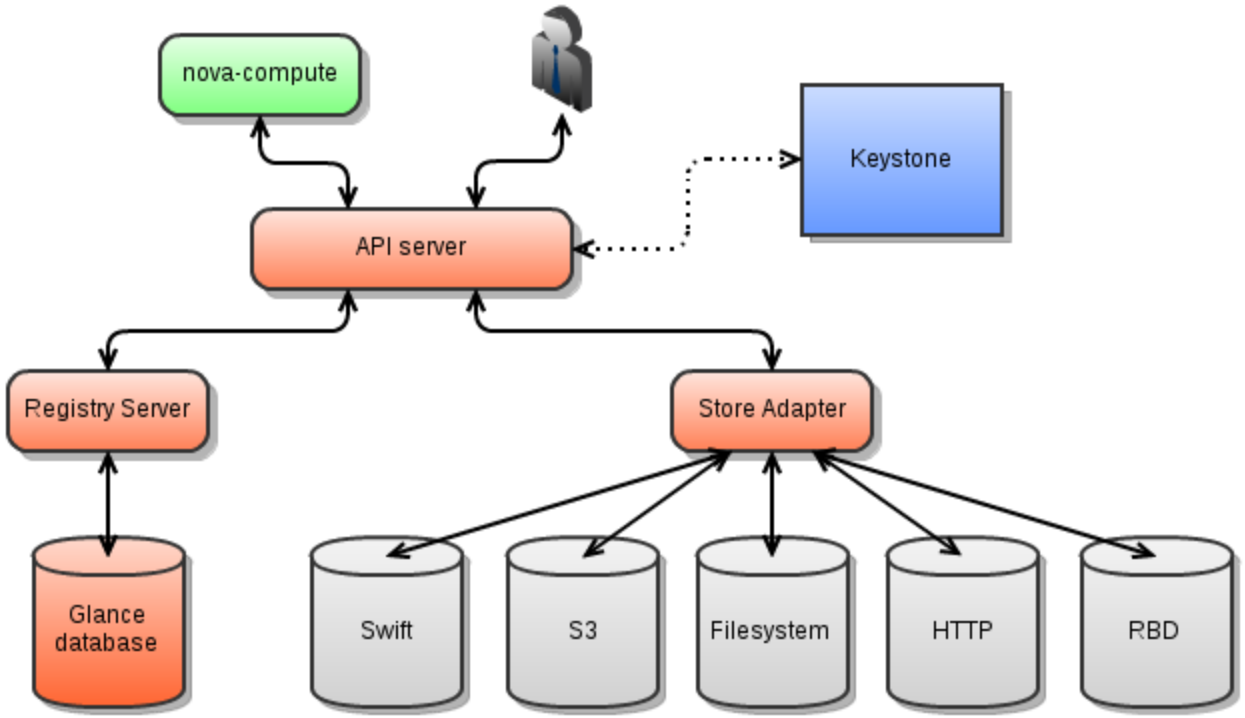
Glance summary

- Images-as-a-Service
- Can use multiple back-ends for image storage
- Supports multiple image formats

MIRANTIS

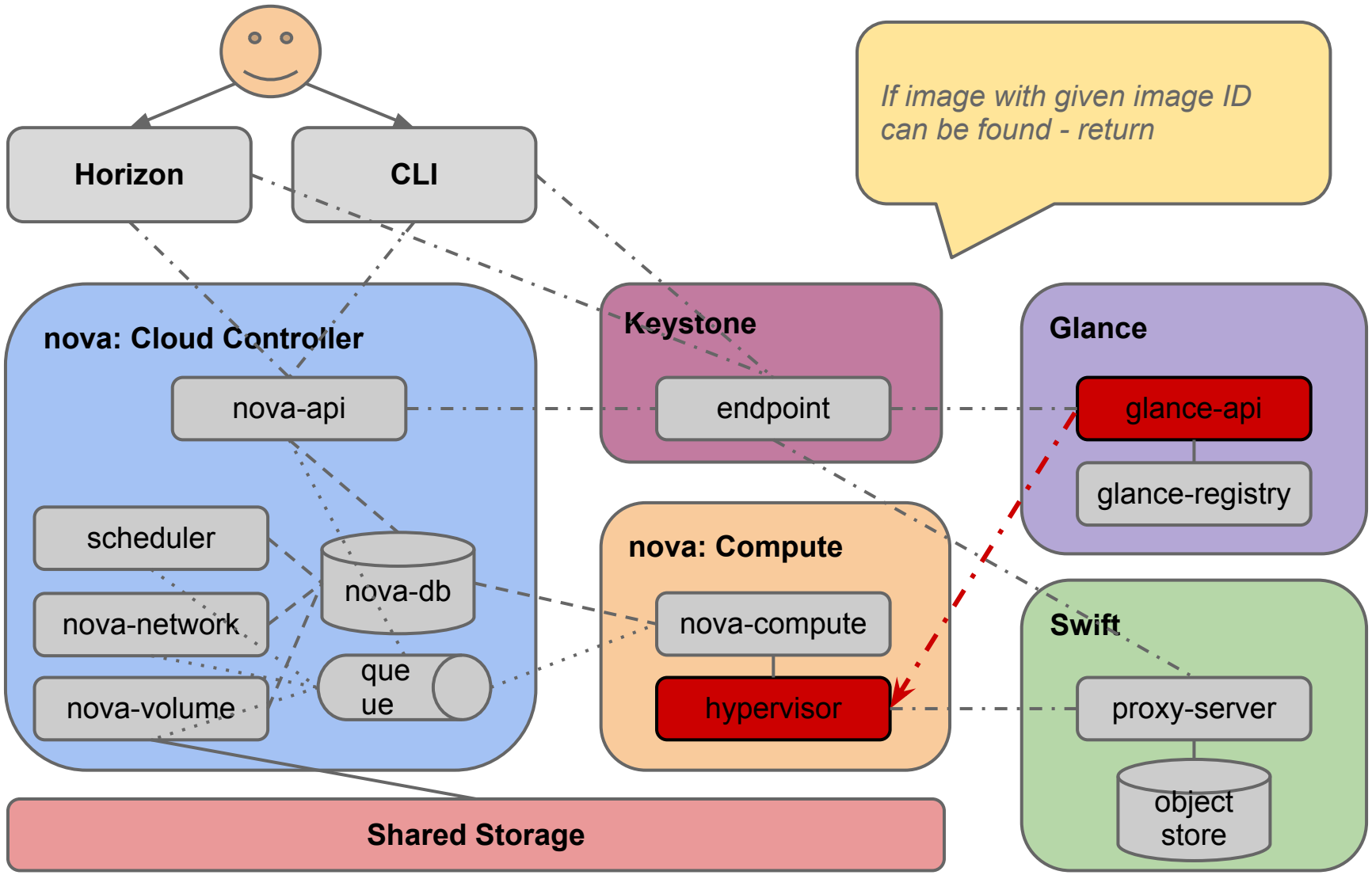
© Mirantis, Inc, 2012. All rights reserved.

Glance architecture



© Mirantis, Inc, 2012. All rights reserved.

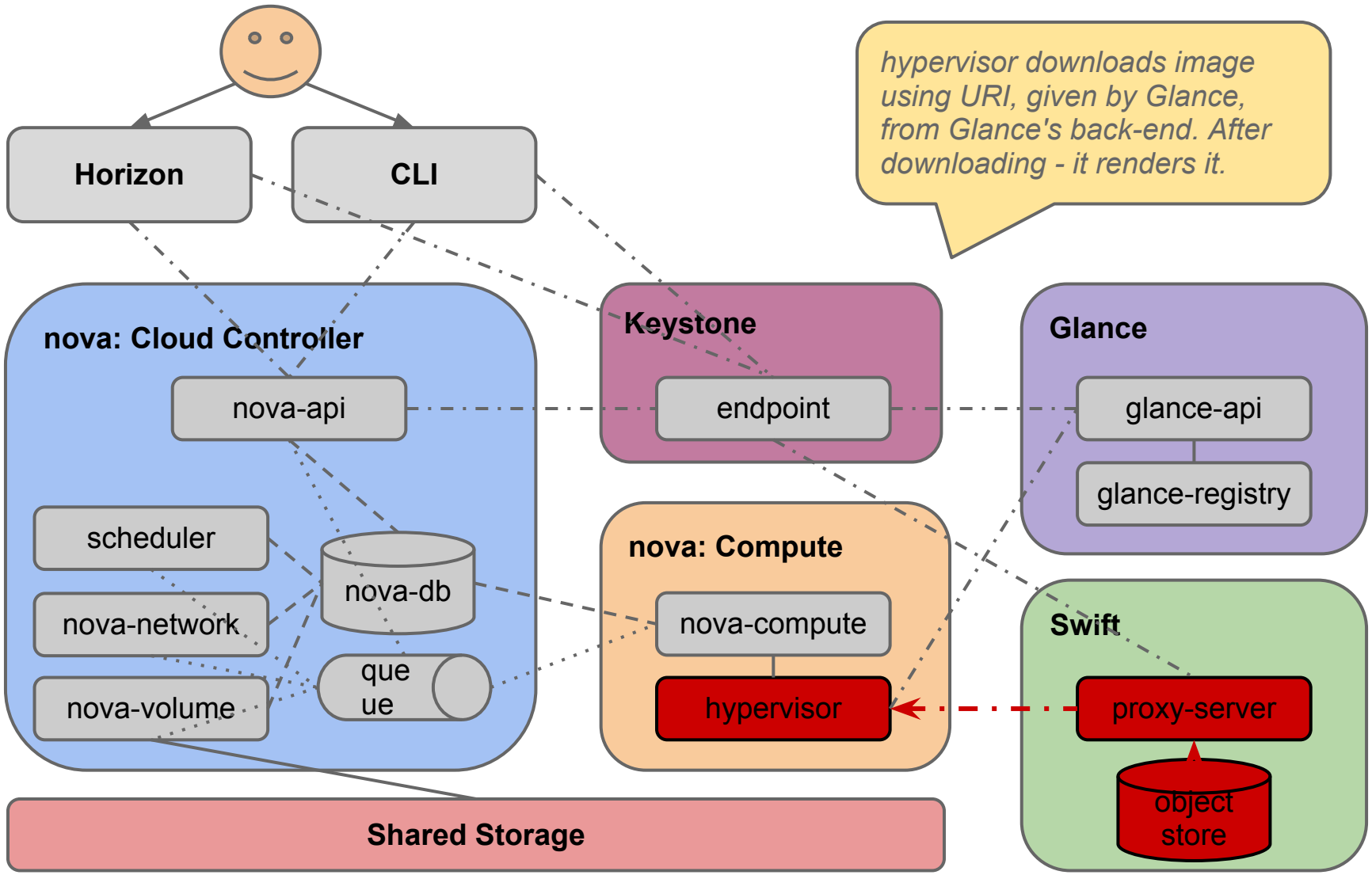
Step 11: Get Image URI from Glance



MIRANTIS

© Mirantis, Inc, 2012. All rights reserved.

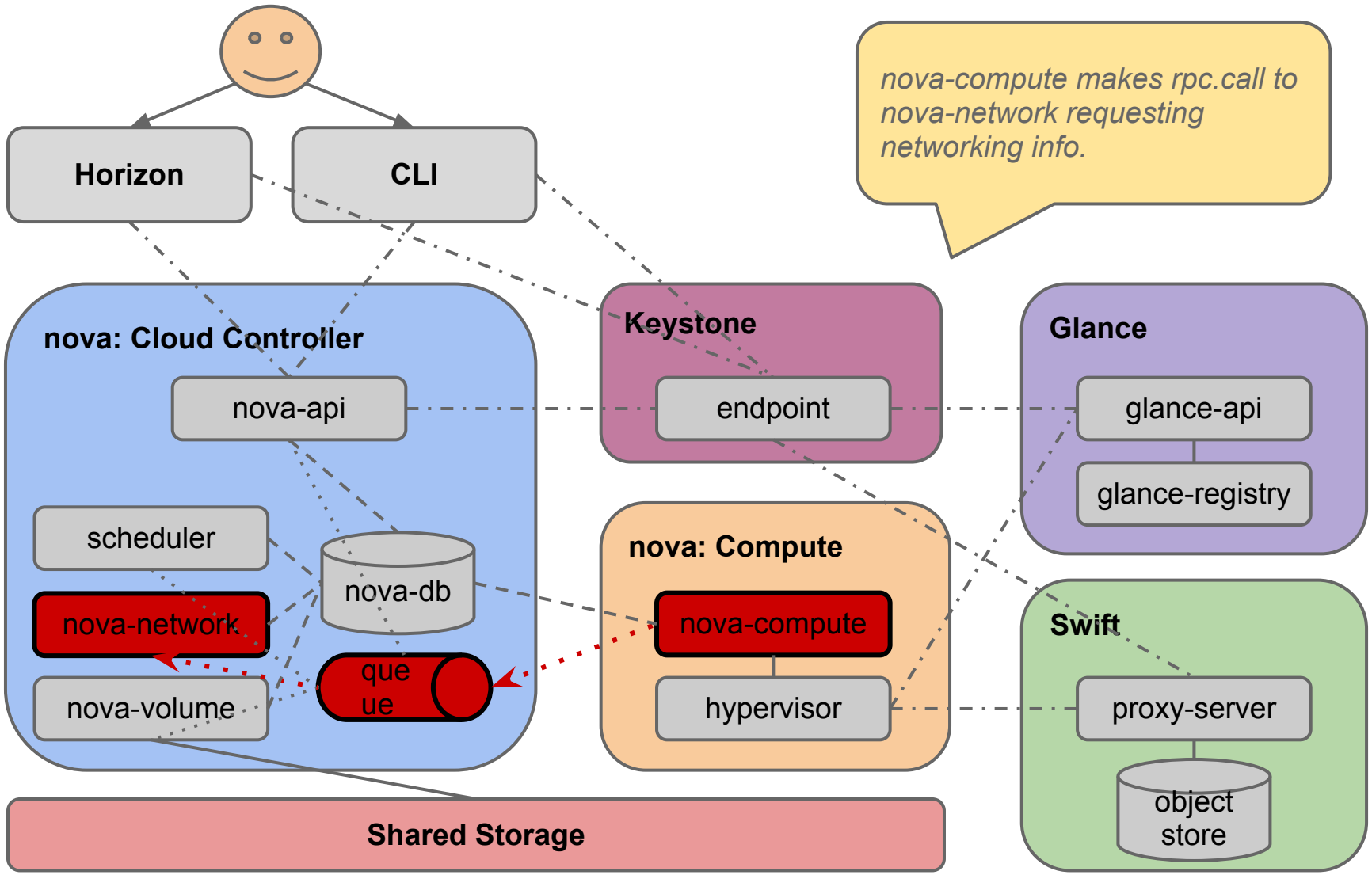
Step 12: Download image from Swift



MIRANTIS

© Mirantis, Inc, 2012. All rights reserved.

Step 13: Configure network



MIRANTIS

© Mirantis, Inc, 2012. All rights reserved.

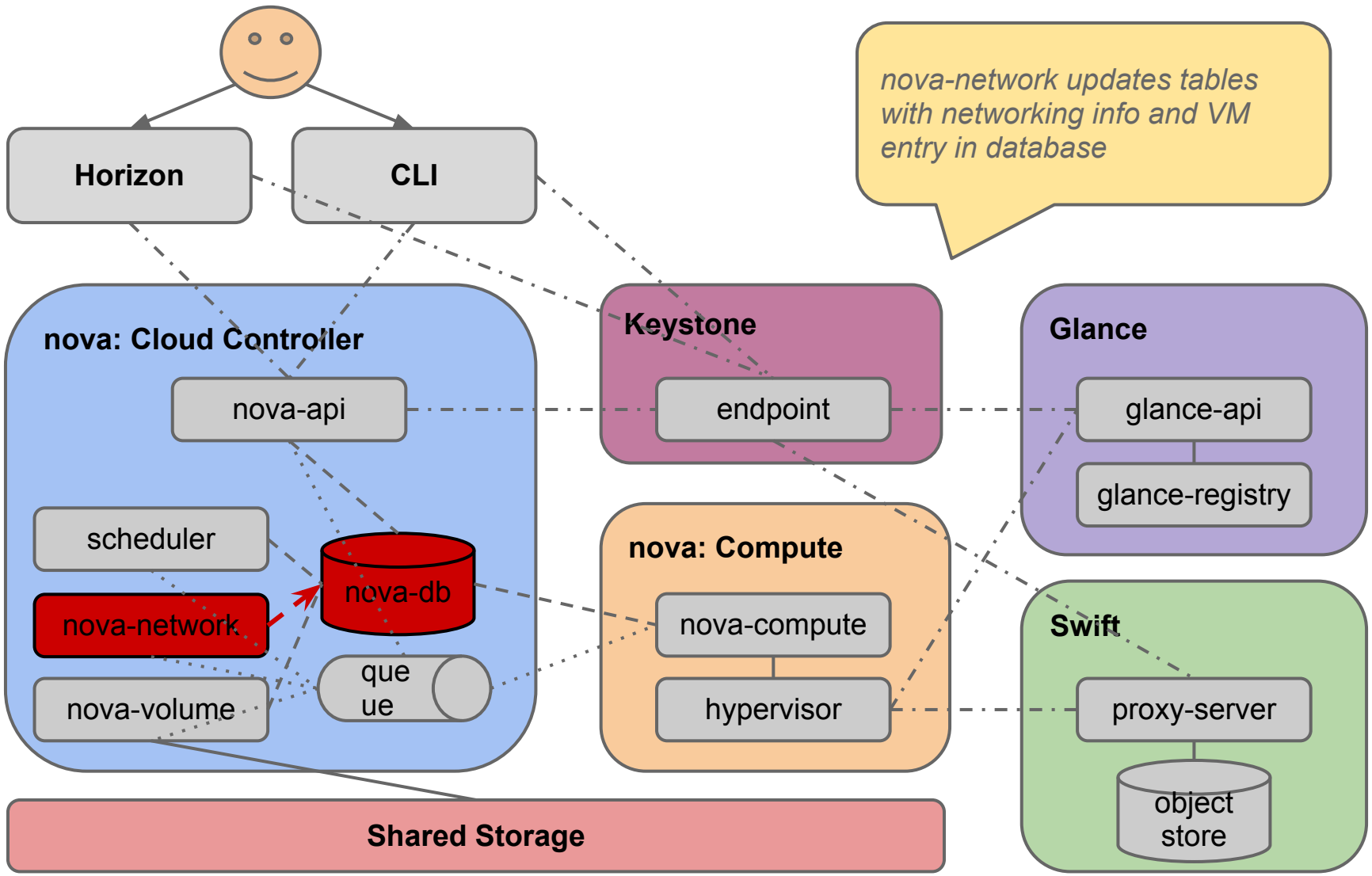
Network configuration flow

1. Allocate MAC addresses
2. Allocate IPs (for each network)
3. Associate IPs with VMs (DB)
4. Setup network on host
   a. Update DHCP config
   b. Initialize gateway
   c. VPN configuration (optional)
5. Update networking info in DB

MIRANTIS

© Mirantis, Inc, 2012. All rights reserved.

Step 14: allocate and associate network



MIRANTIS

© Mirantis, Inc, 2012. All rights reserved.

nova-network

"nova-network is a worker daemon which performs tasks to manipulate network via external commands."

MIRANTIS

© Mirantis, Inc, 2012. All rights reserved.

Floating IPs

- Shared pool of public IP addresses
- Each user gets a quota of how many IPs to use
- Managed by admin

MIRANTIS

© Mirantis, Inc, 2012. All rights reserved.

Assigning Floating IPs

OpenStack Admin

- Dedicate floating IPs to cluster

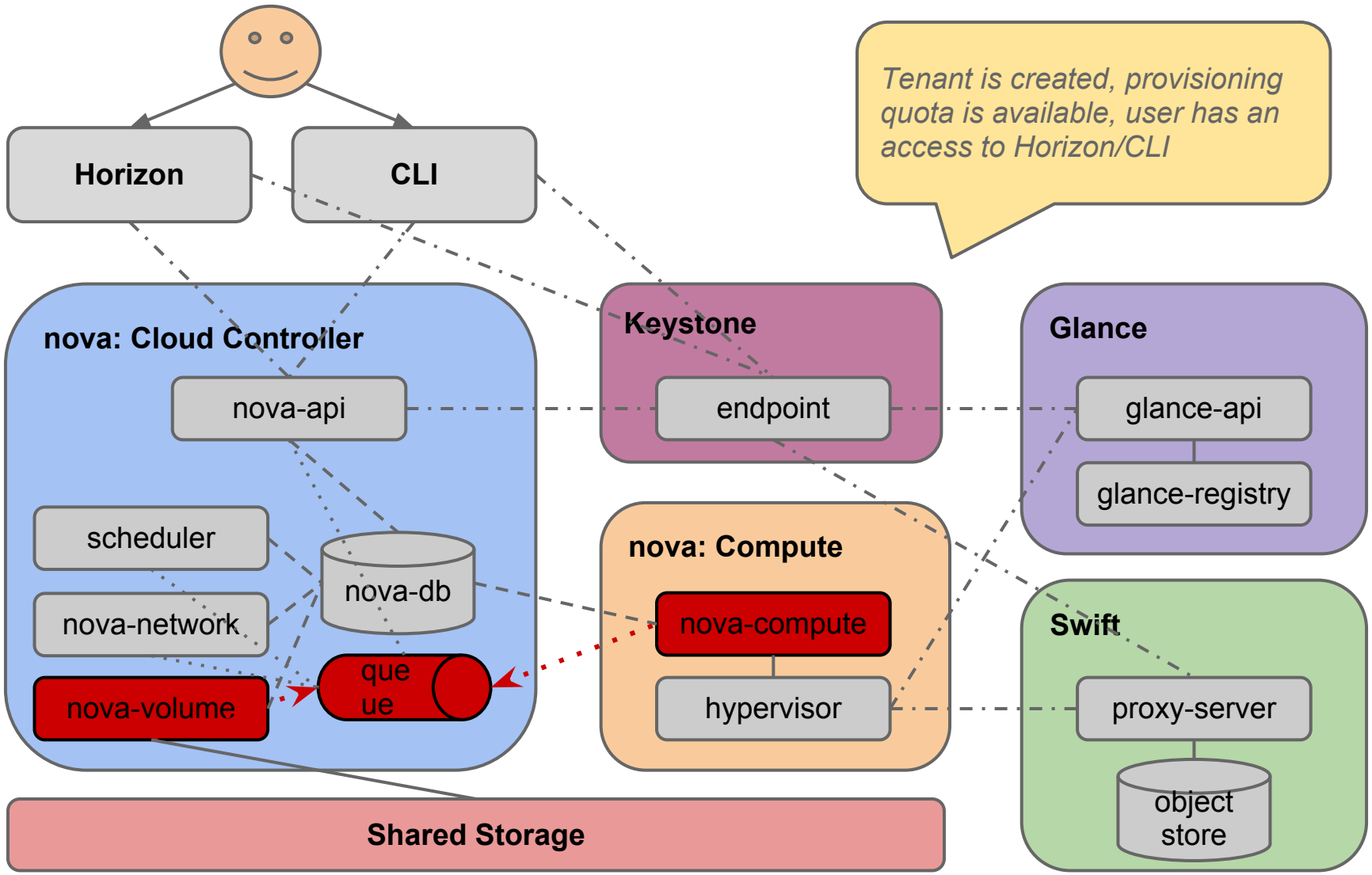
OpenStack User

1. Allocate public IP for tenant within given quota
2. Associate public IP with VM
   a. Find host
   b. Add IP address to public network interface of the host
   c. NATting all network traffic via associated floating IP

MIRANTIS

© Mirantis, Inc, 2012. All rights reserved.

Step 15: Request volume attachment



MIRANTIS

© Mirantis, Inc, 2012. All rights reserved.

nova-volume

"nova-volume manages the creation, attaching and detaching of persistent volumes to compute instances"

MIRANTIS

© Mirantis, Inc, 2012. All rights reserved.

nova-volume summary

- Optional

MIRANTIS

© Mirantis, Inc, 2012. All rights reserved.

nova-volume summary

- Optional
- iSCSI solution which uses LVM

MIRANTIS

© Mirantis, Inc, 2012. All rights reserved.

nova-volume summary

- Optional
- iSCSI solution which uses LVM
- Volume can be attached only to 1 instance at a time

MIRANTIS

© Mirantis, Inc, 2012. All rights reserved.

nova-volume summary

- Optional
- iSCSI solution which uses LVM
- Volume can be attached only to 1 instance at a time
- Persistent volumes keep their state independent of instances

MIRANTIS

© Mirantis, Inc, 2012. All rights reserved.

nova-volume summary

- Optional
- iSCSI solution which uses LVM
- Volume can be attached only to 1 instance at a time
- Persistent volumes keep their state independent of instances
- Within single OpenStack deployment different storage providers cannot be used

MIRANTIS

© Mirantis, Inc, 2012. All rights reserved.

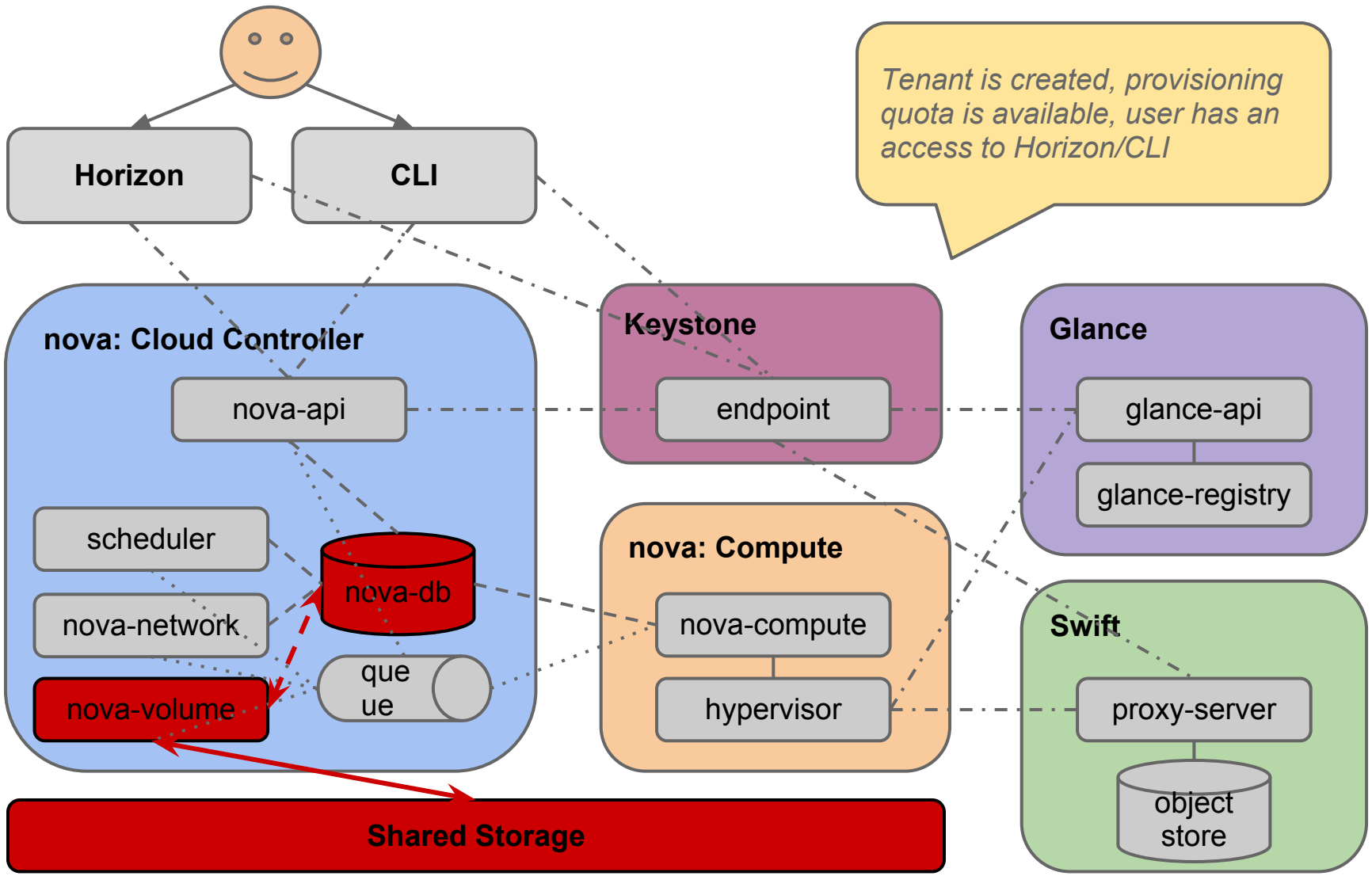
nova-volume drivers

- iSCSI
- Xen Storage Manager
- Nexenta
- NetApp
- SAN

MIRANTIS

© Mirantis, Inc, 2012. All rights reserved.

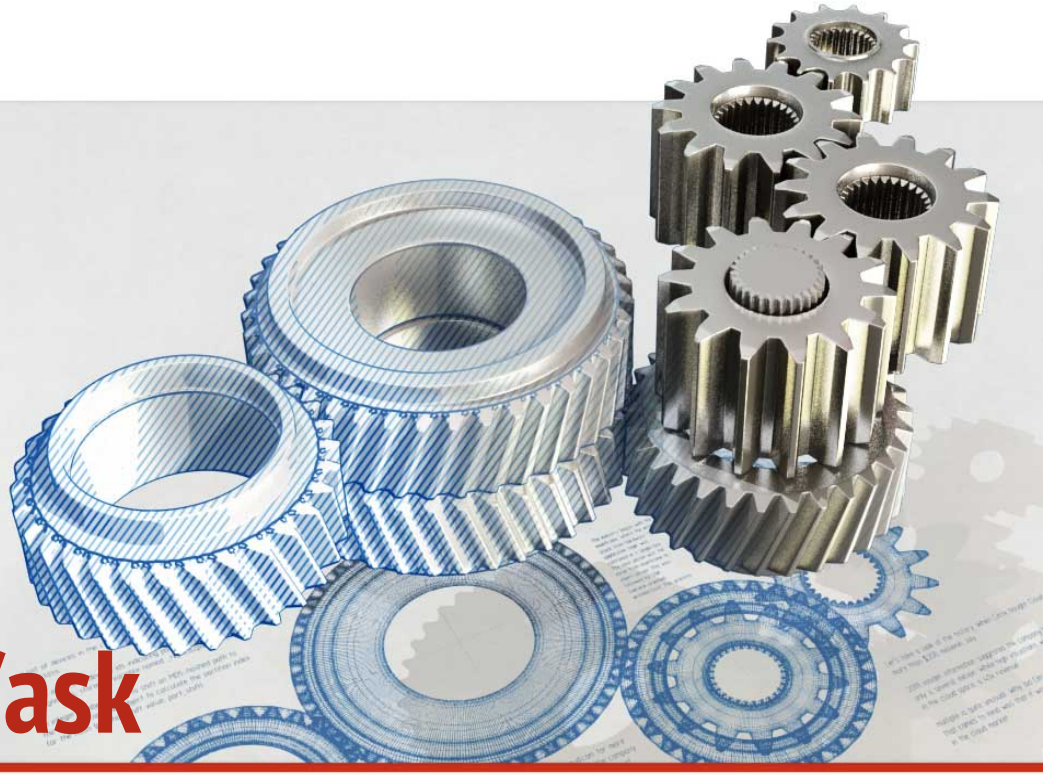
Initial State



MIRANTIS

© Mirantis, Inc, 2012. All rights reserved.

MIRANTIS



www.mirantis.com/ask

Mirantis

September 2012