OpenStack Architecture and Pattern Deployment using Heat

Ruediger Schulze

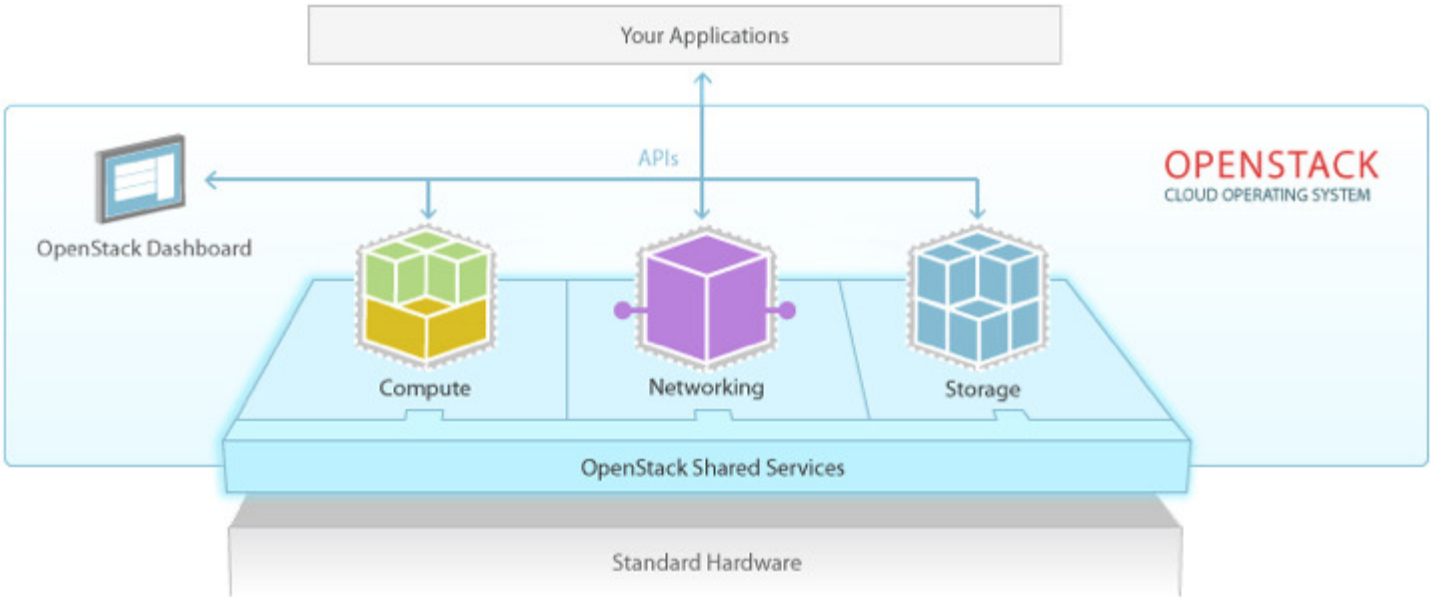
IBM Cloud

Agenda

1. OpenStack Architecture
2. Pattern Deployment with Heat
3. Summary

What is OpenStack ?

"At its most basic level, **OpenStack is a set of open-source software tools for building clouds**. The code that comes out of the OpenStack community is used to deploy compute, storage and networking resources in a data center …. Users can take control of their application infrastructure environment and manage those resources faster and with greater agility."
Jonathan Bryce (Executive Director OpenStack Foundation) | Information Week | OpenStack: Driving The Software-Defined Economy 30 January 2015



OpenStack Mission Statement

The OpenStack Mission: to produce the **ubiquitous** Open Source Cloud Computing platform that will meet the needs of **public and private clouds** regardless of size, by being **simple to implement** and **massively scalable**.

OpenStack is Huge!

Because an open interoperable Cloud is critical for flexible cloud deployment and customer success…

openstack™

Platinum Sponsors

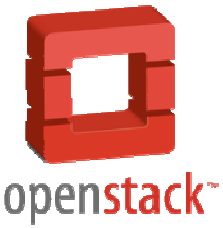
rackspace, intel, redhat, suse, IBM, at&t, hp, ubuntu

Gold Sponsors

HITACHI Inspire the Next, CISCO, YAHOO!, DreamHost, aptira, NetApp, DELL, JUNIPER NETWORKS, HUAWEI, MORPHLABS, vmware, MIRANTIS, cloudscaling, ERICSSON, NEC, Piston, eNovance Cloud & Managed Services Provider

22	IBM has 22 core contributors
2	IBM is #2 in contributions to OpenStack integrated projects
+100	IBMers active developers in OpenStack projects
+400	IBMers working on OpenStack– from formation of the Foundation to Code Quality & New Function

Key messages from the OpenStack Foundation 2H/2015 Marketing Plan



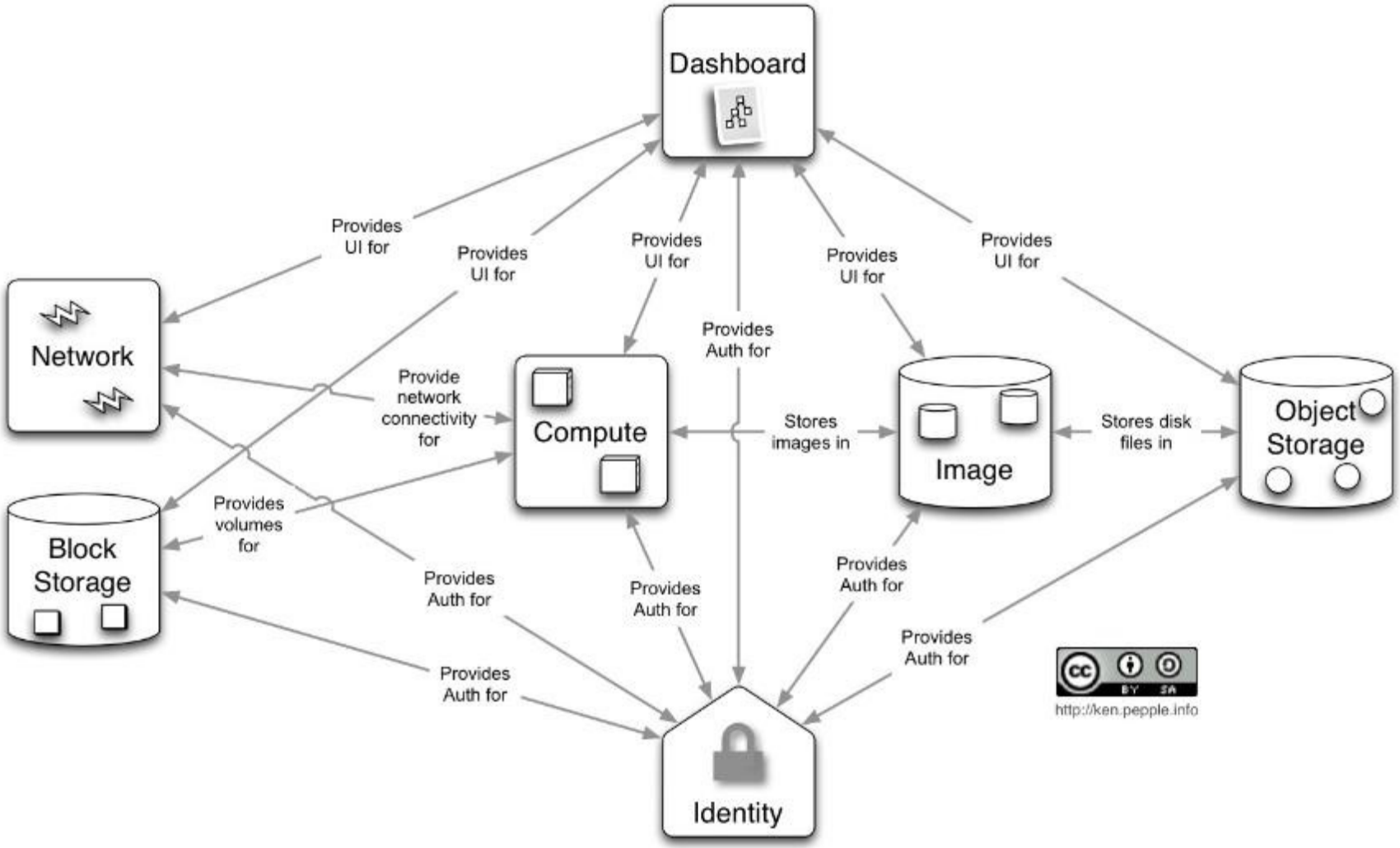
OpenStack Powered Planet: realizing the vision of an **interoperable, global network of public / private clouds** running a **common set of services**.

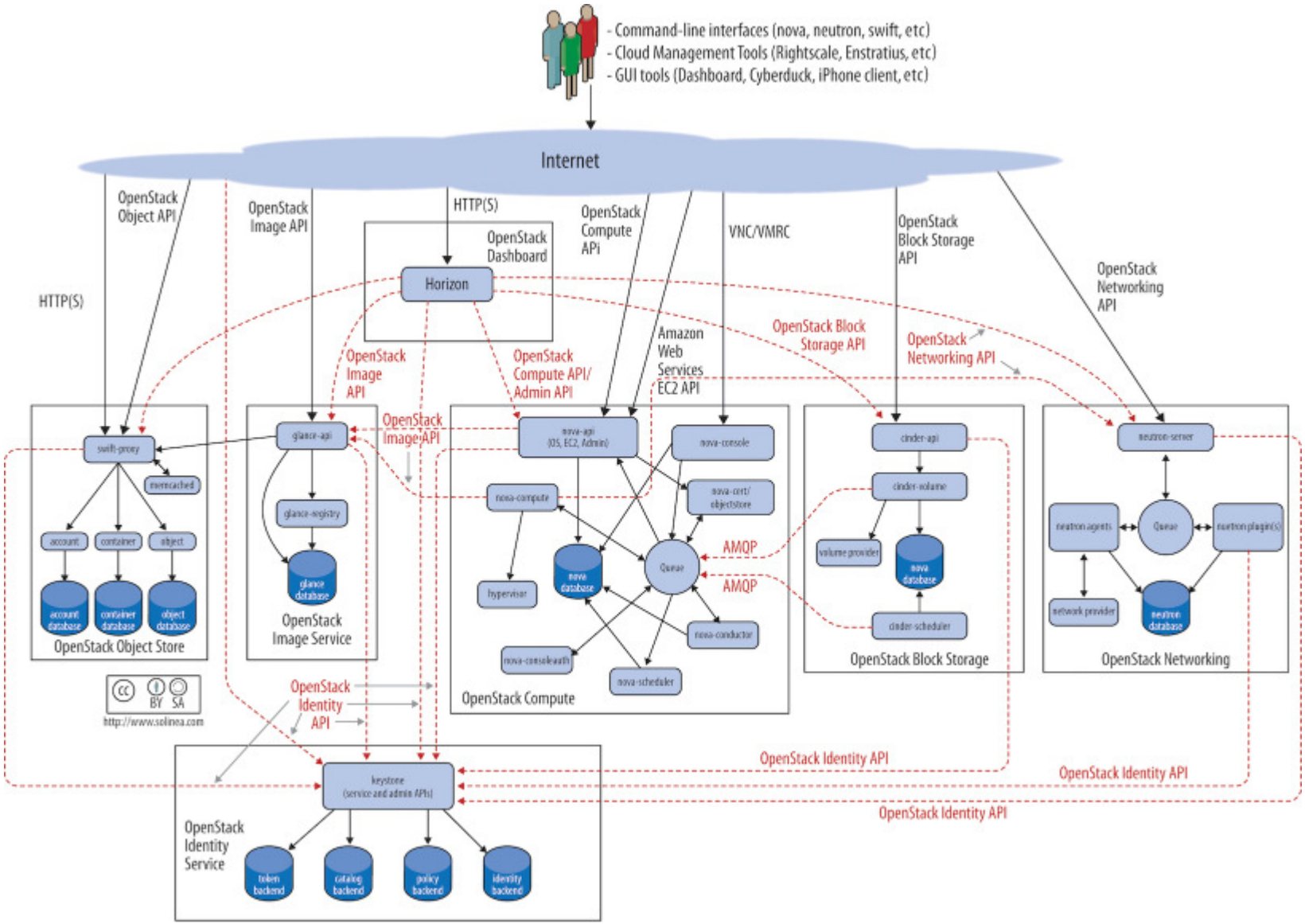
Integration Engine: OpenStack provides a **single pane of glass** for your data center technologies. Users don't want to create islands in their data centers; they want a single operating system to manage VMs, containers, bare metal and whatever comes next.

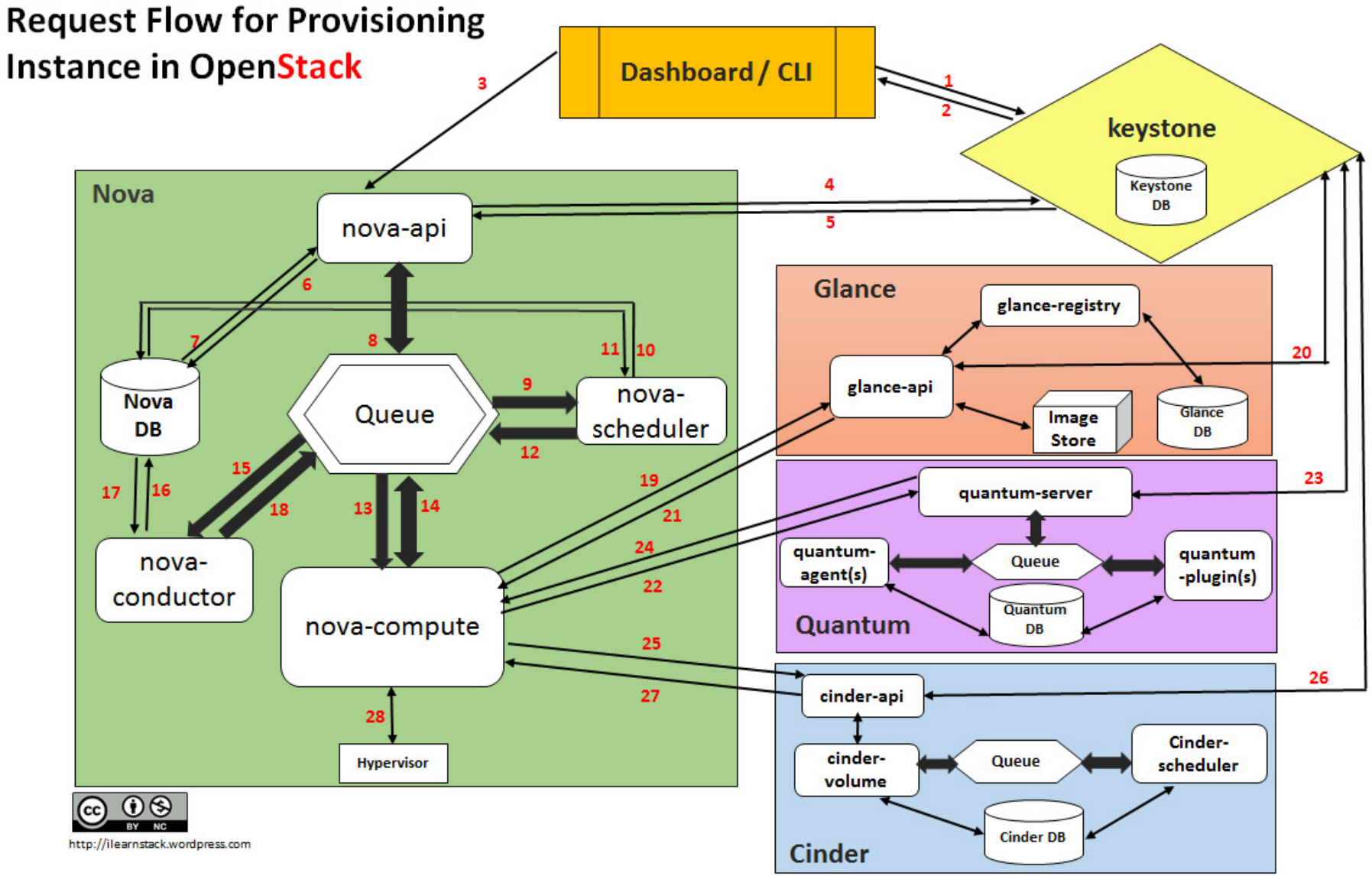
Platform for the next 10 years: OpenStack is a sustainable choice based on the breadth of its platform, community engagement, and integration of new technologies. Position OpenStack as the **path to production for new technologies as they emerge, like containers.**

The business case for OpenStack, focusing on end users.

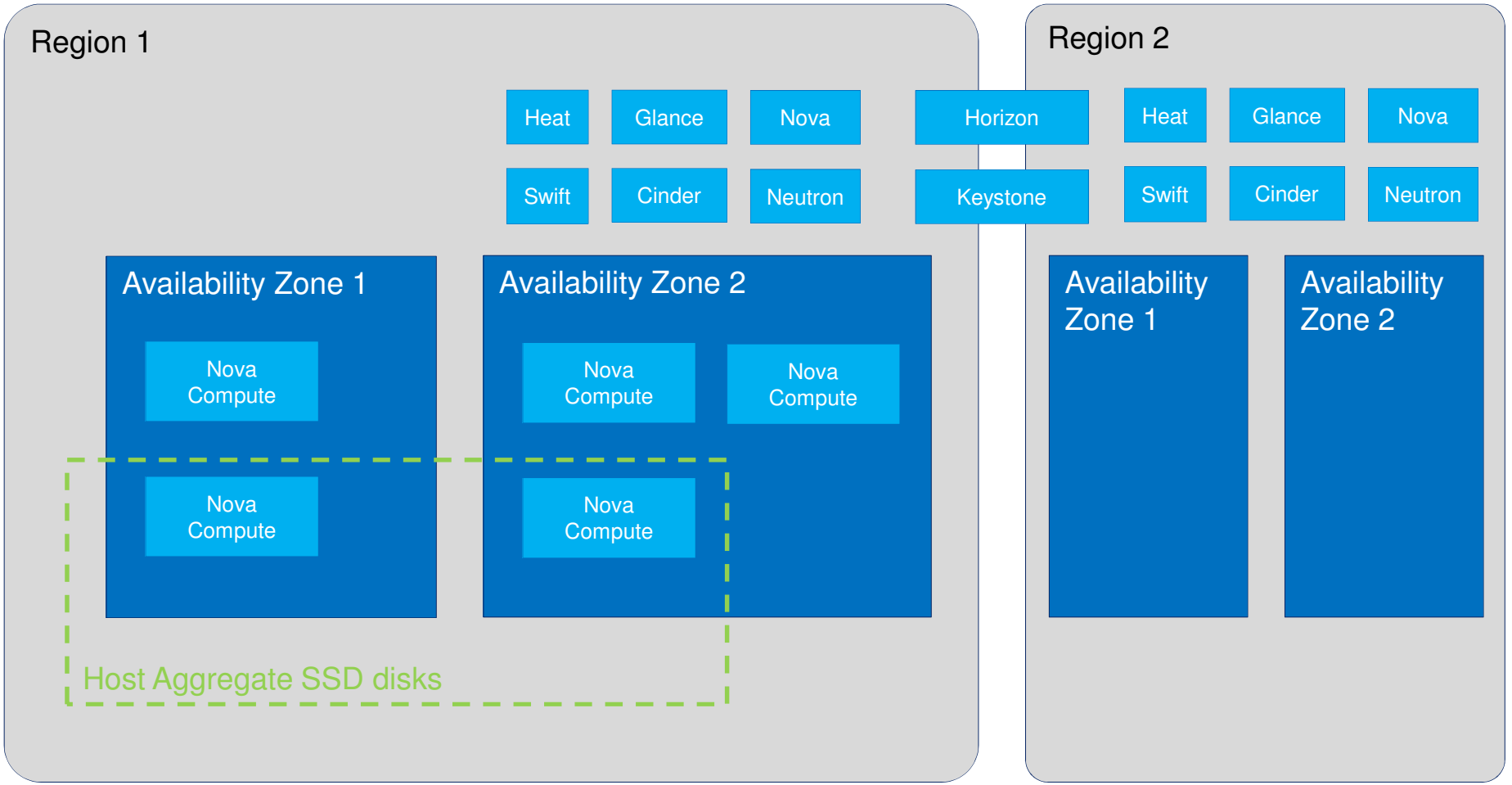
Conceptual Architecture



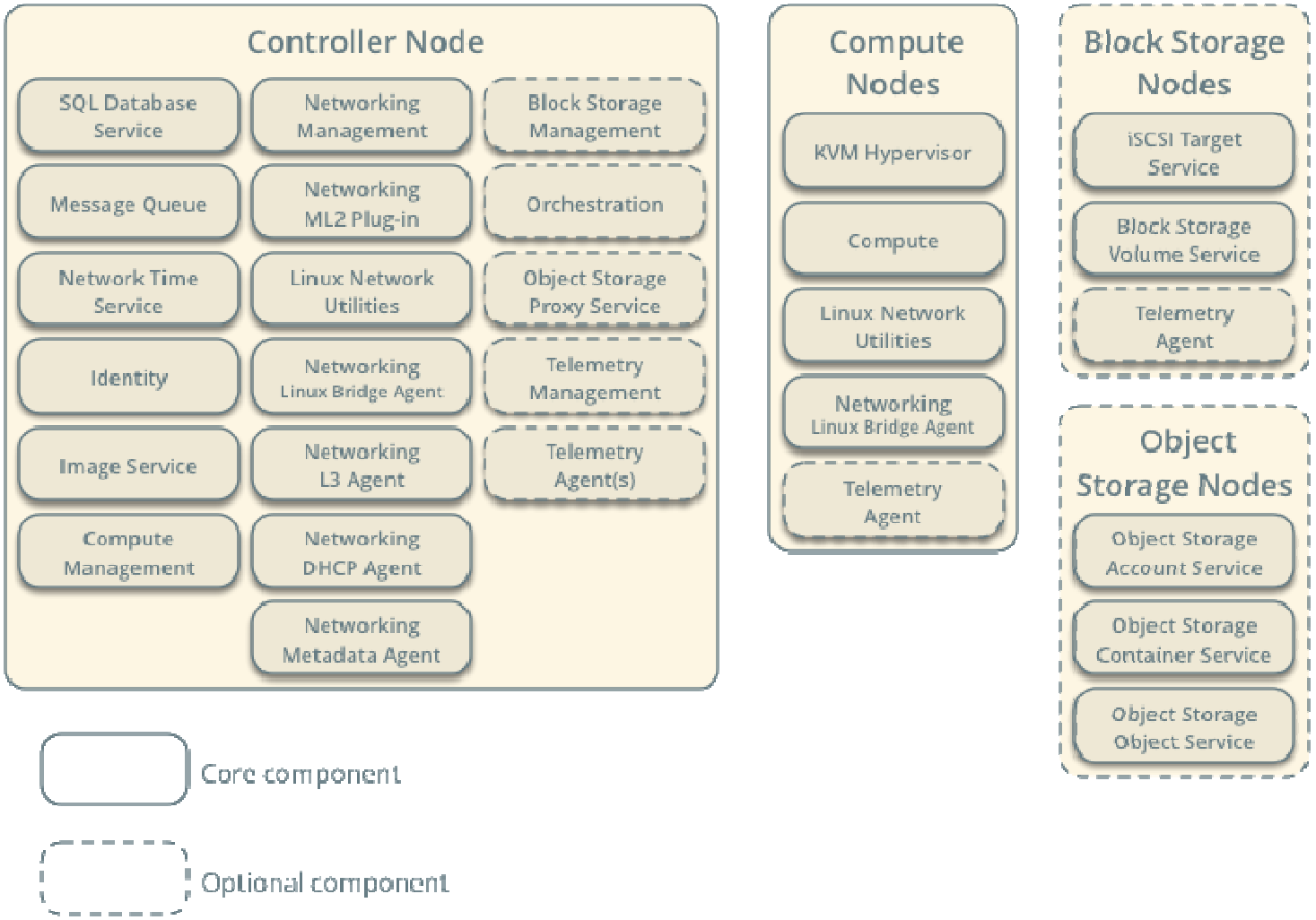

- Command-line interfaces (nova, neutron, swift, etc)
- Cloud Management Tools (Rightscale, Enstratius, etc)
- GUI tools (Dashboard, Cyberduck, iPhone client, etc)
Internet
OpenStack Object API
OpenStack Image API
HTTP(S)
OpenStack Compute API
VNC/VMRC
OpenStack Block Storage API
OpenStack Networking API
HTTP(S)
OpenStack Dashboard
Horizon
OpenStack Image API
OpenStack Compute API/ Admin API
Amazon Web Services EC2 API
OpenStack Block Storage API
OpenStack Networking API
OpenStack Image API
swift-proxy
memcached
account
container
object
account database
container database
object database
OpenStack Object Store
glance-api
glance-registry
glance database
OpenStack Image Service
nova-api (OS, EC2, Admin)
nova-console
nova-compute
nova-cert/ objectstore
hypervisor
nova database
Queue
AMQP
AMQP
nova-conductor
nova-consoleauth
nova-scheduler
OpenStack Compute
cinder-api
cinder-volume
volume provider
nova database
cinder-scheduler
OpenStack Block Storage
neutron-server
neutron agents
Queue
neutron plugin(s)
network provider
neutron database
OpenStack Networking
OpenStack Identity API
keystone (service and admin APIs)
OpenStack Identity Service
token backend
catalog backend
policy backend
identity backend
OpenStack Identity API
OpenStack Identity API
OpenStack Identity API
BY SA
http://www.solinea.com


Request Flow for Provisioning Instance in OpenStack
Dashboard / CLI
keystone
Keystone DB
Nova
nova-api
Nova DB
Queue
nova-scheduler
nova-conductor
nova-compute
Hypervisor
Glance
glance-registry
glance-api
Image Store
Glance DB
Quantum
quantum-server
quantum-agent(s)
Queue
quantum -plugin(s)
Quantum DB
Cinder
cinder-api
cinder-volume
Queue
Cinder-scheduler
Cinder DB
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
http://ilearnstack.wordpress.com

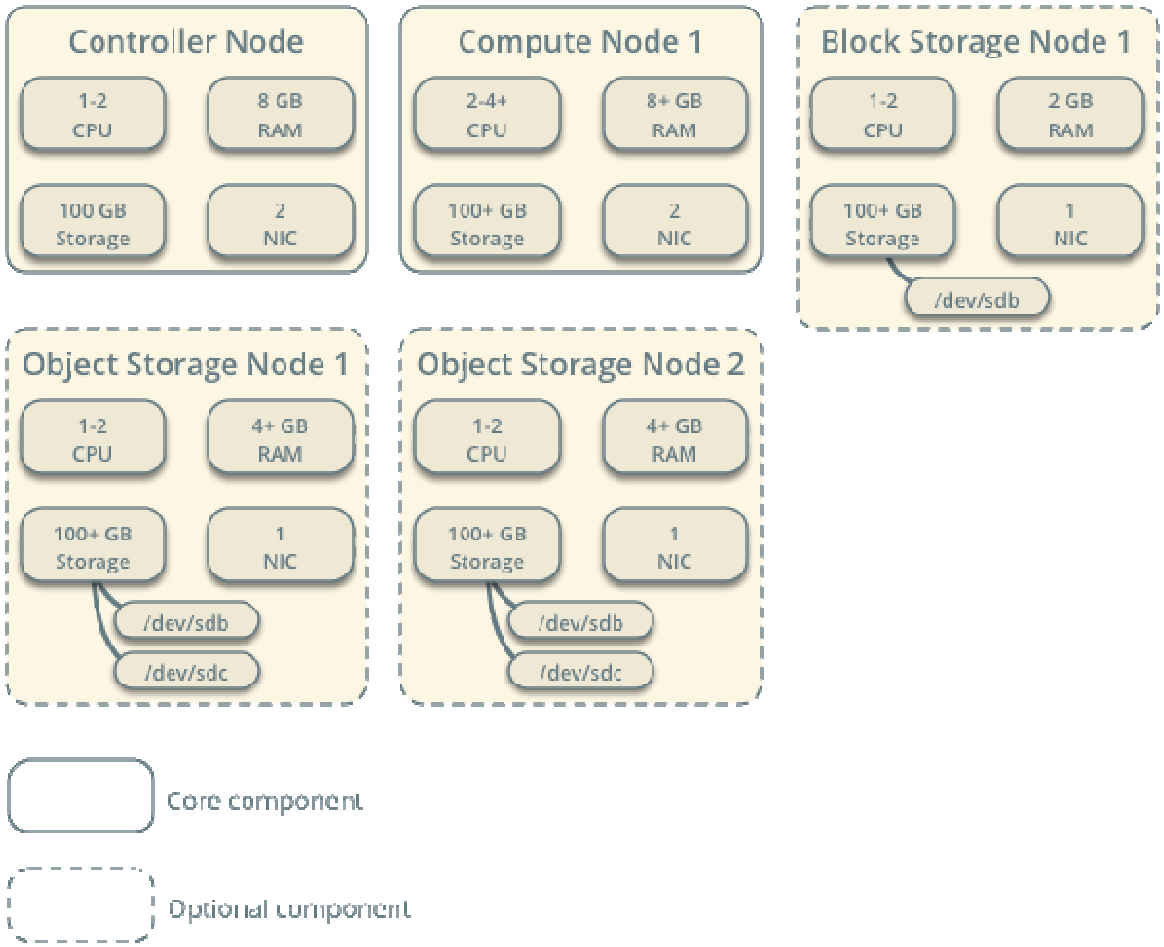
OpenStack Segregation



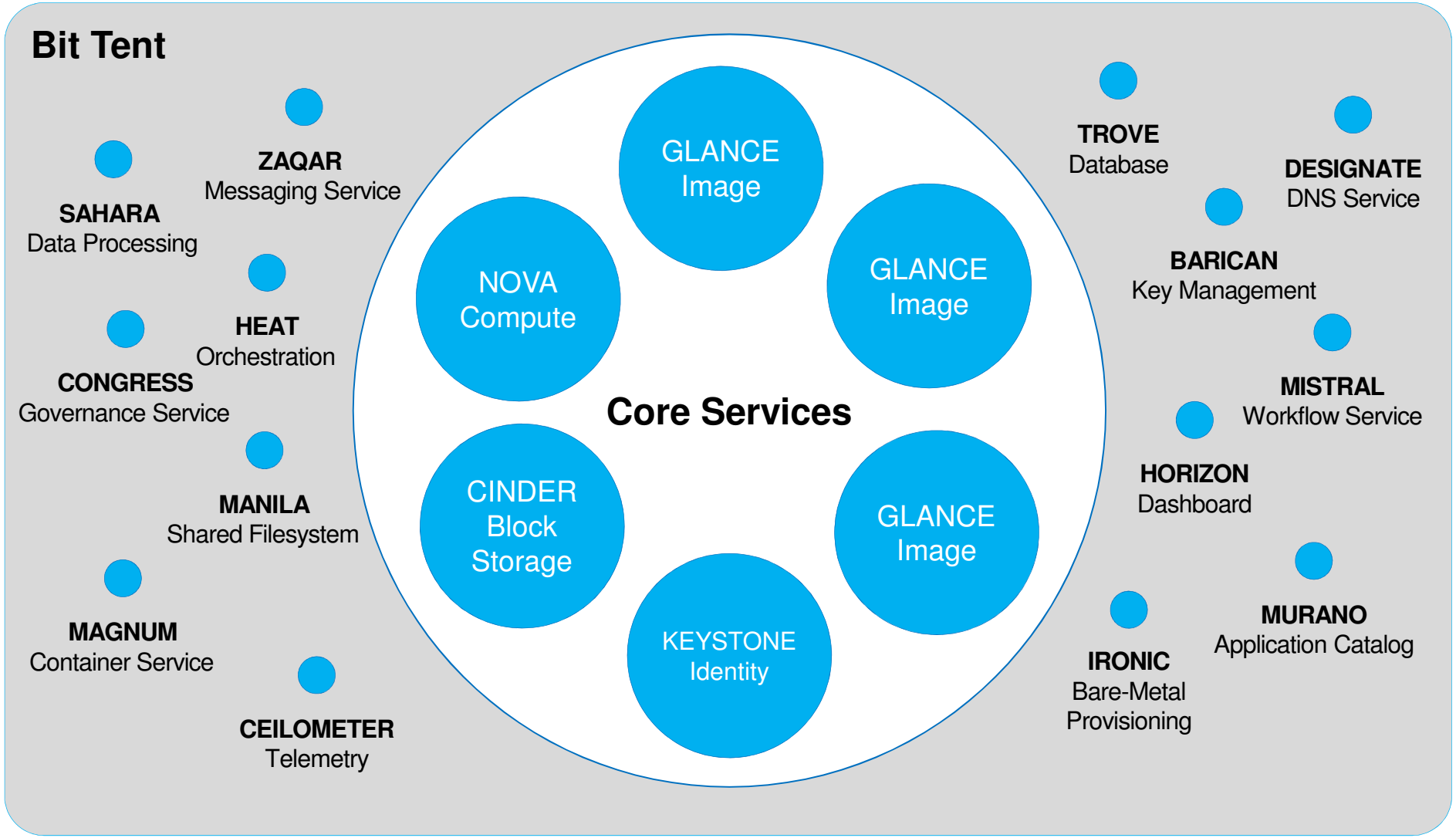
Deployment Model



Hardware Requirements



The Big Tent and Core Services



Patterns and Openstack HEAT

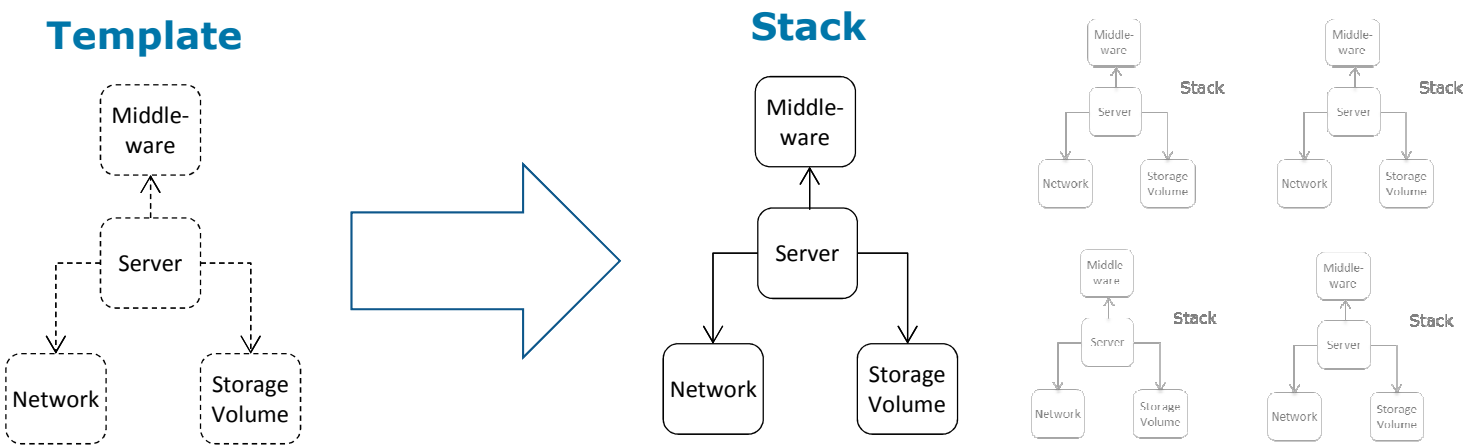
A ***Pattern*** is

- a **reusable template** that describes the structure of an IT solution and
- contains an **abstract model that represents the infrastructure and application components** of the IT solution, for example, servers, disks and an application but also policies and pre-defined configurations.

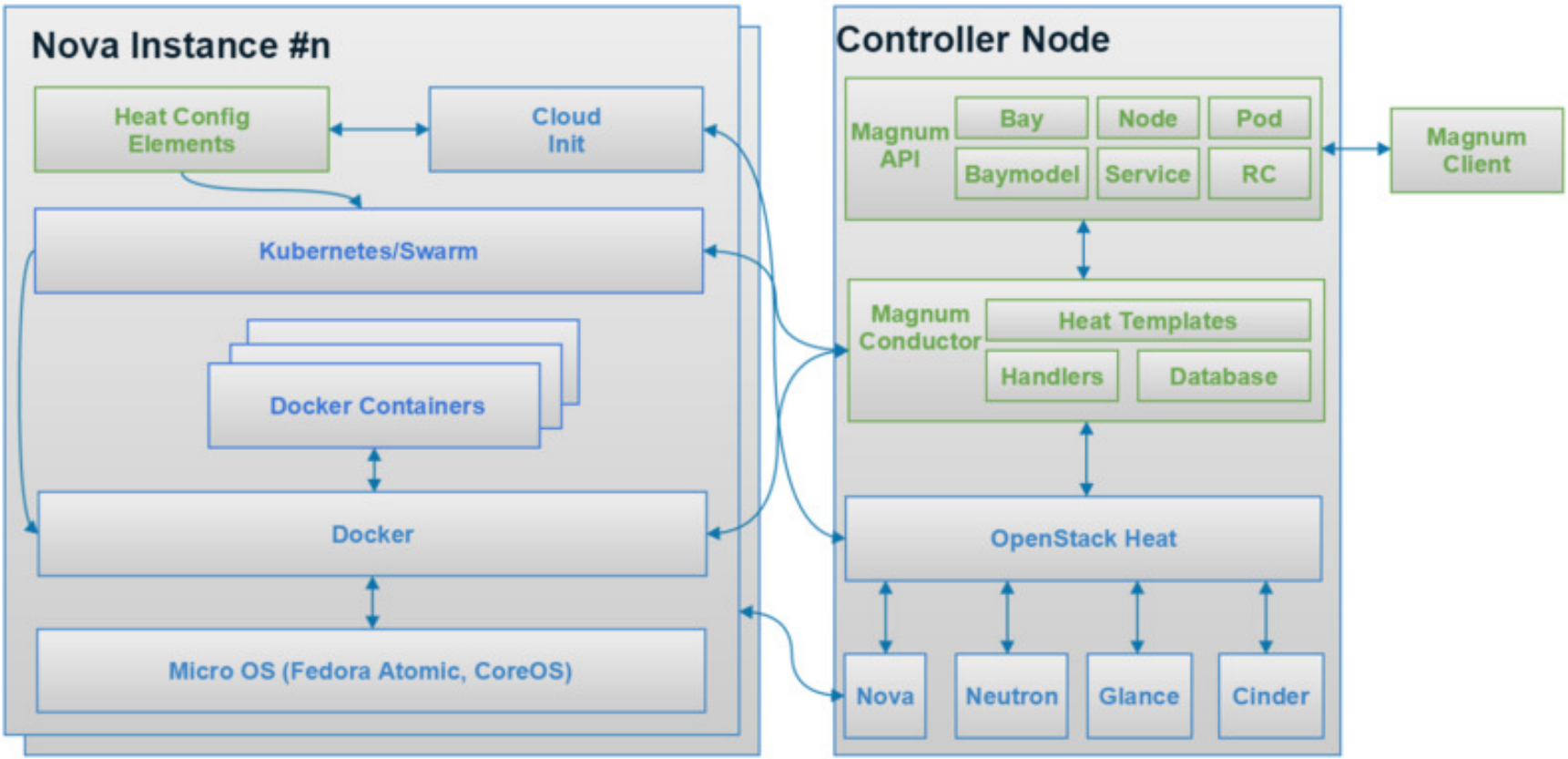
OpenStack ***Heat*** is

- a **open-standard**, **template-based orchestration engine** on-top of OpenStack APIs

For example, a *middleware pattern* may be implemented in form of a OpenStack Heat template and a set of Chef artefacts.



OpenStack Magnum and Containers



What is DevStack?

- A documented shell script to build complete OpenStack development environments
- devstack is written in bash
- Supported on both Ubuntu and Fedora Linux
- Able to run in both single and multi-node environments
- Not Intended for deployment of OpenStack in production environments

Getting started with devstack

1. Ubuntu Server 14.04.3 LTS
2. sudo apt-get update
3. sudo apt-get install git
4. git clone https://www.github.com/openstack-dev/devstack.git -b stable/liberty
5. cd devstack
6. Vim local.conf

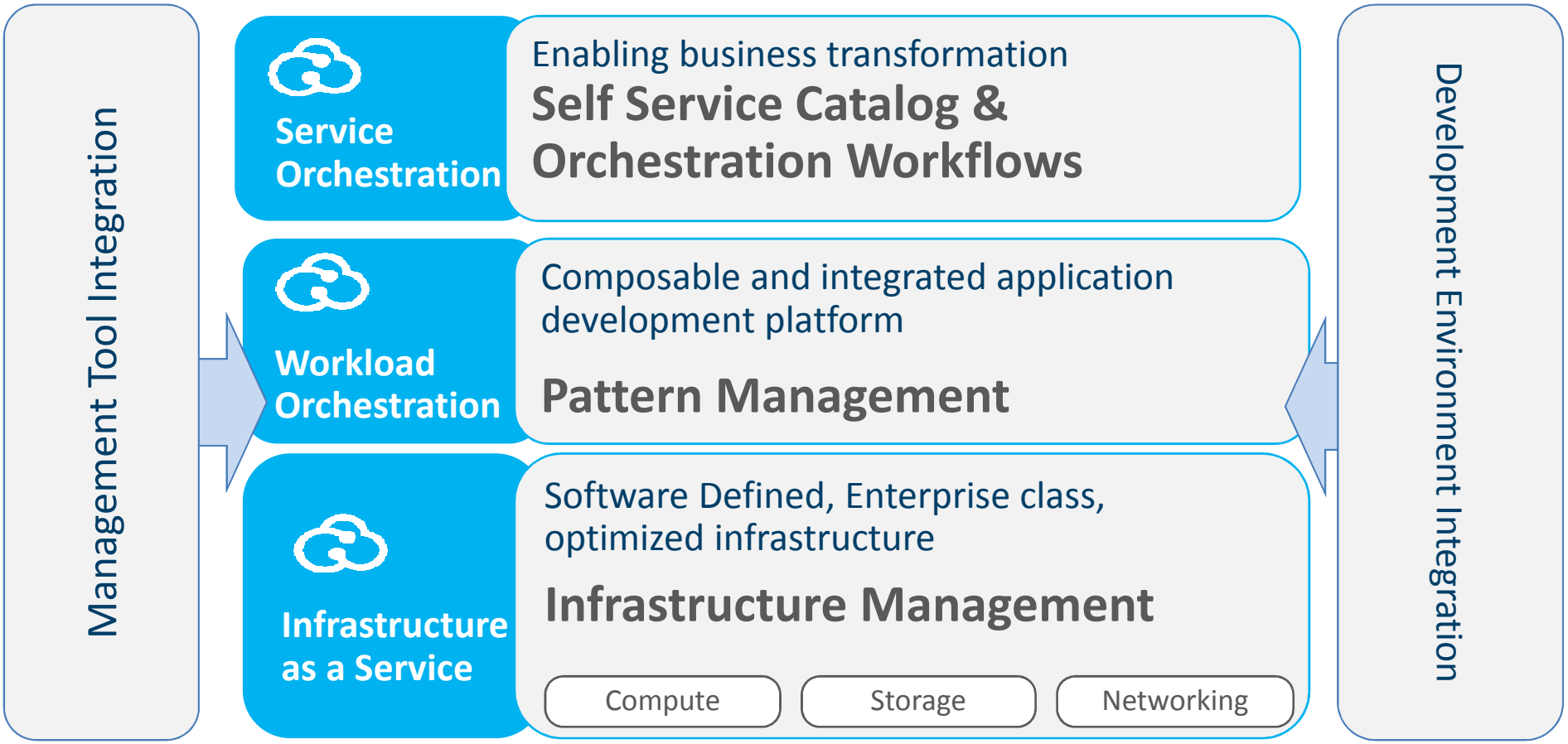
```
[[local|localrc]]
ADMIN_PASSWORD=husa4vik
DATABASE_PASSWORD=$ADMIN_PASSWORD
RABBIT_PASSWORD=$ADMIN_PASSWORD
SERVICE_PASSWORD=$ADMIN_PASSWORD
SERVICE_TOKEN=$ADMIN_PASSWORD
SWIFT_HASH=$ADMIN_PASSWORD
# Enable Neutron
ENABLED_SERVICES+=,q-svc,q-agt,q-dhcp,q-l3,q-meta,neutron
# Optional, to enable tempest configuration as part of devstack
enable_service tempest
# Enable heat services
ENABLED_SERVICES+=,heat,h-api,h-api-cfn,h-api-cw,h-eng
# Enable Swift
enable_service s-proxy s-object s-container s-account
```

7. ./stack.sh

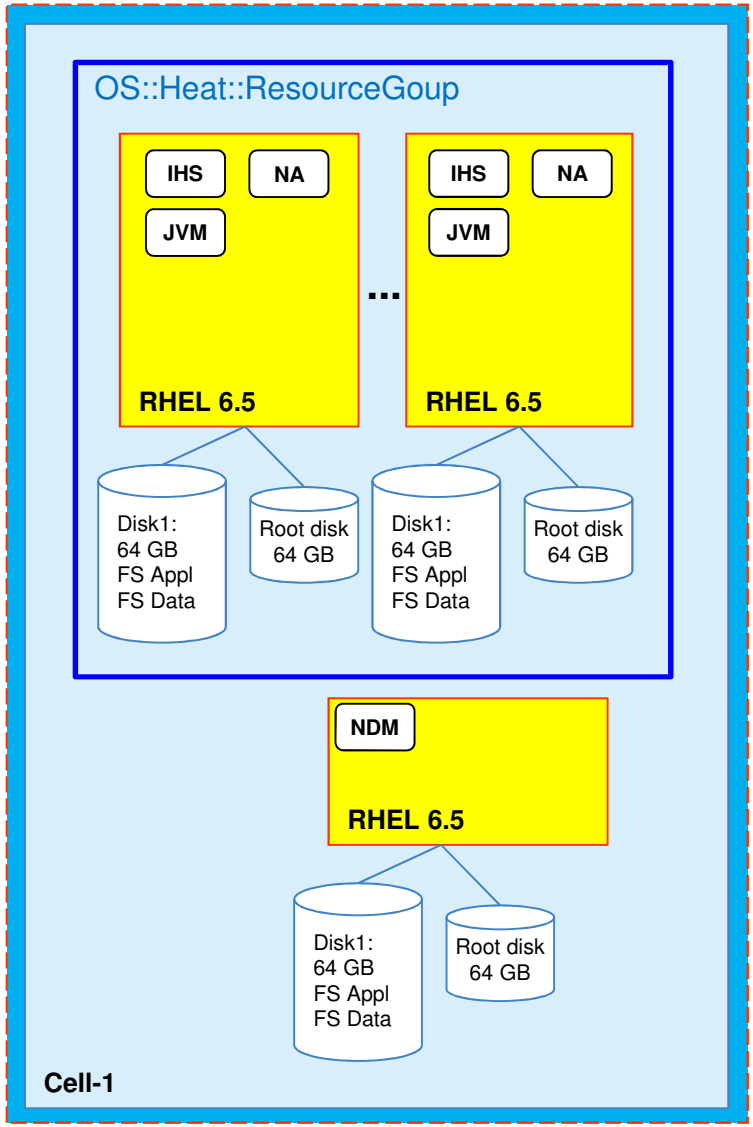
Agenda

1. OpenStack Architecture
2. Pattern Deployment with Heat
3. Summary

Levels of Cloud Orchestrations



WAS ND 8.5.5 clustered with horizontal scalability

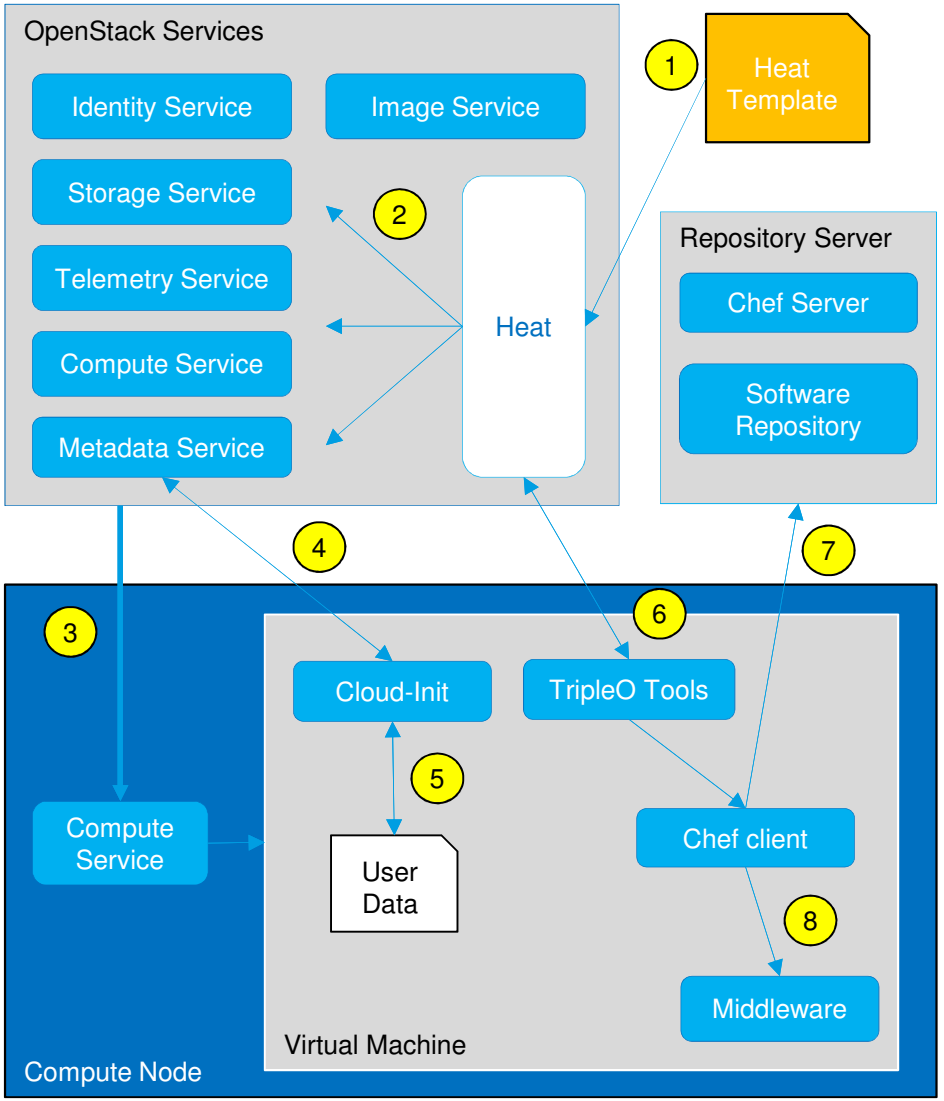


Pattern Specification:

- WAS ND 8.5.5 Multi Node Installation:
  - NDM on dedicated VM
  - Single Cell
  - OS::Heat::ResourceGroup with 1 to n VMs as Application Server nodes
  - Each VM with UCD 6.1 agent
- IHS and JVM will be configured later on by UCD
- Each VM with:
  - Red Hat Enterprise Linux 6.5
  - Default T-Shirt Size: Small 1 vCPUs, 2 GB RAM
  - Root Disk: 64 GB
  - Additional Disk with 64 GB, partitioned and with volume groups using script provided by SOD team
  - TSM file backup enabled
- Installation of WAS 8.5.5 on file system of application volume group
- Availability Zone (DC) selectable
- SLA and disk parameters selectable
  - Default: Bronze and no disk mirroring
- Security Zone and Patch category selectable

OpenStack deployment of a middleware pattern

1. Heat stack-create command is run with the specified Heat template containing the definition of software resources.
2. Heat Engine orchestrates the deployment of associated resources.
3. A new virtual machine is provisioned.
4. On first boot, cloud-init retrieves configuration data from metadata service.
5. The virtual machine is initialized with the configuration and user-data scripts are executed.
6. TripleO tools retrieve from the Heat metadata service the script for installing and initializing the Chef client of the associated software configuration resource.
7. Based on the role assigned to the virtual machine, the Chef cookbook is run on the local node and the installation media is retrieved
8. Middleware is installed and configured



CMS Approach for Pattern Deployment

Service Orchestration:

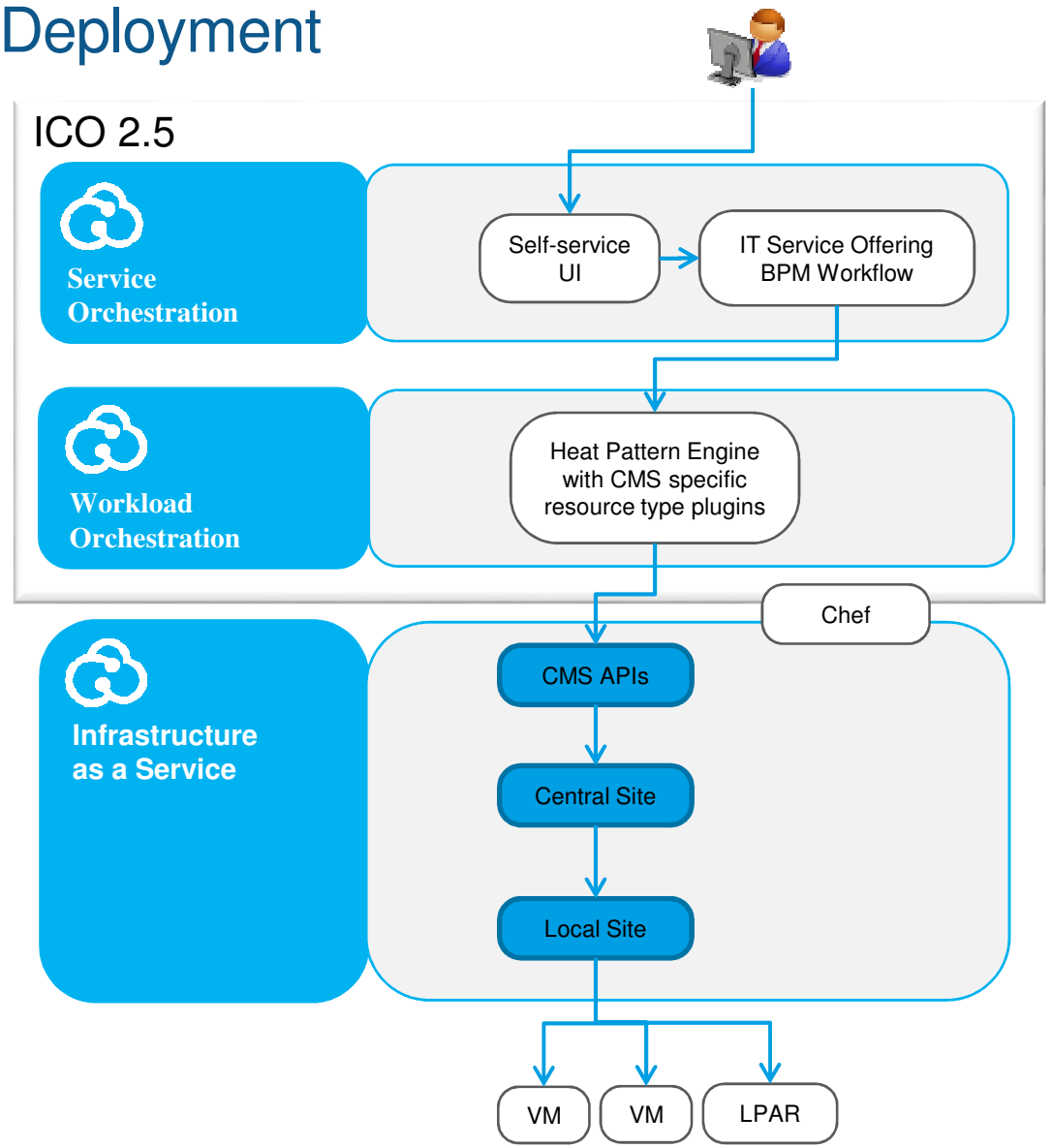
- IT Service Offering selected from ICO Self-service UI or invoked via ICO API
- Deployment of the components of the IT Service Offering is orchestrated by ICO BPM

Workload Orchestration:

1. OpenStack Heat as IBM strategic engine for pattern deployment
2. CMS pattern in open-standard format OpenStack Heat Template (HOT) format is input to the Heat engine
3. Use of Chef for middleware deployment, referenced by the pattern template

Infrastructure as a Service:

1. CMS provides infrastructure resources like compute, storage and network for the pattern instances



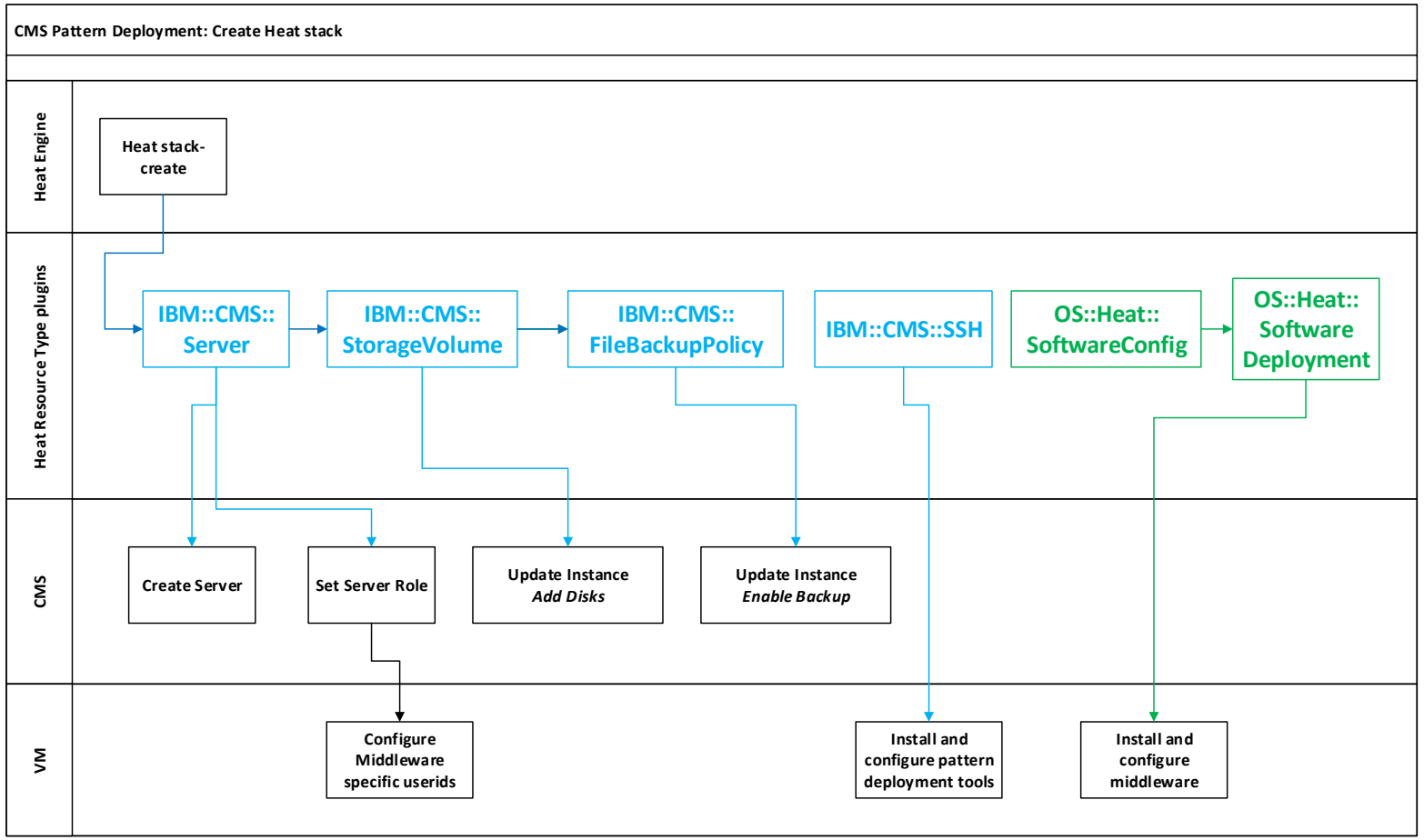
CMS specific OpenStack Resource Types

Resource	OpenStack Resource Type	CMS Resource Type	Description
Server	OS::Nova::Server	IBM::CMS::Server	Create and managed VMs
Storage	OS::Cinder::Volume OS::Cinder::VolumeAttachment	IBM::CMS::Volume	Create and managed additional disks of a VM
IP address	OS::Neutron::Port	IBM::CMS::IPAddress	Reserve and release IP addresses from a given security zone (required for MQ pattern)
File Backup	n/a	IBM::CMS::FileBackup Policy	Enable File Backup for a VM
SSH access	n/a	IBM::CMS::SSH	Access a VM via pre-configure ssh key pair in order to run commands or scripts

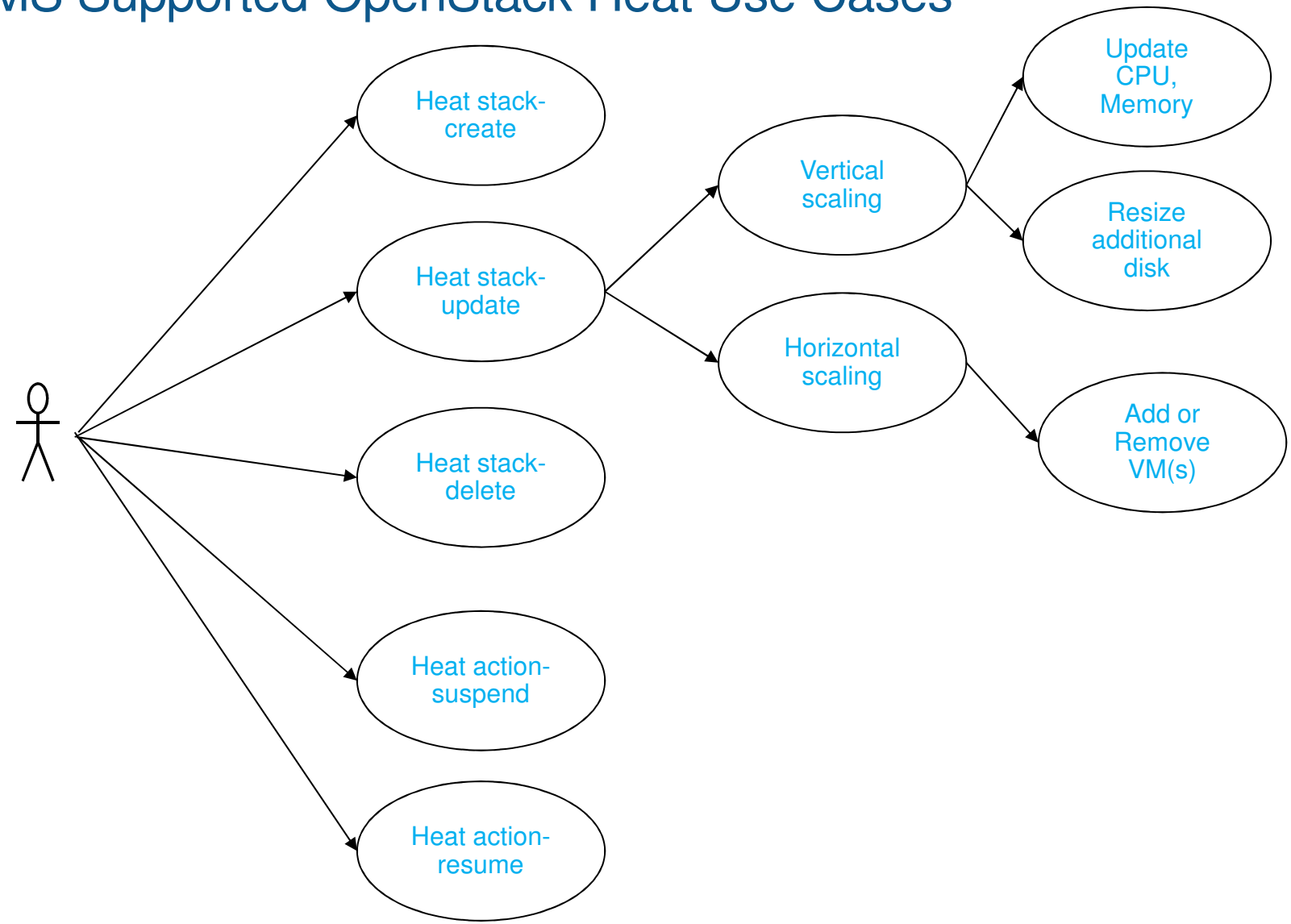
Use of Heat resource type plugins for deployment

Workload Orchestration

Infrastructure as a Service



CMS Supported OpenStack Heat Use Cases



Heat Resource Plug-in Development

A resource plug-in needs to extend a base Resource class and implement some relevant life cycle handler methods.

Life-cycle methods of a resource:

Create:
The plug-in should create a new physical resource.

Update:
The plug-in should update an existing resource with new configuration or tell the engine that the resource must be destroyed and re-created. This method is optional; the default behavior is to create a replacement resource and then delete the old resource.

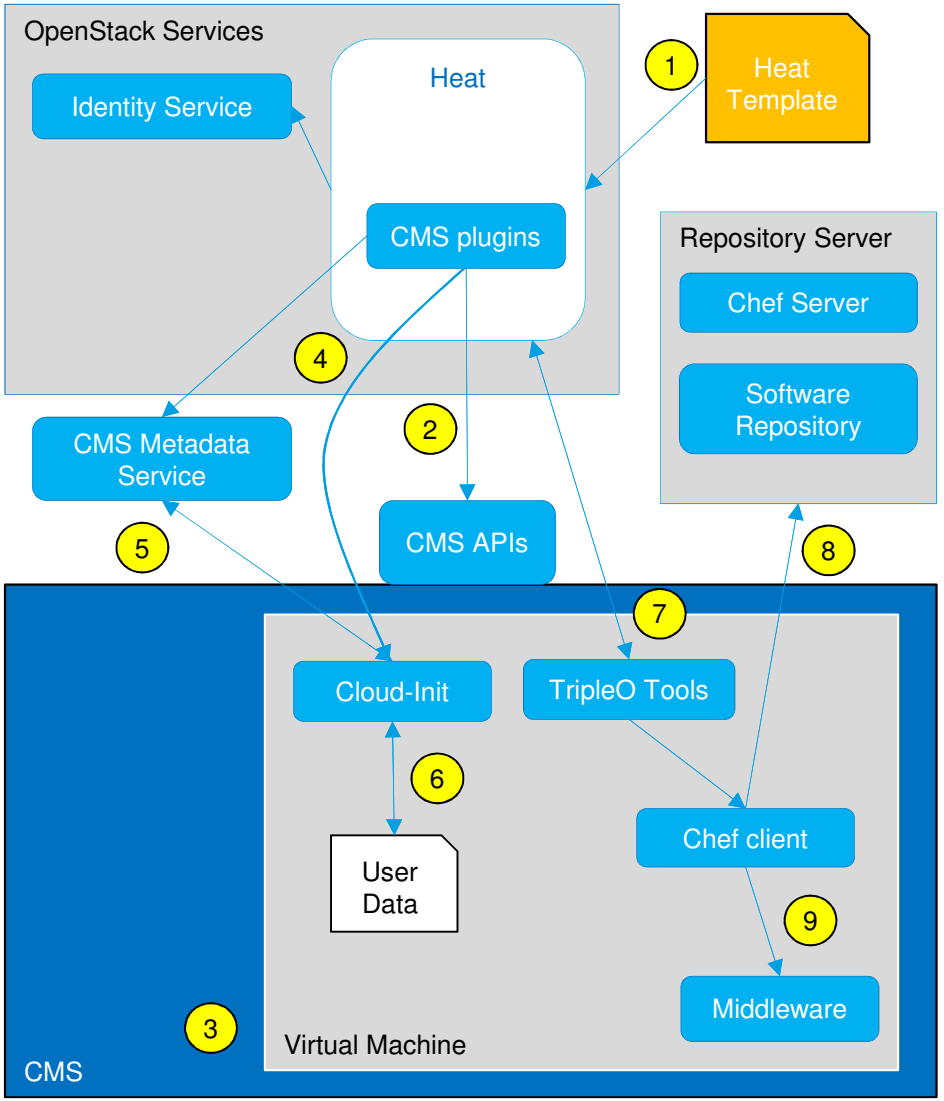
Suspend:
The plug-in should suspend operation of the physical resource; this is an optional operation.

Resume:
The plug-in should resume operation of the physical resource; this is an optional operation.

Delete:
The plug-in should delete the physical resource.

CMS deployment of a Middleware Pattern

1. Heat stack-create command is run with the specified Heat template containing the definition of software resources.
2. Heat Engine orchestrates the deployment of associated infrastructure resources using the CMS API.
3. A new managed or unmanaged virtual machine is provisioned by CMS.
4. Cloud-init is prepared on the virtual machine for execution.
5. Cloud-init is executed as a post-provisioning step and retrieves configuration data from the CMS metadata service.
6. The virtual machine is initialized with the pattern-specific configuration and user-data scripts are executed.
7. TripleO tools retrieve from the Heat metadata service the script for installing and initializing the Chef client of the associated software configuration resource.
8. Based on the role assigned to the virtual machine, the Chef cookbook is run on the local node and the installation media is retrieved
9. Middleware is installed and configured



Agenda

1. OpenStack Architecture
2. Pattern Deployment with Heat
3. Summary

Summary

1. What is OpenStack and why is it important?
2. Which ones are the OpenStack core components and which functions do they have?
3. What can be done with Heat and Magnum?
4. What is devStack and what does it take to get started?