ESCALATOR DESIGN CONSIDERATIONS

Release brahmaputra.1.0 (7cb2951)

OPNFV

March 03, 2016

CONTENTS

Contents:

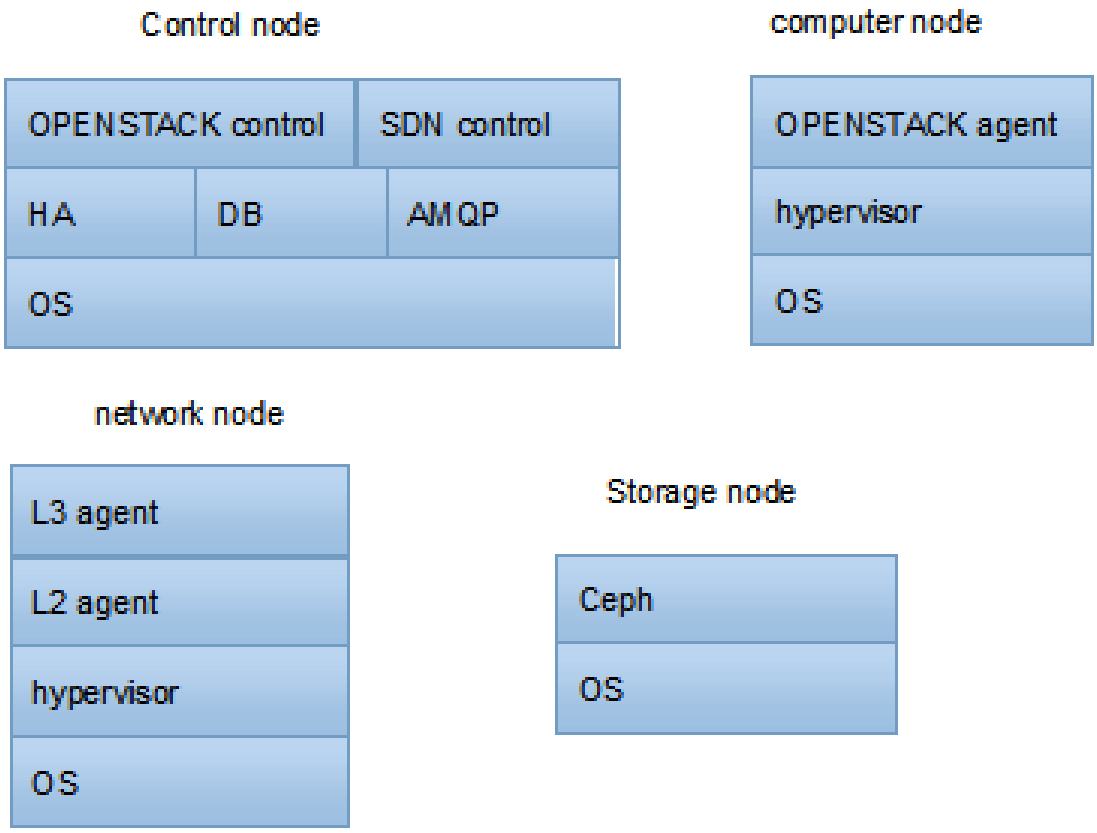
CHAPTER

ONE

REFERENCE ARCHITECTURE

This section describes the reference architecture, the function blocks, and the function entities of Escalator for the reader to well understand how the basic functions to be organized.

The software stack on each node is generally as shown in the table below.



Since the upgrading of control node will not affect the VNFs service in the blue box, this chapter will focusing on the upgrading of compute nodes.

1.1 Precondition of Upgrade

1.1.1 Upgrade related modules in VIM

Upgrade operations are initiated by the user through the VIM. For VIM, upgrade management mainly contains the object:

- **Upgrade Manager**

Mainly responsible for control of the upgrade process.The Escalator is required to know the software deployment information of the platform and will use these information during the upgrading. It will be collected from some place, such as the Installer, Deploy Manager and Escalator itself, etc.

- **VIM Interface**:

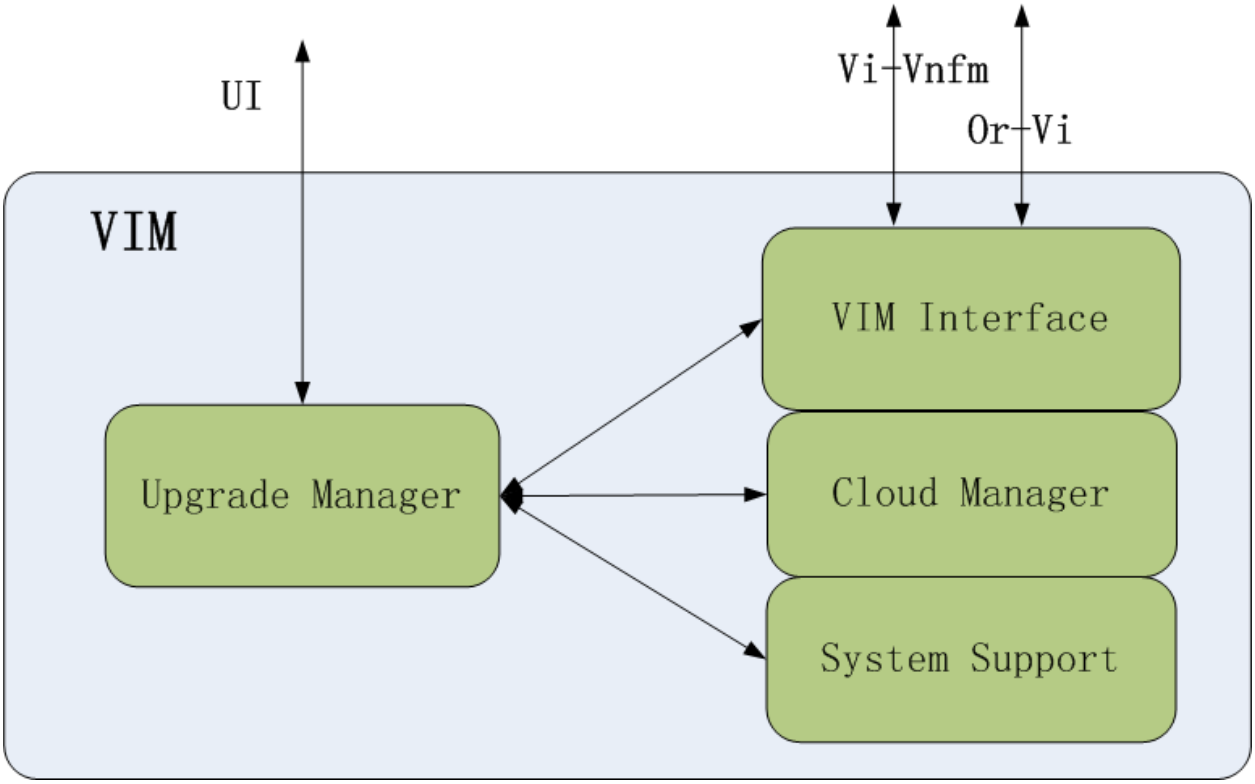
Mainly responsible for the external interface, include Vi-Vnfm, Or-Vi. This module stores VNFO and VNFM external information such as address and authentication.

- **Cloud Manager**:

Mainly responsible for virtualization resources management,which might be considered made up of Openstack and SDN control node.

- **System Support**:

This layer is the runtime support environment of upper layers, e.g. Cloud Manager and VIM interface., including:OS, HA, etc. To upgrade the upper software is based on this module.



CHAPTER

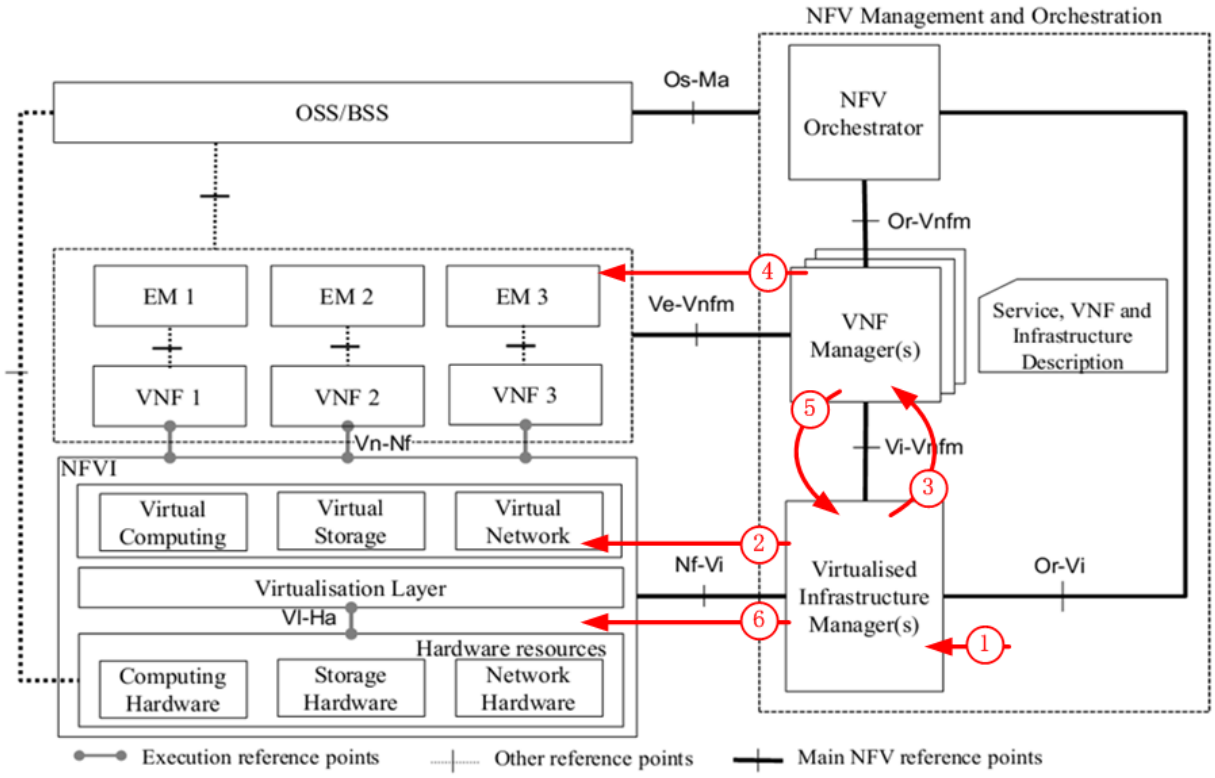
TWO

INFORMATION FLOWS

This section describes the information flows among the function entities when Escalator is in actions.

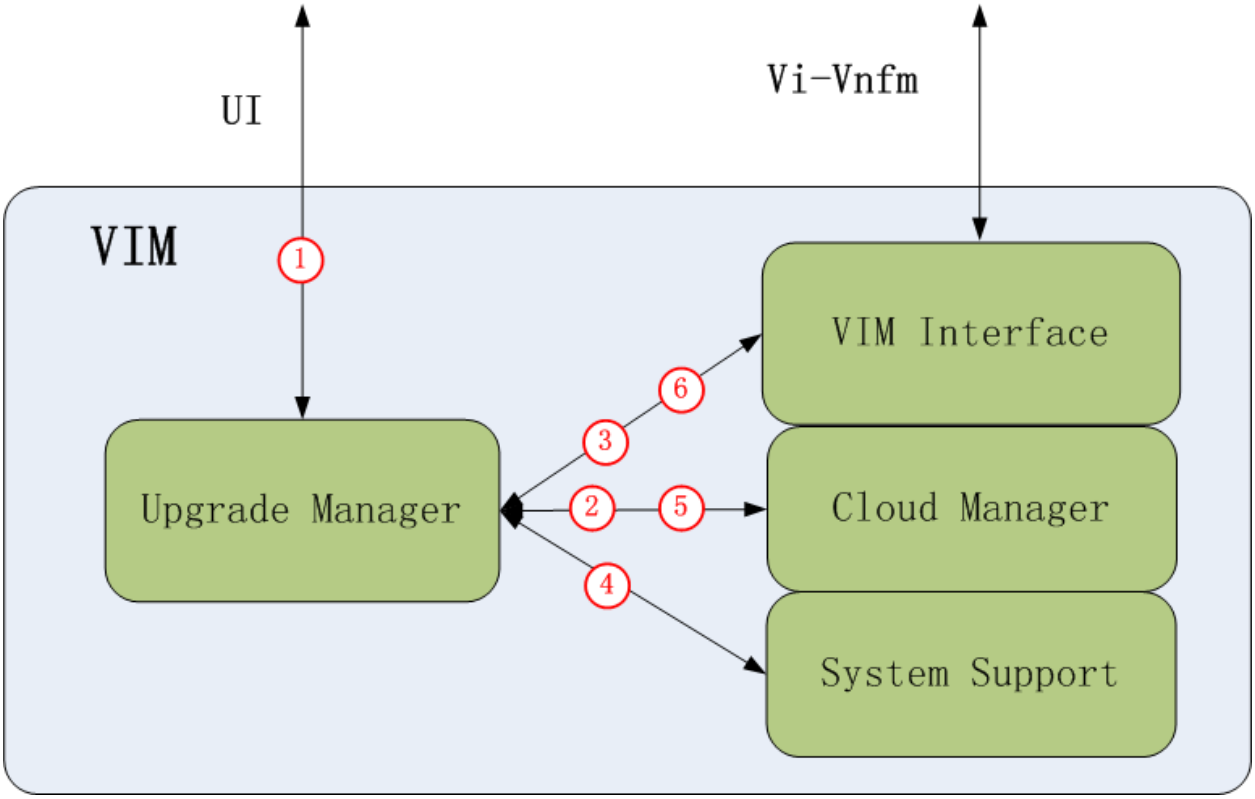
1. Upgrade process of Compute nodes

1.1 consider VIM as a whole

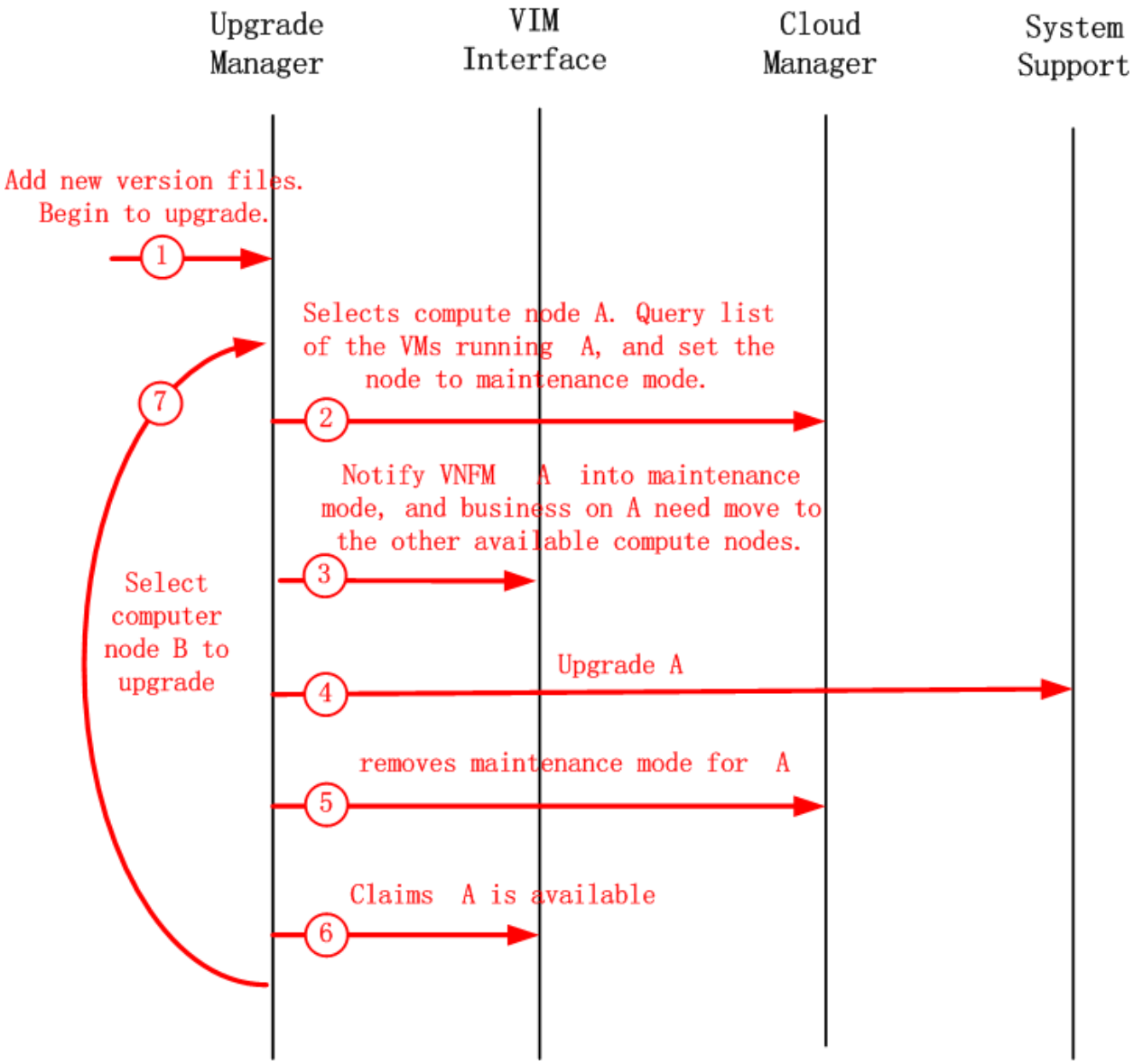


process is: 1. Operators add new version files on the VIM,initiate the upgrade. 2. VIM chooses some compute nodes as the upgrade target nodes, and set them into maintenance mode. VIM queries the list of running VMs on target nodes. 3. VIM notice VNFM corresponding to the virtual machine, to migrate the business. 4. VNFM migrates the business. If the business is in active of active-standby mode, it will initiate switch-over. If the business is in loading balance mode, it will move the business to other node. 5. After VNFM moves business, it notifies the VIM. 6. VIM judges whether the business on the target VM has all been moved. If not, VIM migrates the VM with business loaded to other free nodes. Then VIM upgrades the target computer nodes. After upgrade, VIM set the target compute nodes into normal nodes. 7. If there are computer nodes remained to be upgraded, goto step 2.

4.2 from inside VIM



process is: 1. Upgrade manager receives user operation commands. Add new version files. Upgrade is began. 2. Upgrade Manager selects compute node A to Upgrade. Query list of the VMs running the compute nodes A to the Cloud Manager, and set the node to maintenance mode, that is to say creation or migration of new VM on this node is impossible anymore. 3. Upgrade Manager notifies VNFM compute node A into maintenance mode by VIM interface, temporarily disabling the inserting of business, and business on compute node A need move to the other available compute nodes. 4. When receives the VNFM reply, or waited for a timeout, Upgrade Manager notifies the system support on compute node A to do software upgrade. 5. After upgraded, Upgrade Manager removes maintenance mode for the compute node A. 6. Upgrade Manager claims VNFM computing nodes A available. 7. Select computer node B to upgrade


Upgrade Manager
VIM Interface
Cloud Manager
System Support
Add new version files.
Begin to upgrade.
1
Selects compute node A. Query list of the VMs running A, and set the node to maintenance mode.
2
Notify VNFM A into maintenance mode, and business on A need move to the other available compute nodes.
3
Select computer node B to upgrade
7
Upgrade A
4
removes maintenance mode for A
5
Claims A is available
6

CHAPTER

THREE

ADMINISTRATIVE INTERFACES

This section describes the required administrative interfaces of Escalator.

3.1 CLI Interface

This section describes CLI of Escalator.

3.2 RESTful API

This section describes the API of Escalator for developer.

CHAPTER

FOUR

CONFIGURATION AND LOGGING

This section describes the required configuration and logging of Escalator.

4.1 Configuration Format

This section will suggest a format of the configuration files and how to deal with it.

4.2 Logging Format

This section will suggest a format of the log files and how to deal with it.

- search

Revision:

Build date: March 03, 2016