Project Proposal: "DOCTOR" (NFVI Fault Management/Maintenance)

Contact: Ashiq Khan
Nov. 25, 2014

Problem description

- Virtualised Infrastructure Manager (VIM) e.g. OpenStack cannot detect certain Network Functions Virtualisation Infrastructure (NFVI) i.e. Resource Pool faults, which is necessary to detect and notify in order to ensure the proper functioning of EPC VNFs e.g. MME, S/P-GW.

  - EPC VNFs are often in ACT-SBY configuration and need to switch to SBY as soon as relevant faults are detected
  - NFVI encompasses Physical Machines, Hypervisors, Storage and Network elements

- In addition, VIM e.g. OpenStack needs to receive maintenance instruction from the operator/administrator

  - Empty certain Physical Machines (PMs) so that maintenance works could be performed

Features required

- OpenStack shall be able to collect certain fault information about the elements in its resource pool

- OpenStack shall be able to inform the users/client whose VMs are affected by the resource pool faults

- OpenStack shall be able to receive maintenance instruction (p8) for the elements in its resource pool

OPNFV

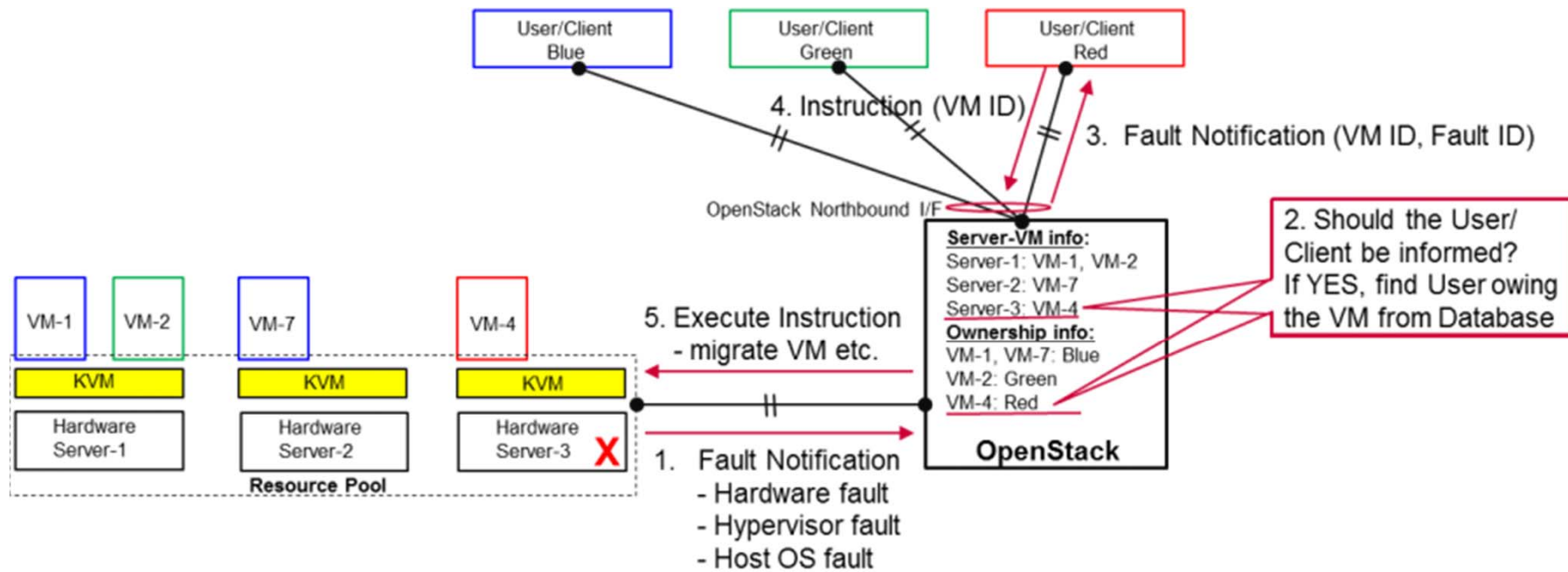# Use-Case 1: Fault Management
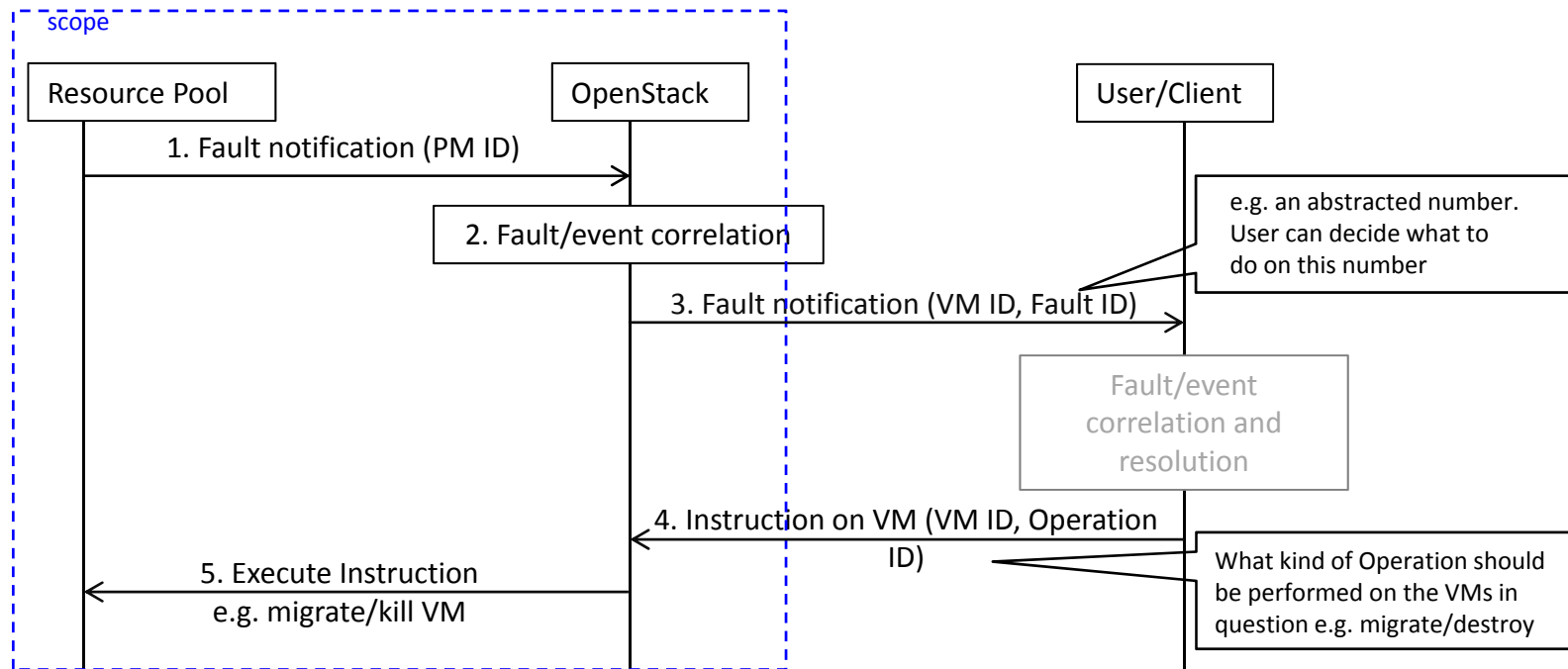

Fig. 1: Steps in Fault Management

https://wiki.opnfv.org/requirements_projects/doctor_fault_mangement_and_maintenance

High-level message flow   Fault mangement

- Scope is OpenStack, internal APIs to resource pool and the norhtbound I/F to be exposed from OpenStack

Internal feature: Fault detection (1,2) ; northbound I/F: two messaging (3,4)
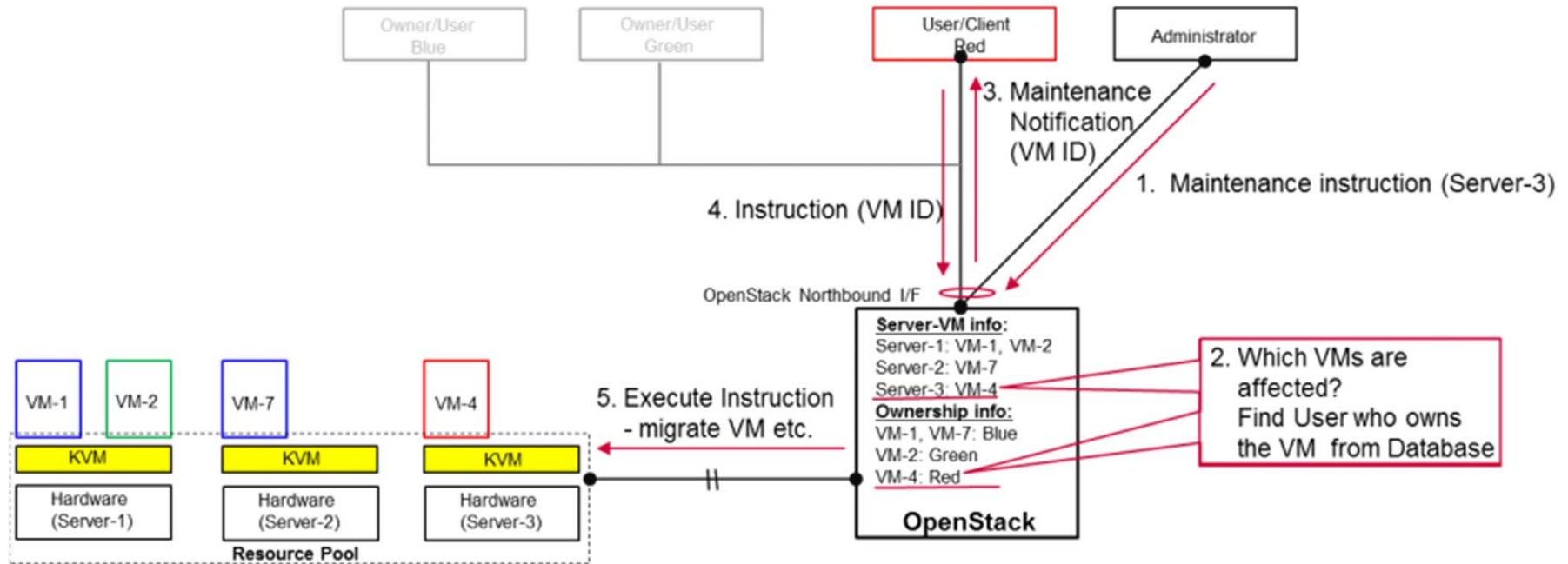
Use-Case 2: Maintenance

Fig. 2: Resource pool maintenance

Project DOCTOR

High-level message flow   NFVI Maintenance

- Only add #1 on the Fault Management (P4, 5)
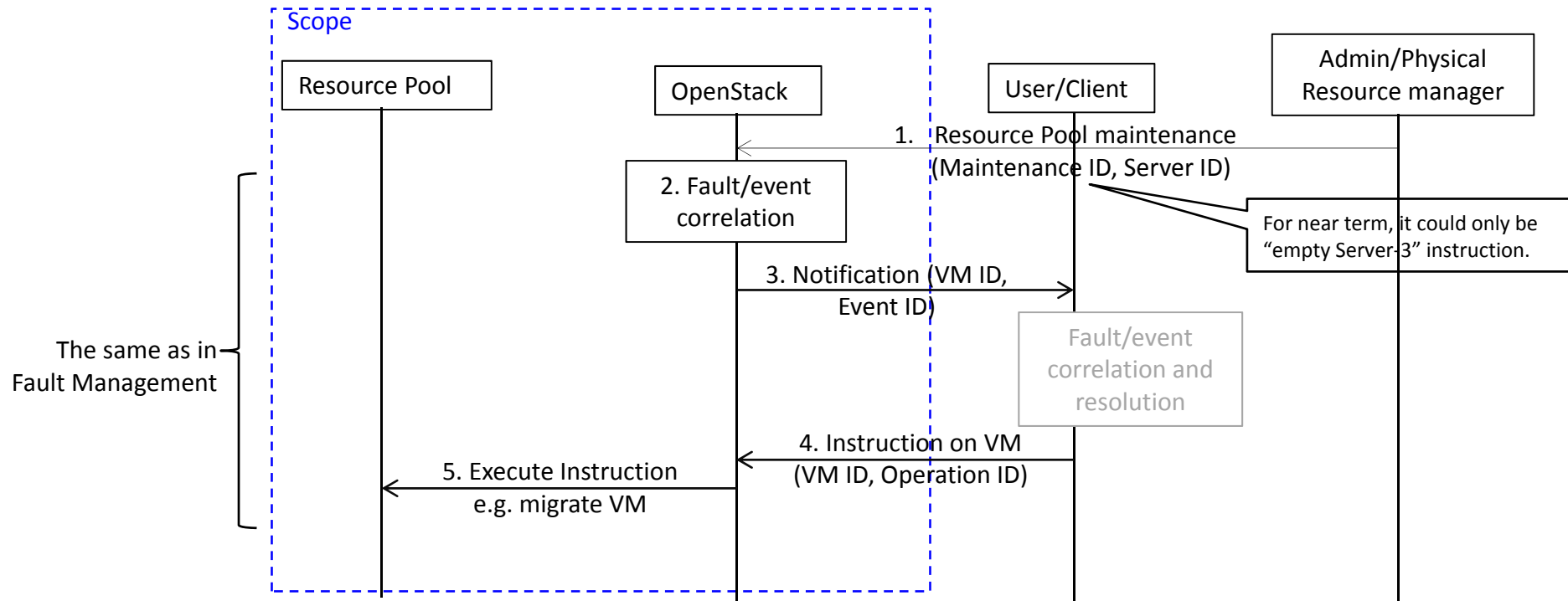
Internal feature: event correlation (2) ; northbound I/F: three messaging (1,3,4)
The objective is to free up the physical server prior to maintenance

Focus Area in Initial Development at VIM

Receive fault or maintenance message
regarding physical resource

Identify affected logical resources

Find user's notification policy and receiver

Send out fault notification

Receive instruction/ACK on fault recovery

OPNFV
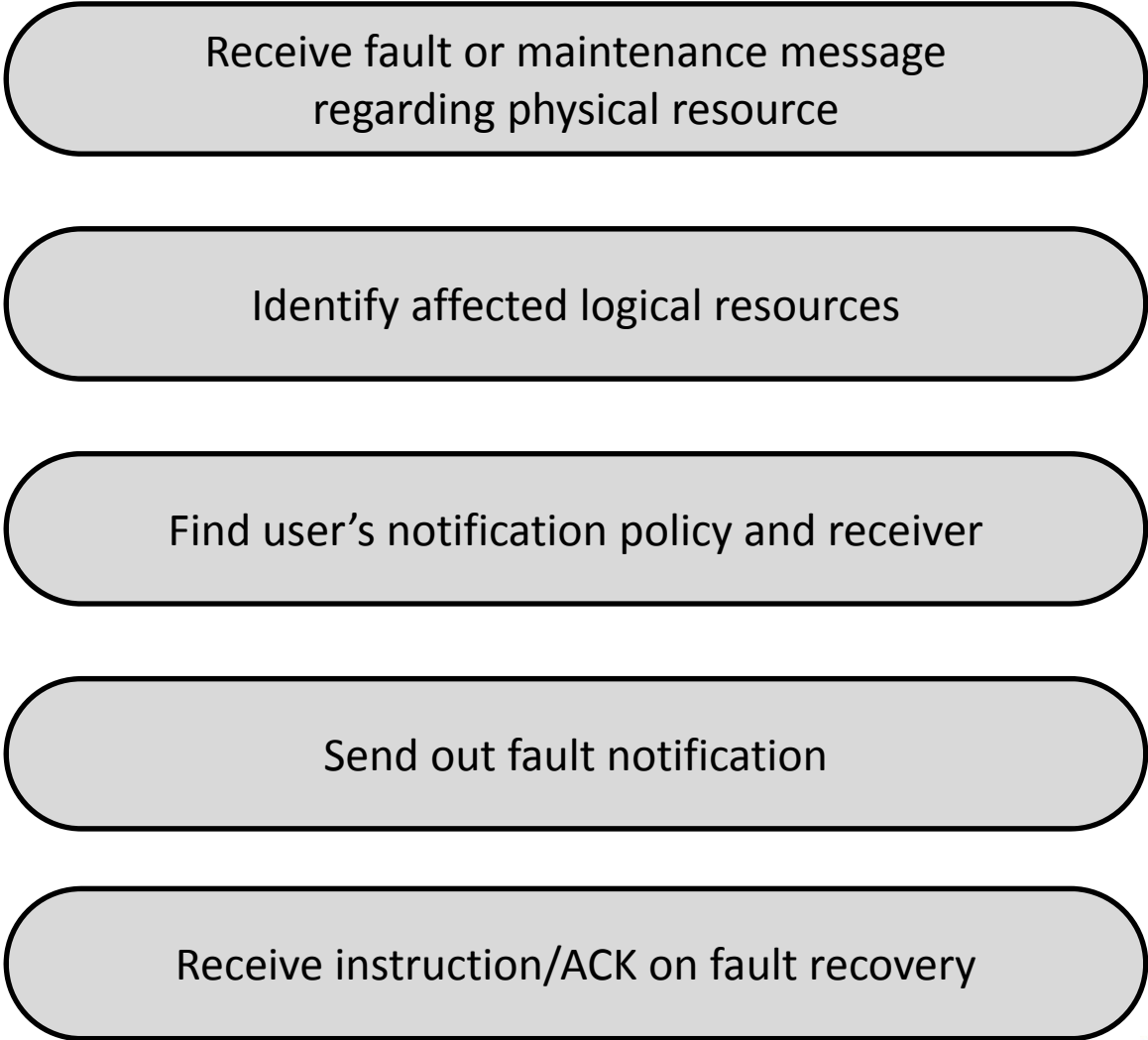
Scope

- OpenStack as VIM

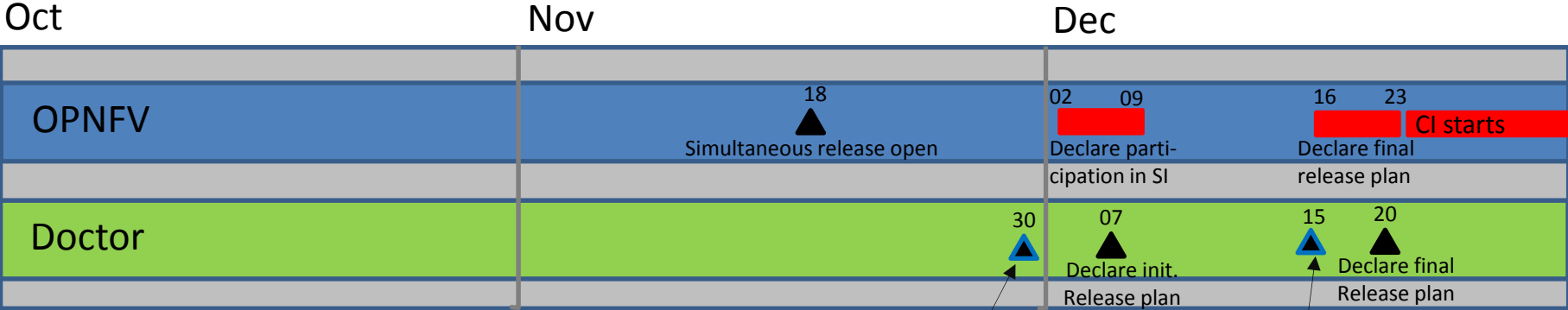- Fault items to be detected

  – Hardware faults to start with

- Fault detection mechanism in between OpenStack and the resource pool

- Northbound I/Fs of VIM

- Extensibility

  – Other NFVI elements
    - KVM, ODL
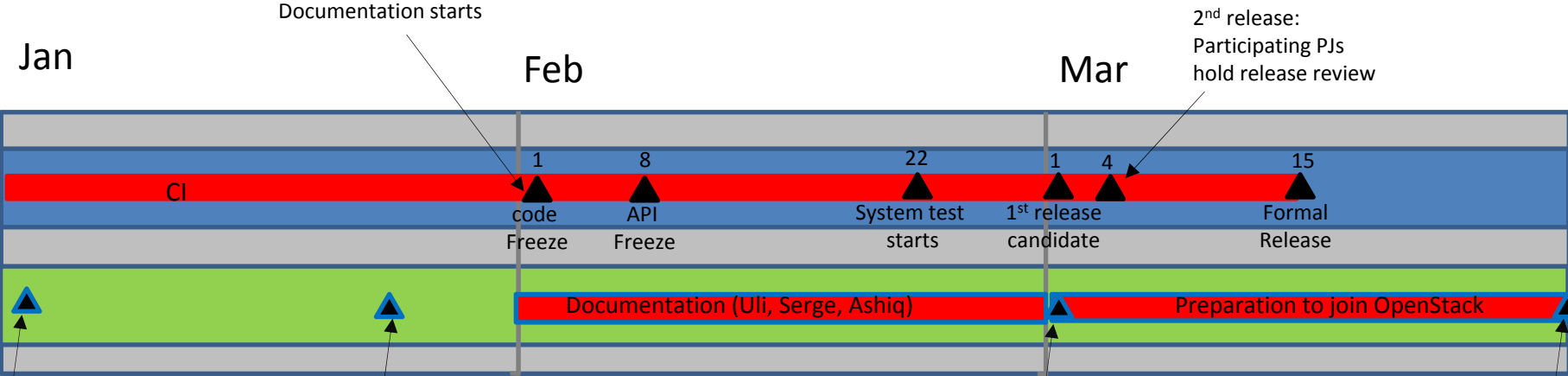
Deliverable (by Mar 2015)

- Documents

  - VIM northbound I/F specifications for fault management and NFVI maintenance
  - Implementation architecture and plan in OpenStack

- For implementation work (collaborative development) in OpenStack

  - Blueprint would be submitted for L-release
  - Blueprint could already be submitted for Kilo based on progress

Timeline of Doctor Project

	Oct	Nov	Dec
OPNFV		18 ▲ Simultaneous release open	02 09 [CI] Declare participation in SI 16 23 [CI] CI starts Declare final release plan
Doctor		30 ▲	07 ▲ Declare init. Release plan 15 20 ▲ ▲ Declare final Release plan

Gap Analysis (OS)
Determine imple-menation scenario

First Spec. ready
- NB I/Fs
- Messaging
- Information elements
- Implementation plan

Documentation starts

2nd release:
Participating PJs
hold release review

	Jan	Feb	Mar
	CI ▲	1 ▲ 8 ▲ code API Freeze Freeze	22 ▲ 1 4 ▲ ▲ 15 ▲ System test 1st release Formal starts candidate Release
	▲ ▲	Documentation (Uli, Serge, Ashiq)	▲ Preparation to join OpenStack ▲

Submit blueprint
to Kilo

2nd Spec ready
-refinement of the 1st

1st Blueprint for L-release
ready

2nd Blueprint for
L-release ready
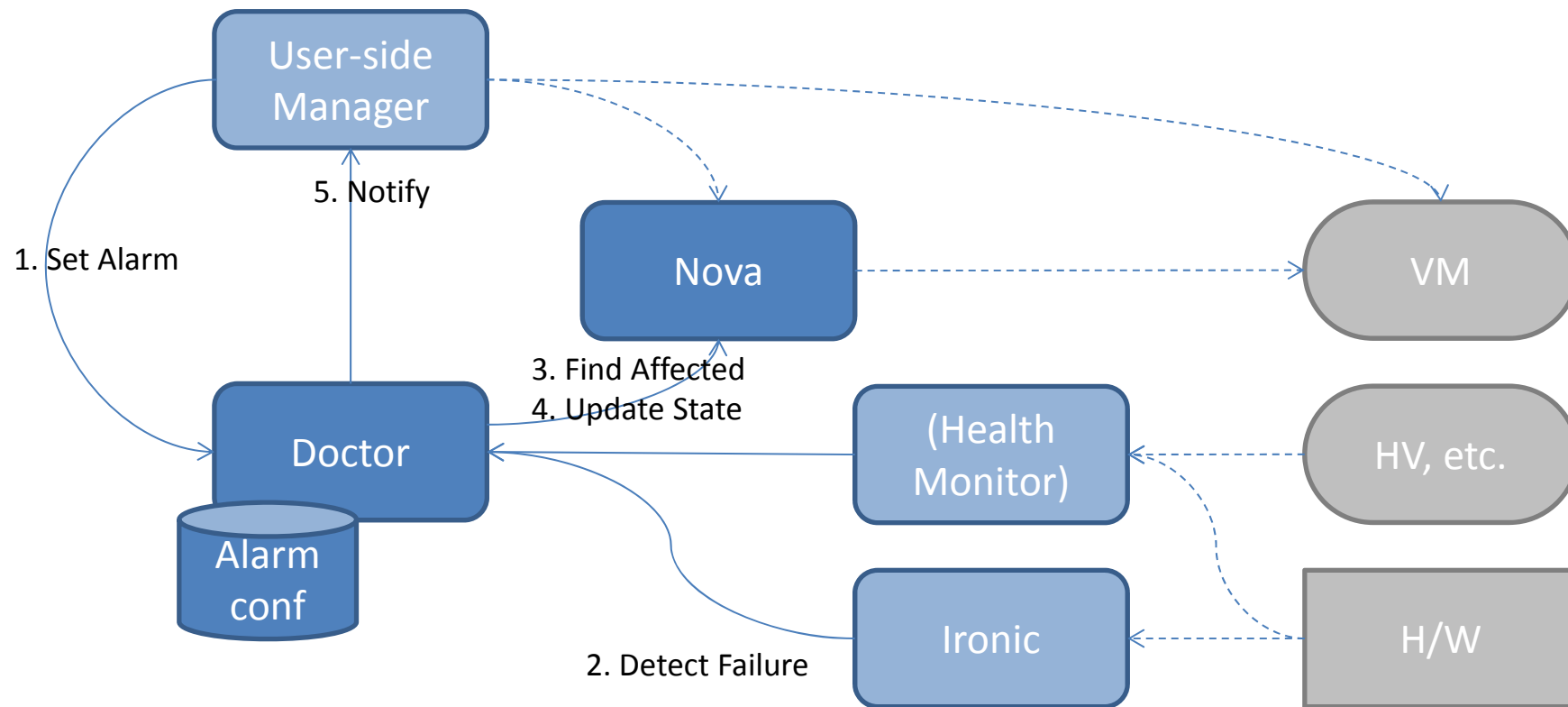
Resources

- Committers:

  - Technical: Ryota, Carlos, Tomi, Zhangyu, Palani and others
    - Responsible for the items shown in ▲
  - Documentation: Ashiq, Uli, Serge, Dirk

- Integration testing: would be determined later (after April 2015)

Appendix:

Infrastructure resources

- Mailing list
- Etherpad
- IRC, MeetingBot
- Telco (GoToMeeting)
- Wiki:
- Code review : Gerrit (not necessary now)
- Code repository : Git repo (not necessary now)

Initial Architecture: tentative

Project DOCTOR

Initial Architecture: Alternative

1. Set Alarm

5. Notify

3. Find Affected
4. Update State

2. Fault Injection

User-side Manager

Alarm conf

Nova

VM

(Health Monitor)

HV, etc.

Ironic

H/W