How to Work Upstream with OpenStack

Ashiq Khan
NTT DOCOMO, INC.

Ryota Mibu
NEC Corporation

Julien Danjou
Red Hat, Inc.

Contents

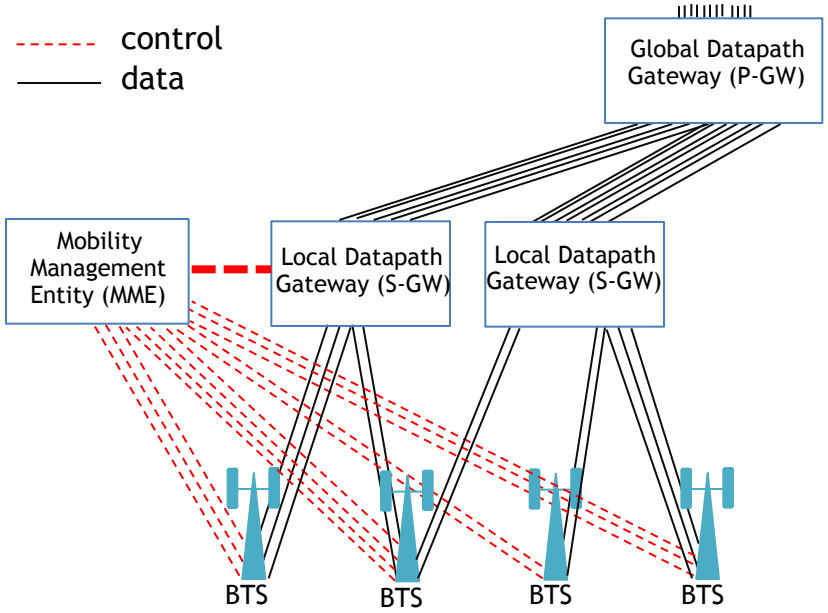
- Operator view
- Telecom vendor view
- Developer view

Operators position

- Convince the community about the use case
  - Get a crowd behind you
- Getting in touch with developers who will write the code
  - Start from your neighbors

Use case

- Improve visibility, clarify benefit



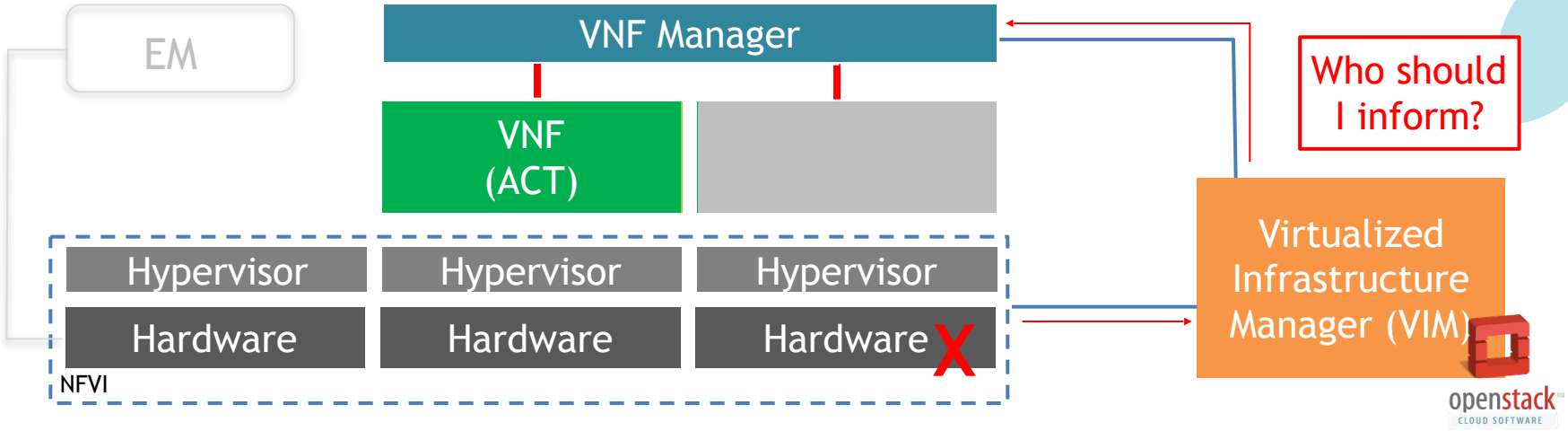
Each of these core nodes **hosts few thousands** subscriber **sessions**
- if down
  - **all mobile phones will be disconnected**
  - consequently will try to reconnect simultaneously
    - creating an '***Attach***' **storm**
      - leading to further congestion/failu

Failure recovery needs to be performed **in sub-second order**

Other operators need to understand that its of interest to them as well

Now, what do you actually need?

- Looking at OpenStack, what is missing?
  - Do gap analysis and define what exactly is needed

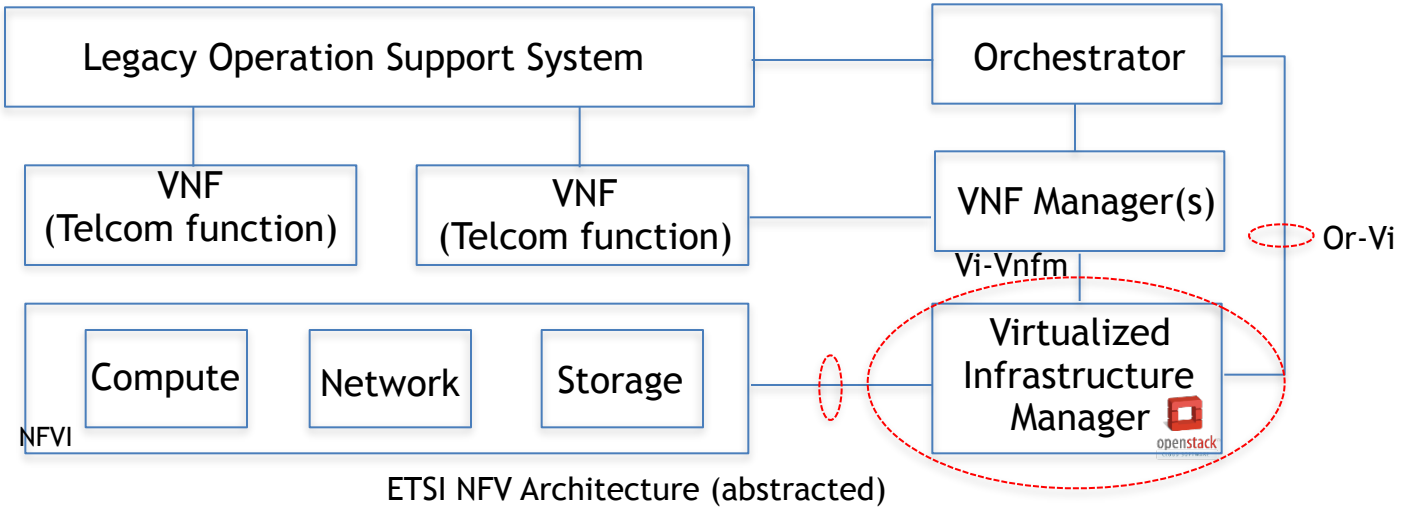


Upstream community requires to understand what exactly they need to develop

OPNFV

Scope your work

- Understand overall system architecture you are responsible for
  - New operational feature affects many other neighboring functional blocks

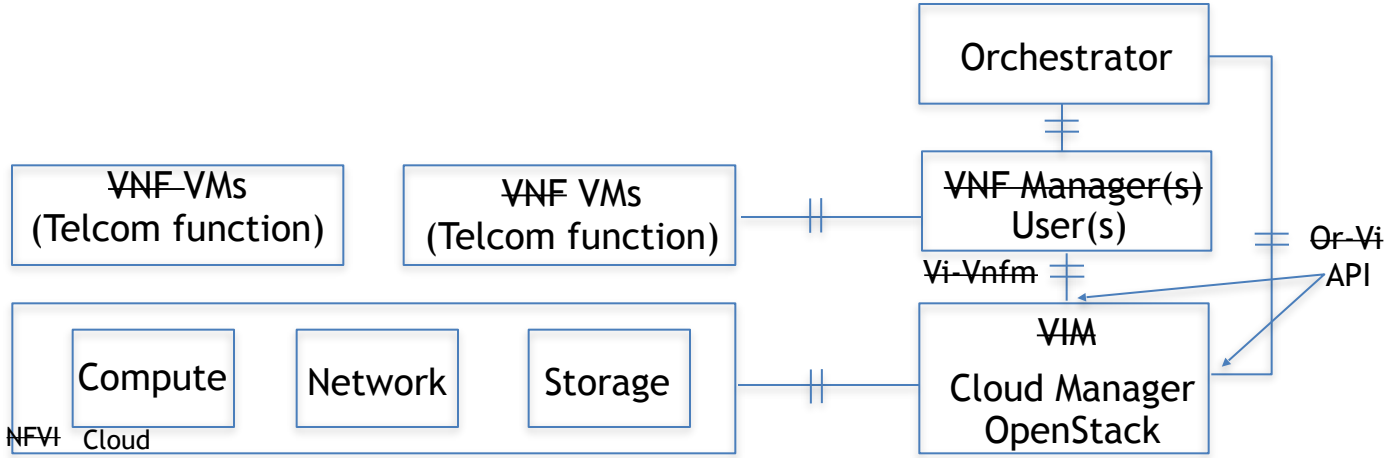


ETSI NFV Architecture (abstracted)

Develop your requirements taking a **reference system** into account, clarify the sco

Talk Open Source

- Translate your requirements into something comprehensible to the Open Source community



ETSI NFV Architecture (abstracted)

The upstream community need to understand the requirement in their own terms

Bring/find resources

- Different skill sets are necessary for a meaningful and successful feature development
- **Operators**: can give a practical requirement from day-to-day real network operation experience
- **Network architects**: provide solution architecture from an end-to-end perspective, check compliance with standards
- **Developers**: write codes
- **System integrators/vendors**: test/integrate a micro-feature into the system
- Operators: deploy into the operational network when requirements are met

For a successful feature development, you'll have to find all the above

Engage your upstream community

- Participate in related events
  - Find out who are interested
  - Feel the intensity of interest
- Reach out to developers
  - Know your key contacts
  - Invite to project break-out sessions
- Look for specialists in your ecosystem
  - Operator-vendor-developer….

A multi-party presentation in an OpenStack summit matters

Summary
Operator

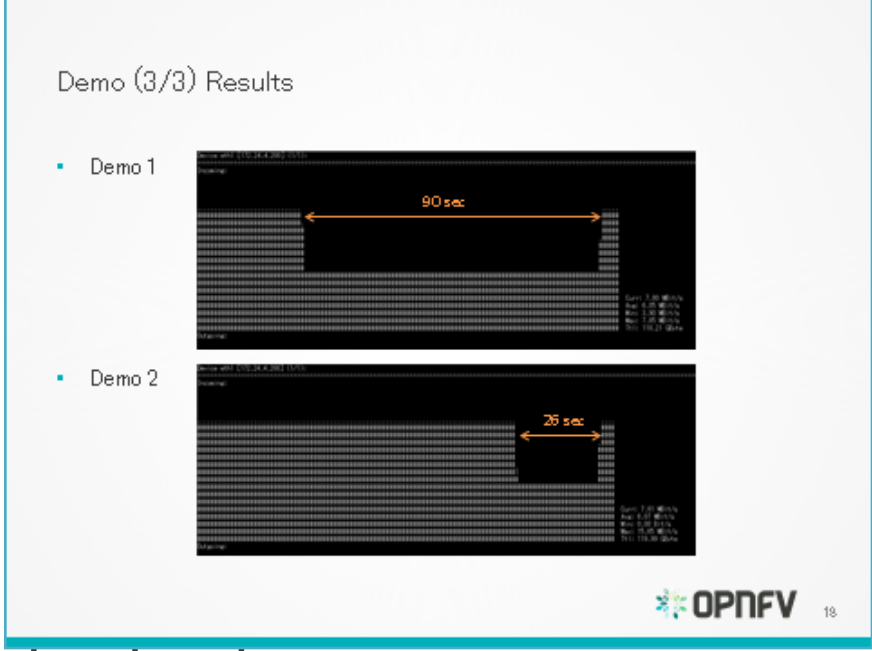
- From my experience in Open Source community as an Operator,
  - **Socializing is the key**
    - Talk to people, create a community, drive your surroundings
  - Have to have a decent crowd behind you
    - What is the interest in the user community
    - Leads to interest in the developers community
  - Keep your requirement simple
    - One step at a time
    - Care about negative impact on the existing code base
    - **You yourself need to participate**

Telecom Vendors view

1. Support operators to develop their desired features
   - You will see requirements and use cases
2. Interpret operator needs and developer idea back and forth
3. Integrate software components into huge telecom system
   - If there is a missing piece, just implement it
   - Respect industry specification and de-facto standard, especially I/Fs
   - Bring the PROBLEM first
     - Solution/Code comes later

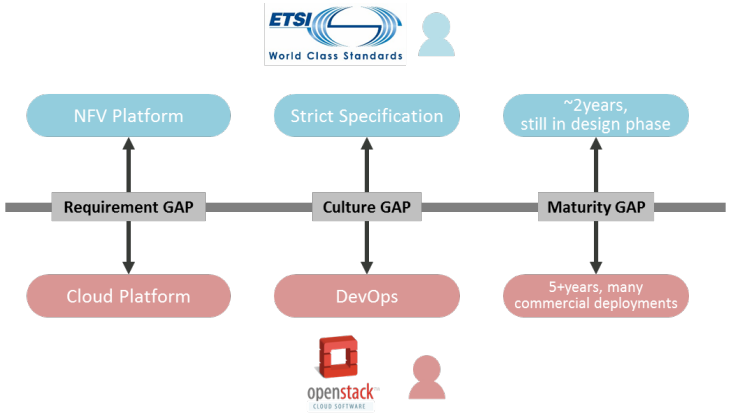
Bring the PROBLEM

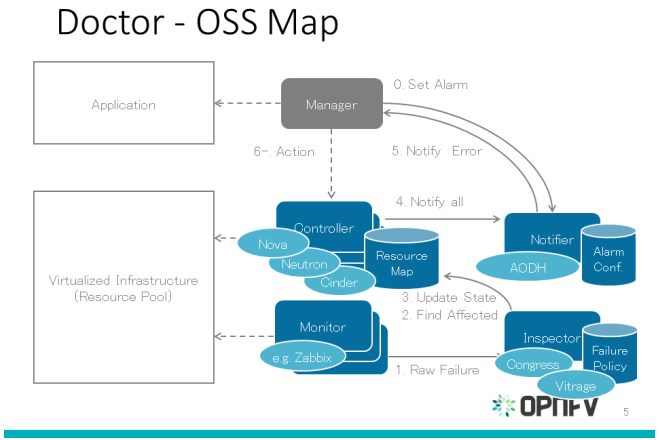
- Operators have clear use cases, but do not dictate a solution
  - Operators are generally solution independent
- Show what the PROBLEM is
- Communicate the PROBLEM with the developers
- Check with operators if the PROBLEM analysis is correct



Solution

- Find GAPs between Telco and Cloud with the developers
- Develop the SOLUTION with the developers

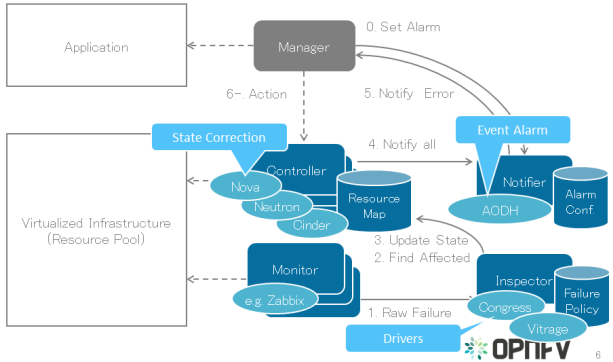




Jump into the upstream community

- Get support to elaborate HOW to solve in spec review
  - Do not change WHAT you need
  - Consider HOW to solve flexibly to find the best solution with them
- Learn the software and the project direc
- Find key developers in the specific area of the component
- Send your patch to the Master

Doctor - OSS Map + Collaboration



Telecom Vendors view (again)

1. Support operators to develop their desired features
   - You will see requirements and use cases
2. Interpret operator needs and developer idea back and forth
3. Integrate software components into huge telecom system
   - If there is a missing piece, just implement it
   - Respect industry specification and de-facto standard, especially I/Fs
   - Consider to contribute the IDEA you implemented (not CODE)

What is Upstream?

- Something you are relying on

- Component that can be shared for various purposes
  - The more a software get used, the more stable it becomes
  - A good developer knows, and even implements good library

The usual issues

- **Corporate** culture vs **opensource** culture
  - Meetings
  - Customers
  - Hierarchy
  - Cathedral vs Bazaar
  - Individuals vs Corporations
- Inability to understand the **model**
  - Or licenses, StackForge vs OpenStack
- **Ideas** without **resources** to offer
- Inability to explain the **ideas** and **business context**
  - *And sometimes for developers to understand them*

The good approach

- **Observe**
- **Learn**
- **Help** and be **constructive**
  - Resources
  - Testing
  - Bug report/fixing
- Build **trust**
- **Teach** your business and use cases
- **Generalize** your ideas
- **Split** the work in small tasks

- Train your **engineers** to **open source** too
  - http://docs.openstack.org/upstream-training/
  - Make them understand the **culture**
- Do not **antagonize**
- Show **humility**
- There's not only the **PTL**
- Find alternate **routes**

*Open Source developers: **open** your project*

(See https://julien.danjou.info/blog/2016/foss-projects-management-bad-practice)

Thank You