Look Inside.™

VSM Open Source Project
Roadmap and processes - The Way Forward

Wang, Yaguang

Ferber, Dan

March 2015

Agenda

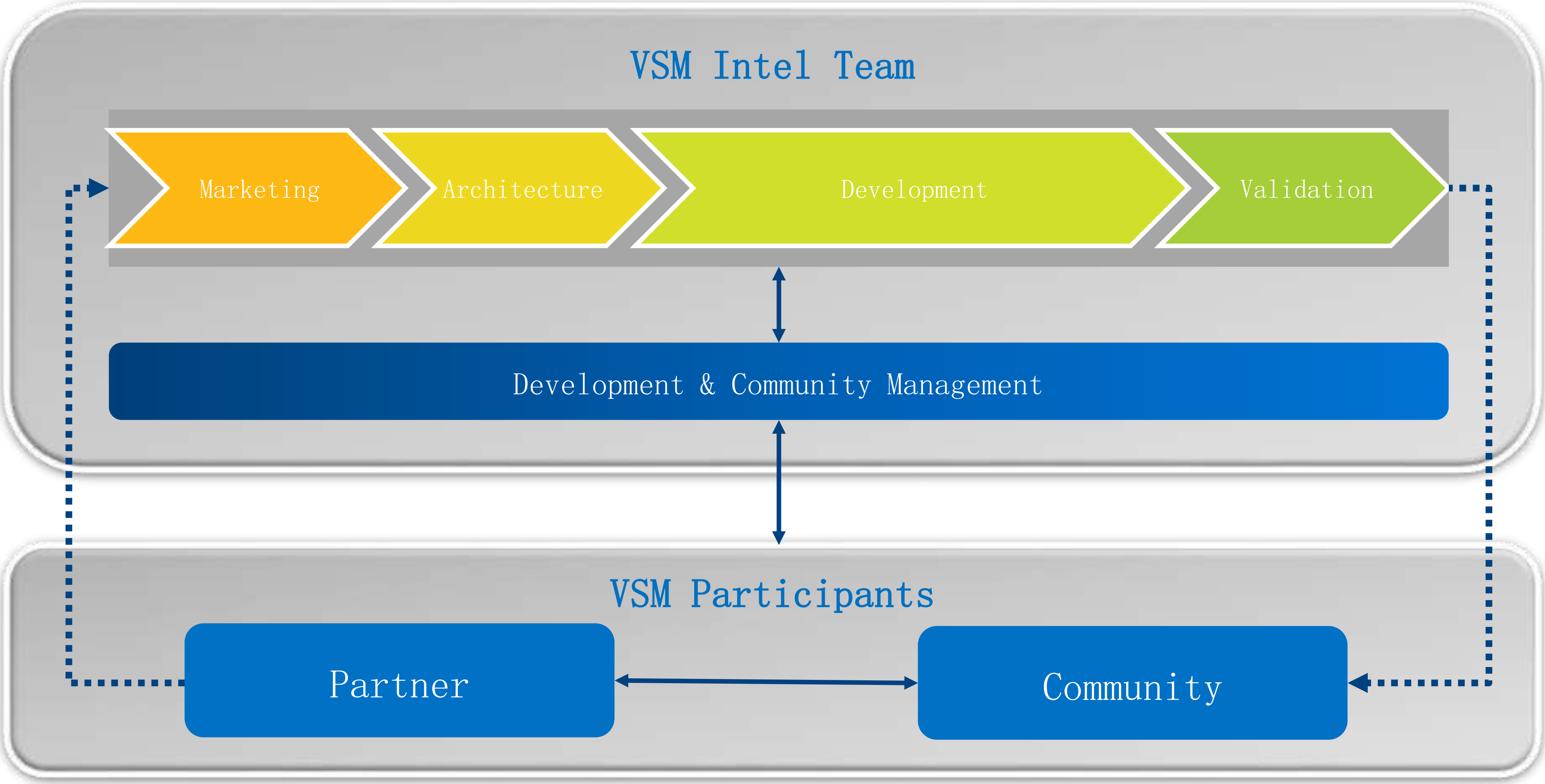
- Meet the VSM Team
- Roadmap
  - Strategic priorities
  - Release Plan
- Open source processes
  - Where is the code, how is it organized and managed
  - Where to report bugs or request features
  - How to participate

Meet the VSM Team

VSM Intel team
- Focus

- Fixes bugs and develops features based on community priorities
- Publishes plans on roadmap pages
- Tests and packages engineering and production releases
- Does development in the open on github (starting with 1.0)
- Manages public bug/feature Jira tickets, and public mail list
- Welcomes community developers to fix bugs and develop features
- Holds periodic community meetings

Meet The VSM Team
- Structure



VSM Intel Team

Development & Community Management

VSM Participants

Meet The VSM Team
- Staffing

- Wang, Yaguang (Yaguang.wang@intel.com)
  - Development Management, Gatekeeper

- Ferber, Dan (Ferber.dan@intel.com)
  - Community Management, Marketing

- 1 QA + 3 developers

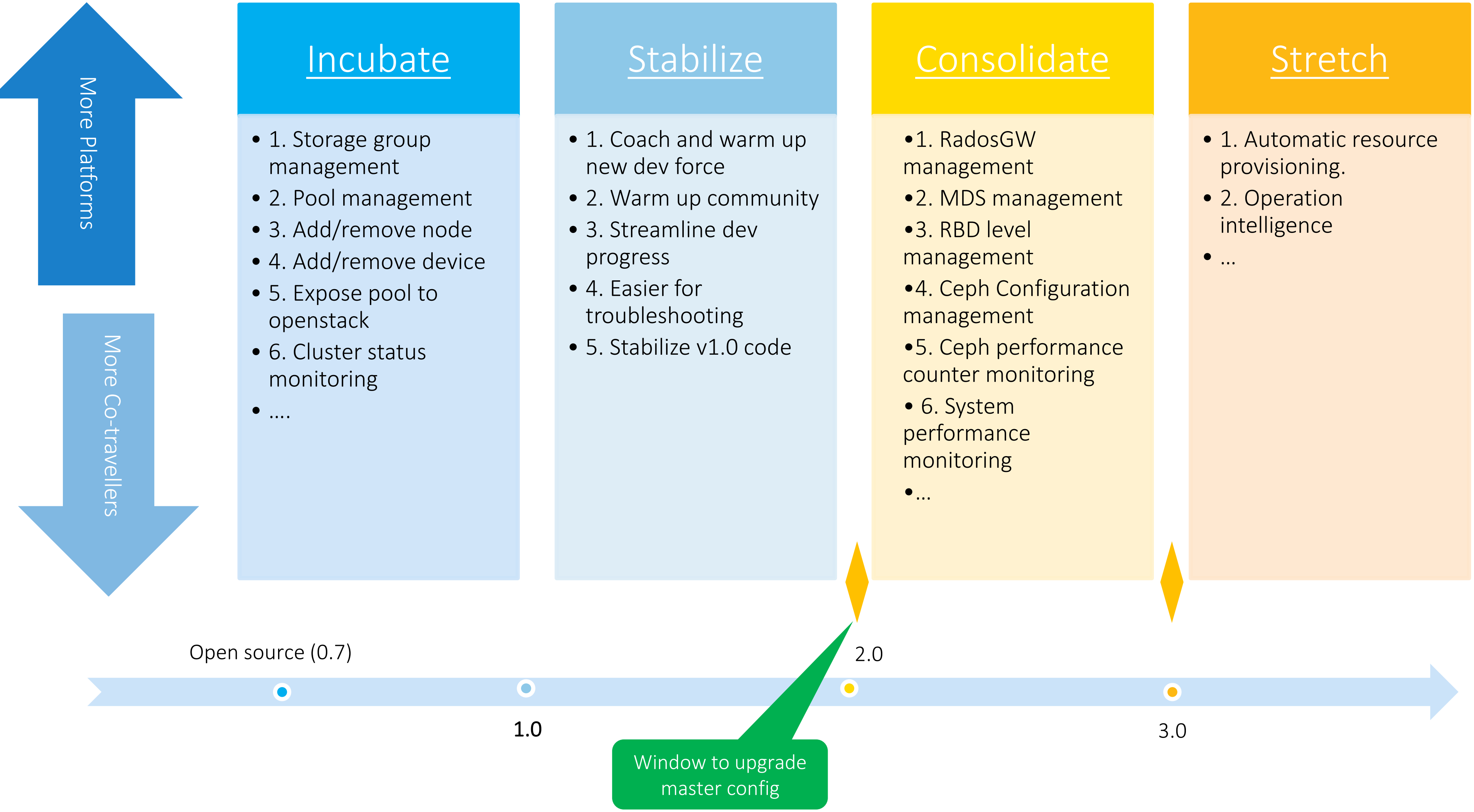
- *Note: we are maintaining a mailing list and JIRA system, they are the preferred approaches to contact engineering.*

Roadmap

VSM Strategic Priorities

- For 2015 development, we will focus on VSM features around **the management of Ceph**.
- We prefer to select and test on one **master config** (currently CentOS 6.5 + Ceph Firefly + Openstack H/I):
  - This is because in general our partners use VSM/Ceph clusters as a “Black box” or “solution” cluster, and then Ceph clients on other OS environments, and/or many OpenStack environments – can use the Ceph storage.
  - We expect community could help with other environments, such as SuSE and Ubuntu. In very rare case, we may touch other config for technical evaluation.
- Generally, **management functionality will have a higher priority** than supporting lots of environments.
- The Intel team will generally give priority (for what Intel engineers work on) to requests made by partners who are using VSM to deploy VSM/Ceph based solutions. Work from other community members is up to the member doing the work, and Intel will land community code when it is complete, tested, submitted, and reviewed.

VSM Roadmap– History and Future



Incubate

- 1. Storage group management
- 2. Pool management
- 3. Add/remove node
- 4. Add/remove device
- 5. Expose pool to openstack
- 6. Cluster status monitoring
- ….

Stabilize

- 1. Coach and warm up new dev force
- 2. Warm up community
- 3. Streamline dev progress
- 4. Easier for troubleshooting
- 5. Stabilize v1.0 code

Consolidate

- 1. RadosGW management
- 2. MDS management
- 3. RBD level management
- 4. Ceph Configuration management
- 5. Ceph performance counter monitoring
- 6. System performance monitoring
- …

Stretch

- 1. Automatic resource provisioning.
- 2. Operation intelligence
- …

Window to upgrade master config

V1.0

- **Starting from this release, we will go to open source project way for feature requests, bug fixing. It follows "giving more, getting more", if you want to have more features or bug fixes, the best way is to increase the credit by contributing, hence influence directions and priorities. Contributions could be code/documenting/mailing list Q&A, and code is weighted.**

- V1.0 will be a long term release, our team will work with FJ to work through critical bugs.
- V1.0 and v1.1 are branched from the same code base, all change set on v1.0 will be merged into v1.1.
- To keep v1.0 stable, we will not merge major fixes or features from v1.1 to v1.0.

V1.1: Feature list

For each release starting from v1.1, the work item list will include two parts:

i) JIRA feature list: this is the major area dev team will work on and try to grantee.

ii) New feature incubation: this is to evaluate some new risky features as input for next release, occasionally, they could be included in current release.

iii) Optional JIRA issue list: it’s really floating, dev team will allocate them depending on urgency, importance and dev load.

- This is a warm-up release to help warm up development team to familiarize open source processes, and further consolidate v1.0.
- No major features are granted to be added
- Majority of the work will be bug fixing
- Work in making VSM easy to support and troubleshoot
- Begin new feature incubation for subsequent releases

V1.1: New Feature Incubation

Beside those listed JIRA issues, in v1.1 development period, we will optionally start some new feature evaluation work:

- Add sanity check tool (VSM-156) [done]
- Add issue reporter tool (VSM-159) [done]
- Work with existing Ceph cluster (VSM-53) [pending]
- VSM auto deploy (VSM-184) [done]
- centos 7 support (VSM-135) [wip]
- multiple network supporting (VSM-202) [wip]

V2.0: Feature list

Although the strategic priority is on ceph management, we heard many requests about master configs. After all current master configs (centos 6.5 + ceph firefly + openstack Havana/Icehouse) seems a bit out of date.

To answer those requests, we will upgrade the master configs in v2.0, and to make sure people can get the upgrade earlier, the feature set in v2.0 will keep small, target date is in the Mid of the summer. The proposed coverage will be:

i) centos 7 support (VSM-135)

- try to use packages from DVD or EPEL as possible

ii) ceph hammer support (VSM-212)

- hammer is preferred as it’s LTS

iii) openstack juno support (VSM-130)

iv) upgrade openstack dependencies to juno (VSM-221)

v) multiple network supporting (VSM-202)

V2.0: New Feature Incubation

Meanwhile, we will do below new Feature incubation:

i) Visual dashboard (VSM-199)

ii) Ceph performance counters monitoring (VSM-220)

iii) OSD disk health status check (VSM-185)

iv) Ubuntu support (VSM-134)

v) Work with existing Ceph cluster (VSM-53)

We will also hear feedback from community in the following two weeks to help fine tune the coverage in v2.0.

V2.0+: General Information

As there is still a long feature list needs to fulfill, it’s expected to have a series of releases after v2.0. For those releases, candidate features are:

- Enhance Ceph Management Core Functionalities

  RadosGW management (Object, VSM-80)

  MDS management (File, VSM-158)

  RBD enhancement (Block, VSM-157)
- Ceph Configuration Management (VSM-94)
- Performance monitoring
  - ceph performance counters monitoring (VSM-220)
  - system performance monitoring like CPU, IO, Network...(optional, VSM-92)
- Work with existing Ceph cluster (VSM-53)
- Visual dashboard (VSM-199)
- Add disks to cluster (VSM-190)

We expect to hear feedback from community in the following two weeks to help fine tune the coverage in v2.0.

Open source Vehicles

VSM Open Source

- VSM is still young in open source community, and we are listening voices from community to help improve it.
- We intend to make all open source VSM feature proposals public, and do all development publicly.
- VSM is in active development, and we leverage a few vehicles on managing the project.

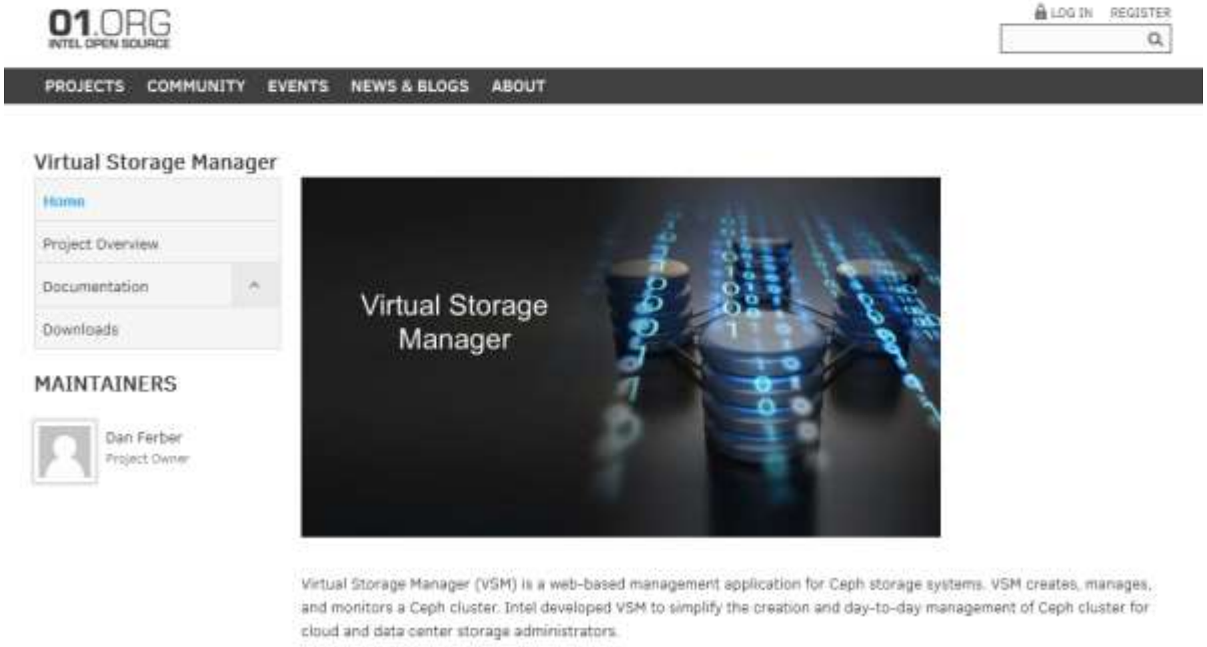
Preparations

Accounts:

1. One github account (http://github.com)
2. One 01.org account (http://01.org)
3. One Nabble account (http://nabble.com)

Open source Project Vehicles

- Home page:
  - https://01.org/virtual-storage-manager
  - The pointer to other VSM related resources like Github and JIRA
  - **Documents (doc, ppt, etc.)**
  - Major releases



Virtual Storage Manager

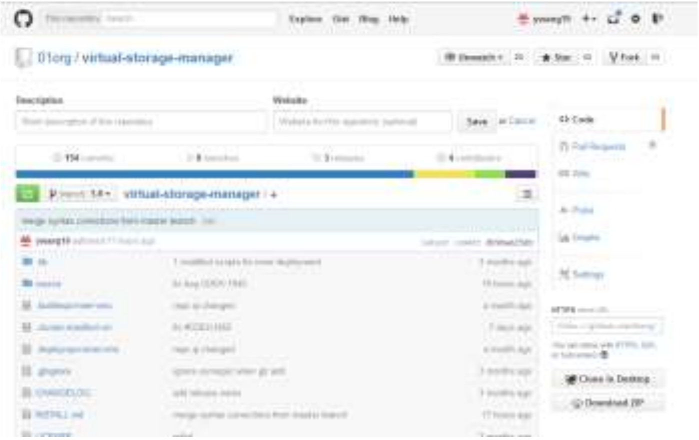
MAINTAINERS

Dan Ferber
Project Owner

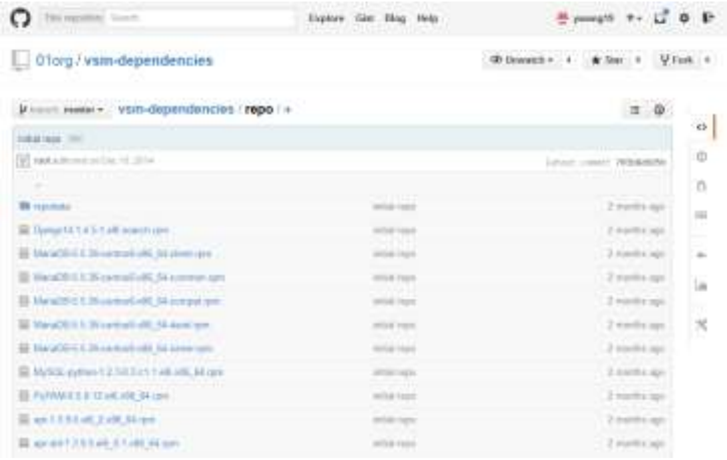
Virtual Storage Manager (VSM) is a web-based management application for Ceph storage systems. VSM creates, manages, and monitors a Ceph cluster. Intel developed VSM to simplify the creation and day-to-day management of Ceph cluster for cloud and data center storage administrators.

Open source Project Vehicles
- see “work with Github” section

- Code Repository:
  - https://github.com/01org/virtual-storage-manager
    - **Source code**
    - Installation instructions (INSTALL.md)



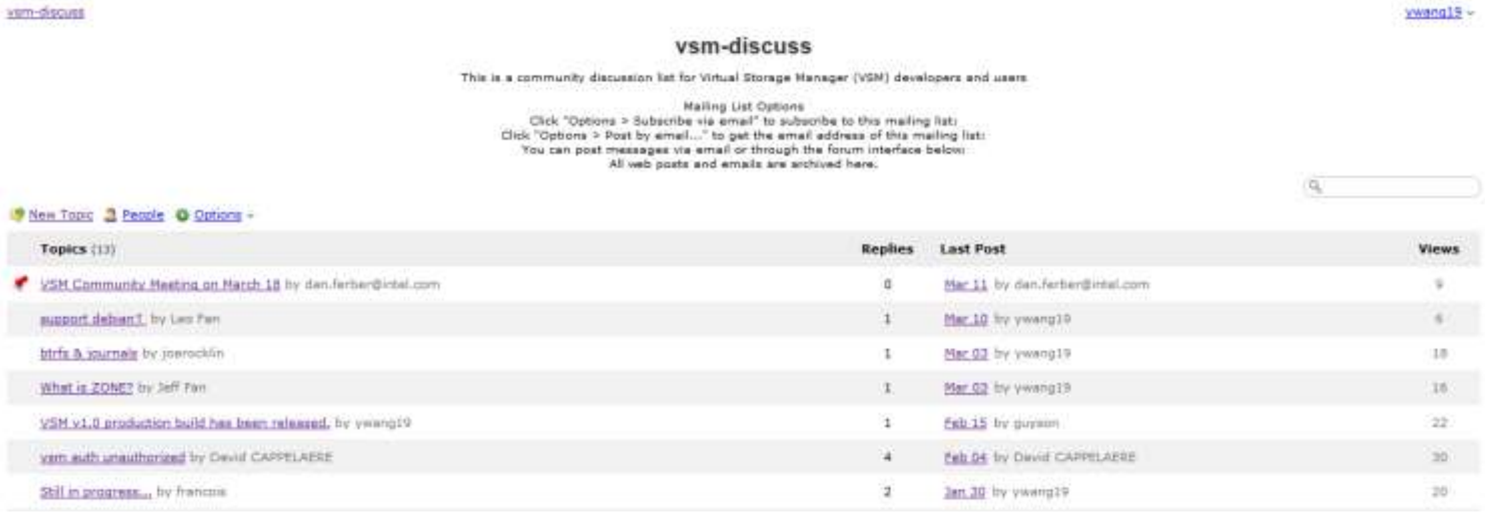
- Dependencies Repository:
  - https://github.com/01org/vsm-dependencies
    - **Dependent packages binary**
    - Taking in effect from v1.1.



Nabble

Open source Project Vehicles

- Mailing list:
  - http://vsm-discuss.33411.n7.nabble.com/
  - The preferred channel to connect Intel team with community, it’s also one source to collect requirements.



vsm-discuss

ywang19

vsm-discuss

This is a community discussion list for Virtual Storage Manager (VSM) developers and users

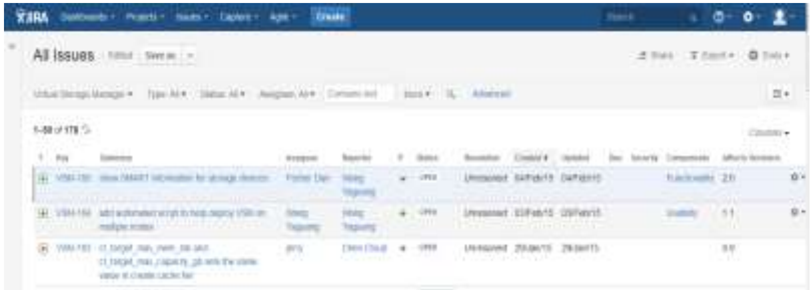
Mailing List Options
Click "Options > Subscribe via email" to subscribe to this mailing list;
Click "Options > Post by email..." to get the email address of this mailing list;
You can post messages via email or through the forum interface below;
All web posts and emails are archived here.

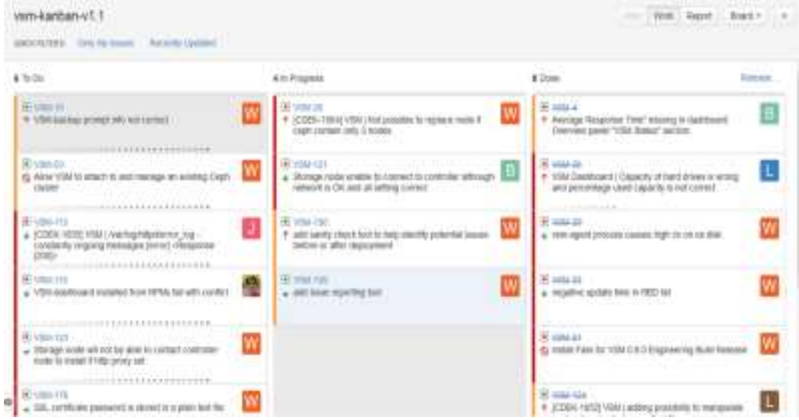
New Topic · People · Options

Topics (13)	Replies	Last Post	Views
VSM Community Meeting on March 18 by dan.ferber@intel.com	0	Mar 11 by dan.ferber@intel.com	9
support debian? by Leo Fan	1	Mar 10 by ywang19	6
btrfs & journals by joerocklin	1	Mar 03 by ywang19	18
What is ZONE? by Jeff Fan	1	Mar 03 by ywang19	16
VSM v1.0 production build has been released. by ywang19	1	Feb 15 by guyson	22
vsm auth unauthorized by David CAPPELAERE	4	Feb 04 by David CAPPELAERE	30
Still in progress... by francois	2	Jan 30 by ywang19	20

Open source Project Vehicles

- Bug tracker (JIRA):
  - https://01.org/jira/browse/VSM
  - Issues/features
  - Kanban





Work with Github

Preparations

Tools:

1. Download and Install Git Bash from: http://git-scm.com/downloads

2. Set up Git

```
git config --global user.name "Your Name Here“
git config --global user.email "your_email@example.com"
```

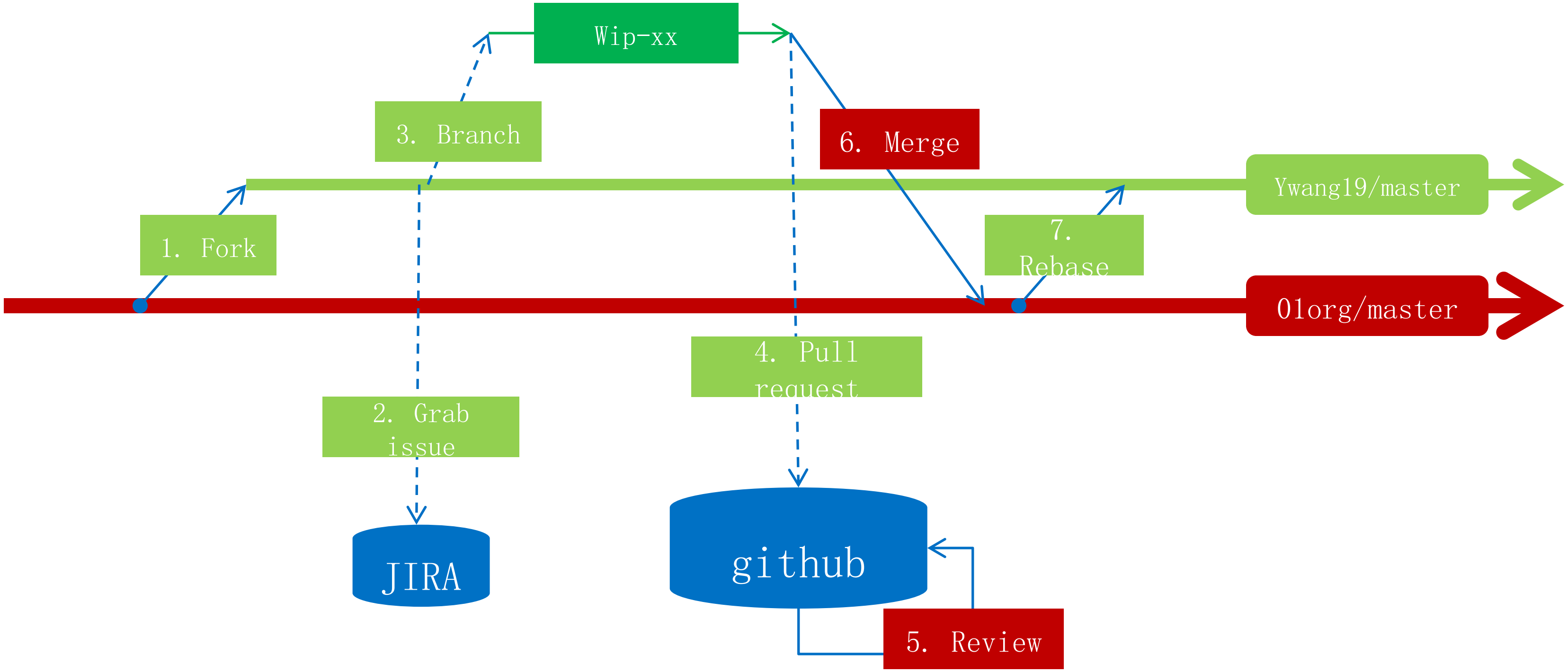
Contributing to VSM

Working Cycle

FPM: (Fork – Pull – Merge)

🟩 Contributor

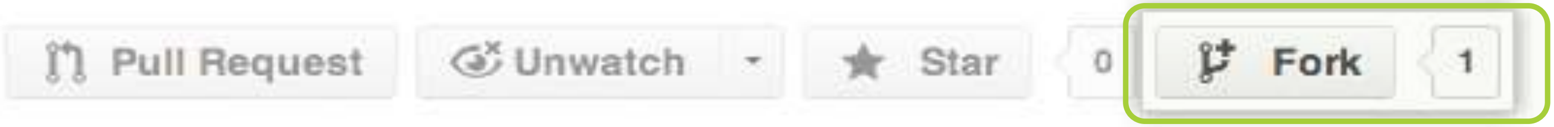
🟥 Gatekeeper



Contributing to VSM

1. Each contributor is suggested to fork from 01org\virtual-storage-manager repository on github, and work on its own repository.

- Fork repository (on github)



- Clone your repository (on local)

```
git clone https://github.com/ywang19/virtual-storage-manager.git
```

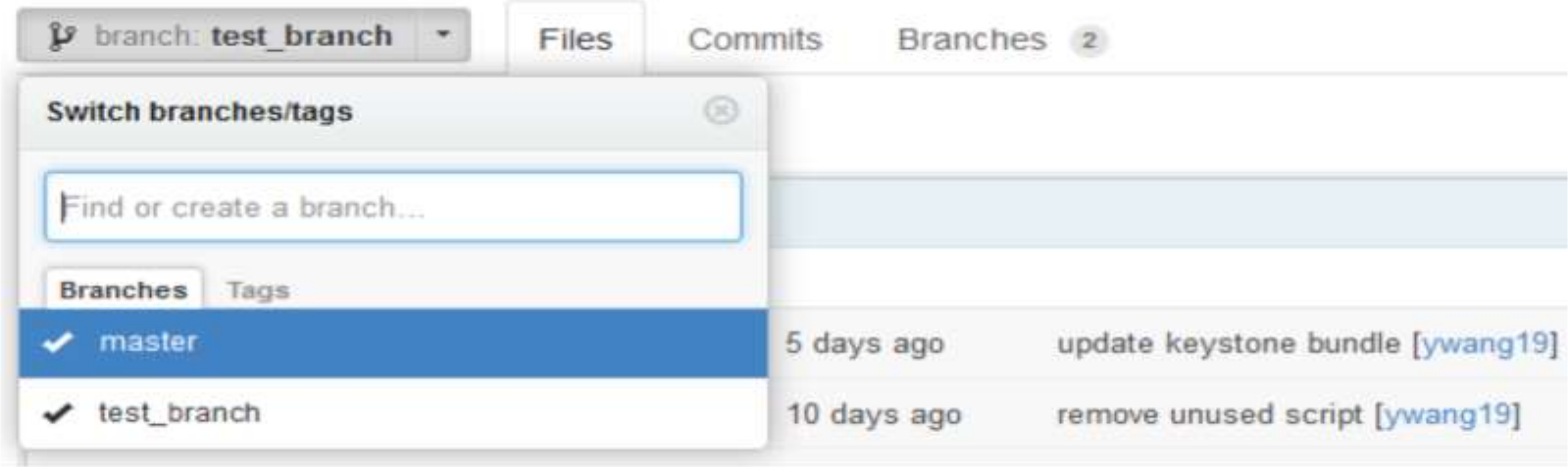
- Configure remotes (on local)

```
git remote add upstream https://github.com/ywang19/virtual-storage-manager.git
```

Contributing to VSM

2. Fill one issue into the issue list on JIRA (https://01.org/jira/browse/VSM) , or just choose one from existed list.

3. Then create one branch with name like “*wip-<issue no>*” on its own repository.

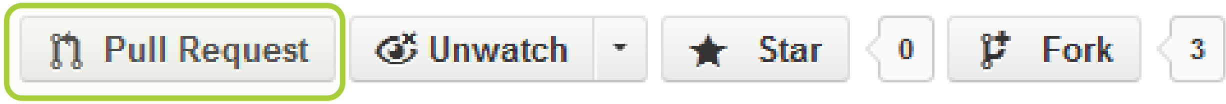


5 days ago update keystone bundle [ywang19]

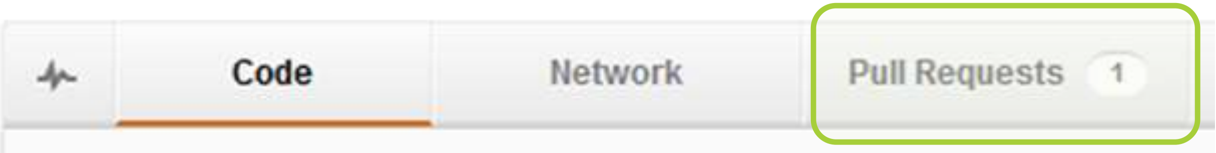
10 days ago remove unused script [ywang19]

Contributing to VSM

4. Finish the issue fixing, and verify locally. then generate one pull request on github.

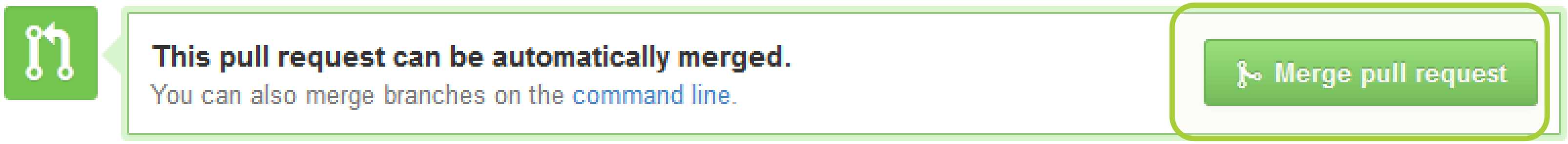


5. the pull request will be reviewed by gatekeeper.



Contributing to VSM

6. if review is passed, the pull request will be merged into 01org repository.



This pull request can be automatically merged.

You can also merge branches on the command line.

7.

```
git rebase upstream/master master
```

Work with Community (Nabble)

VSM Community Summit (VCS)

To connect developers and end users together, we will hold virtual summit at the period from one major release to its first point release. Some activities will include:

i) Introduce new features in new release

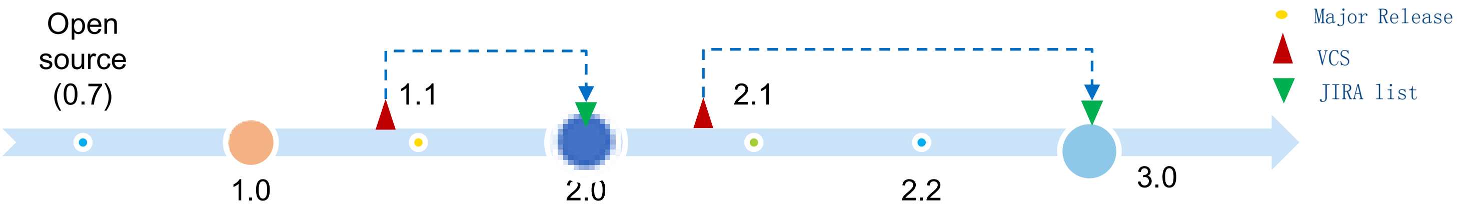
ii) Listen to developers and users 'feedback

iii) Clarify JIRA list candidate for next major release

VPS	Date
#1	Mid March
#2	2H15 (TBD)

Note:

- The first VCS Date is 19 March, a notification should also be delivered on mailing list .



Work with JIRA

Issue Reporting Channel

How to make bug or feature requests

VSM as an open source project, it will follow up common approaches seen in open source community. Any bug or feature requests need to come via Jira, and any general discussion via the mailing list or those tickets.

In the (hopefully rare) event, if there are proprietary things to ask, we could assign a special security level, all tickets in this level can only be visible for the reporter and Intel engineers, but use of that should be very rare.

JIRA Clarification

Overview

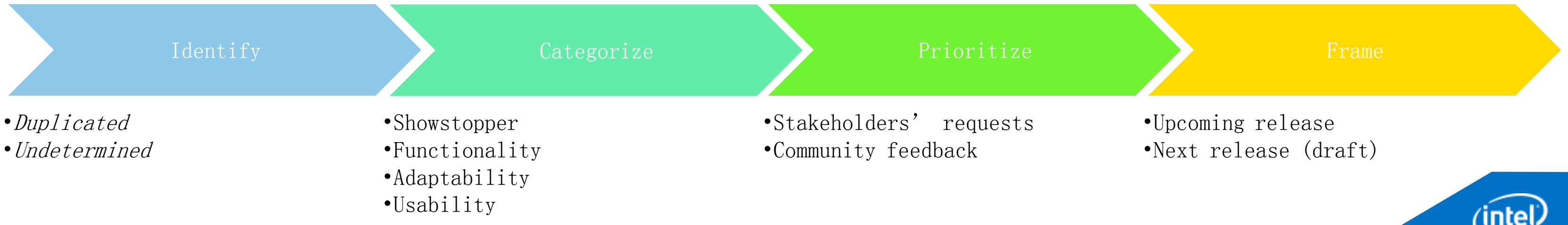
This activity will determine major JIRA tickets to be covered in +2 release, it will occur on VPS, we will determine what will be included. The expected outputs are ticket list for +2 release

The ticket list will become the base for +2 release, and we will keep communicating with community to fine tune the list.

Guideline

- We will try and associate a development task item to a specific release in general, and then fine tune when we work on each release.
- By default, bug fixes will go to current development release. New features go to backlog and waiting for next JIRA clarification cycle.
- Clarify JIRA issue list for the next next (+2) release after an new release is delivered, and tune the issue list for the next version. that is when v0.9 is release, we are here to clarify v1.1, when v1.0 is released, we will tune v1.1 issue list, and clarify after v1.1 issue list.

Workflow



Identify
- *Duplicated*
- *Undetermined*

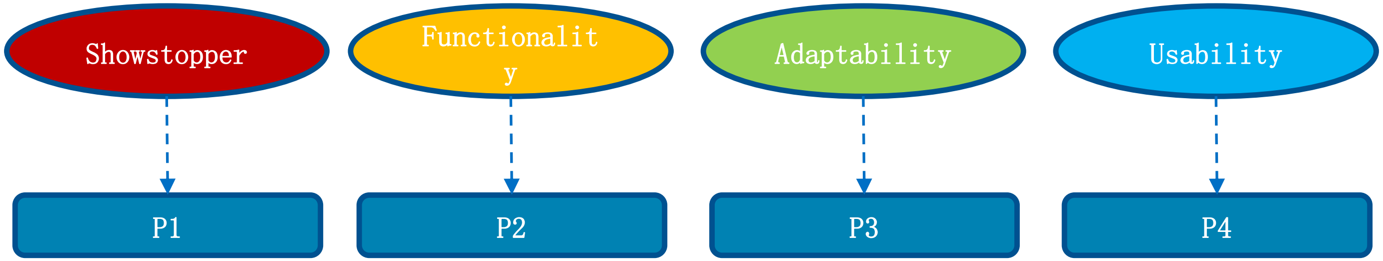
Categorize
- Showstopper
- Functionality
- Adaptability
- Usability

Prioritize
- Stakeholders’ requests
- Community feedback

Frame
- Upcoming release
- Next release (draft)

JIRA Tickets Triage

We have introduced in Jira a "components" field to help triage issues and features into the 4 options below.
Default mapping between components and development strategic priorities,
though it may vary case by case.

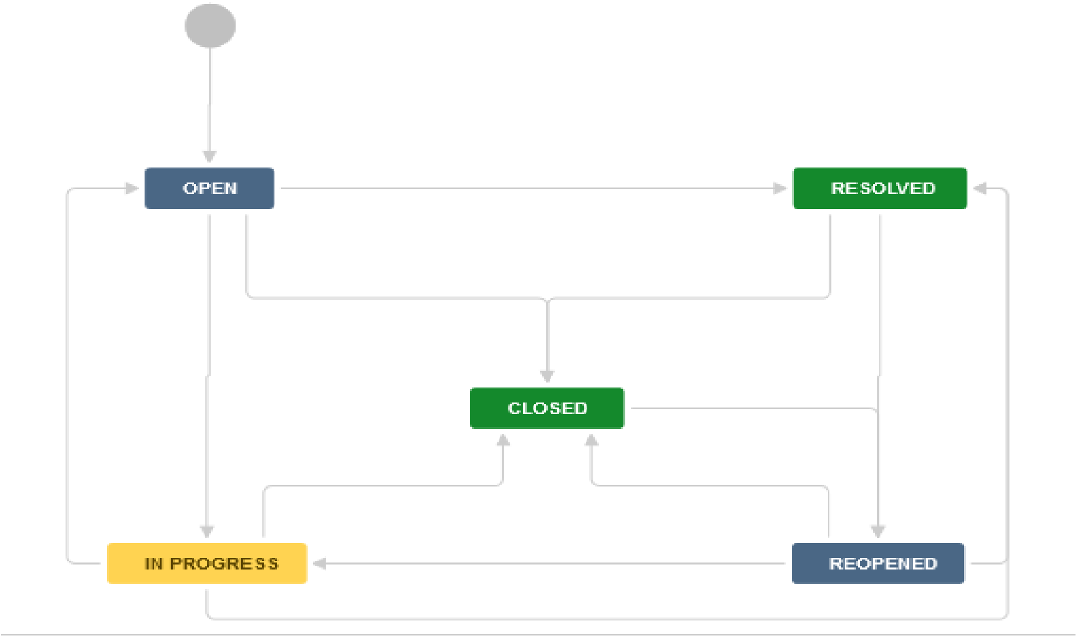


- ***Showstopper***
  - *An issue which stops critical operation paths, with no workarounds.*
- ***Usability***
  - *Involves something which allows more ease of use with VSM*
    - *minimal add-ons to core functionalities, such as easier are all under this component*
  - *Some usability items could be converted into functionality items*
- ***Functionality***
  - *Resolves issues or features that will extend or change VSM core functionalities, for example something like radowGW support, EC, Cache tier - are under this component*
- ***Adaptability***
  - *requirements which need VSM working on other un-tested environments, like other OS versions, OpenStack versions, Ceph versions, or working with other as yet un-tested third-party components.*

JIRA Ticket Handling workflow

1. When a ticket is resolved by developer, the status should be changed to **RESOLVED** instead of **CLOSED**. Changing from RESOLVED to CLOSED normally occurs at a few weeks before one release, where the ticket is separately verified by tester. Exceptional cases are, if we identified invalid/duplicated/won't fix ticket, we will set them **CLOSED** directly.

2. If one issue is in OPEN, it normally means it's an new one and we don't have a chance to look at it yet. When we have some understanding of the ticket, and decide to work on it, the status will be changed to IN PROGRESS.



intel Look Inside.™

Thank You!