White Paper: *Project OpenNMS*
January 1, 2008
David Hustace

ABSTRACT

I. The Art of OpenNMS

Artistically, and most importantly, OpenNMS, in its entirety, is open source software forged from the philosophies of both the Free Software Foundation (FSF) and the Open Source Initiative (OSI) providing a free, enterprise grade, network management system to the world.

The FSF and the OSI obviously have different views and separate agendas, yet the success of the OpenNMS project is credited to its stewardship based on both ideologies. The maintainers would like you to believe that this has been methodically planned, carefully negotiated, and profoundly intentional. However, having been very early adopters of a free and open source philosophy, the project's adaptation of these two models has been more of an evolution than a strategy because, often, it was quite accidental or necessary. The project has stumbled upon and maintained its success due to a true commitment to the shared values of both the FSF and the OSI movements.

The user's freedom to use OpenNMS is the software's catalyst that bonds the large number of network and system administrators with the core developers into a cohesive community participating in the advancement of OpenNMS. This is, without question, what ultimately feeds the project's success in producing quality software.

Freedom to use is defined by the FSF as the users ability to run, copy, study, and change the code without restriction. Participation is defined by the OSI as the community's willingness to voluntarily test, document, bug fix, and supply new features to the software. In search of a means to best leverage that potential, the OpenNMS developers have adopted a a unique software design strategy. OpenNMS' API and extension points attempt to move as much of the intellectual property (IP), as possible, out of contributed code and into it's configuration. This strategy aims to avoid code forks, thereby, enabling code contributions that benefit the project and freeing users from having to maintain a forked version when customization is required.

Often times code forks eventually lead to the "kiss of death", a.k.a. code paralysis, for any software deployment. Because OpenNMS' API facilitates contribution, software deployments avoid the potential paralysis of maintaining a code fork. This eases system administration, employee transitions, and allows for the continued benefit of bug fixes and code improvements from the mainstream release. (*It is important to note that this applies to their contributed code as well!*) While the GPL is designed to fork, this "behavior defined by configuration" strategy has helped to foster greater contribution and a long successful project life for OpenNMS.

OpenNMS serves as visible testimony to the ideals championed by the FSF and the OSI. Why is the free and open source model a good fit for the OpenNMS project? Here are the three key reasons:

- Network management tool users have historically been, and continue to be, a highly collaborative community
- Network management in traditional, non-service provider companies, is seen as a cost center
- The OpenNMS maintainers are network management service professionals

Personal rancor: *This implies that the model chosen for OpenNMS doesn't always fit with every software development effort. For me, commercial models are certainly viable models given different sets of constraints. There are two attributes of free and open source software that always trump the traditional commercial closed models: code review and no code escrow requirement. This is why I believe that free and open source is the "best" overall software development model and that partially free and partially open models are only marketing fodder. Although the business model appears to be working, they believe they can fully benefit from the community development aspects proposed by the OSI. For the record, I do believe that dual-licensing is an acceptable model, as long as the project's stewardship fosters contribution, because all the code is still free and open. But hey, this enough subject matter for another paper.*

Additionally, beyond the fundamentals promoted by of the FSF and OSI, the OpenNMS community has also recognized a need for corporate sponsorship in free and open source projects. When a project encompasses a body of work as complex and as ambitious as OpenNMS, the resources required to sustain the community are far beyond what is freely and voluntarily available (today, at least) and must be provided in some way. Also, corporate adoption demands viable commercial support and training. In very much the same way that Red Hat sponsors the Fedora Project, the OpenNMS Group sponsors the OpenNMS project through commercial services: support, training, consulting, and custom development. The main difference is that Red Hat creates and sells a distribution of Fedora with its own branding. The OpenNMS Group does not currently do this.

The vision of the FSF is that software is free, as in free to use. This is often confused with "free" as in "free of charge." This freedom is a matter of liberty, not price," according to the FSF. Certainly the implication of free to use is a null cost, but it doesn't imply that free software is a gift. Free software can be packaged for distribution to be sold, as does RedHat and Novell, as long as all of the software is still freely re-distributable and access to all the source is still provided. Projects such as CentOS make use of this redistribution right. The significant change made to the CentOS distribution from the Red Hat production is the removal of all Red Hat branding. In the context of this definition, OpenNMS is free as well and is trademarked by The OpenNMS Group, Inc.

II. The Science of OpenNMS

Scientifically, the core technologies that are central to OpenNMS' design act to fulfill the requirements of what is called a "management station".

The term management station (MS) was first defined in paragraph 3 of RFC 1157, "SNMP Architecture" and, again, more recently, in paragraph 2 of RFC 1901, "Components of the SNMPv2 Framework". The industry has since coined the term NMS, or Network Management System; hence OpenNMS.

Since the publication of these RFCs in 1990 and 1996, respectively, internet network (internet) and related technologies have improved, adapted, and grown in ways that were never anticipated. Yet the fundamental architectures of SNMP and the underlying transport, TCP/IP, have not significantly changed from their original designs. These designs have stood the test of time by sustaining the tremendous growth of IT and the network management problem as we see it today. No truer testament can be awarded to the vision set forth by the designers in their original specifications. The developers of OpenNMS are continuously crafting a better and more robust MS based on these transport technologies. The goal of the project's community of scientists is to provide users with quality software that gives them the ability to manage any advanced application of technology today and in the future.

There are two network management models that are fundamental to the implementation of a MS, such as OpenNMS, that were also introduced in 1996. The International Telecommunication Union (ITU) published the Telecommunications Management Network (TMN) model as a design for carriers to manage service delivery. Then, in 1997, the ITU published the FCAPS model for comprehensive management of an organization's technology assets. These models are still used today by network management professionals as a guide for designing comprehensive monitoring solutions. The OpenNMS maintainers also use these references as milestones toward achieving the project's goals and continued improvement of the software.

TMN is a model for managing "Open Systems" within a telecommunications network and defines four logical layers: Business Management (BM), Service Management (SM), Network Management (NM), and Element Management (EM). FCAPS is an acronym for a model that further defines NM using the terms:

- Fault Management (FM)
- Configuration Management (CM)
- Accounting (A)
- Performance Management (PM)
- Security (S).

Consequently, these two models are also often used as a checklist when evaluating and choosing a MS. A quick evaluation of OpenNMS will find that it provides features that support the SM and NM logical layers of the TMN model and each component of the FCAPS model with an emphasis on FM and PM.

OpenNMS provides extremely comprehensive FM. Faults in OpenNMS are detected via three distinct and separate mechanisms: service polling, receipt of unsolicited messages (typically SNMP traps), and thresholds evaluated against performance data. OpenNMS also provides extremely comprehensive PM via several mechanisms that are based on a robust data gathering API called the Service Collector Interface. Current implementations of the Service Collector, SNMP, JMX, HTTP,

and NSClient, gather data that can then be utilized in performance graphs, thresholds, and TopN analysis.

The remaining components of the FCAPS model: CM, A, and S, are each addressed, to some degree, within OpenNMS. However, they are not implemented as comprehensively as defined in the model or as dictated by de-facto standards established by software that focuses on each these components. For example, the A component is partially implemented by providing usage data for the network and system resources such as bandwidth, CPU, and disk usage. While this data isn't directly exposed to customer billing systems via the WebUI or an API, the data can be exported for use in these billing systems. An NMS, supporting component A, should synchronize with the company's provisioning system so that customer records can be easily associated with the resource's management data. OpenNMS provides this capability via one of the enterprise integration features available in OpenNMS, the Model Importer Service.

The C component of FCAPS is also partially implemented through the use of OpenNMS' asset records and with its ability to shut down or turn on network interfaces via the WebUI. OpenNMS partially implements the S component by providing support for SNMPv3, user based access control to the WebUI via local or LDAP security models, and with its integration to the intrusion detection and vulnerability assessment systems, Snort and Nessus.

III. Credits

> *Today, the OpenNMS software is truly a free to use, robust, and reliable enterprise grade network management station.*

Due to the freedoms inspired by the FSF and the OSI, OpenNMS software has been used to improve a continuity in businesses, globally, by simply removing the worries over issues involved with licensing, code escrow, security, and vendor lock-in all the while providing software that is of higher quality at a significantly lower cost. By supporting standard network management protocols with an eye toward the use of well-defined network management models, sound engineering practices, and community contribution, OpenNMS software has become an extremely reliable, robust, scalable, and supportable network management solution for today's corporate enterprises.

Accordingly, credit should be given where credit is due. So firstly, credit to Messrs. Richard M. Stallman and Eric S. Raymond for their well-known and untiring work supporting free and open source software and to Mr. Steve Giles for the initial vision for this project.