Management Intranet: Integrating Web-based Network Management Applications

Jim Turner
Cisco Systems Inc.
125 Rio Robles
San Jose, CA 95134 USA
jiturner@cisco.com

Swami Jayaraman
Cisco Systems Inc.
125 Rio Robles
San Jose, CA 95134 USA
swjayara@cisco.com

Tizil Zecheria
Cisco Offshore Development Center
(HCL Technologies Ltd).
158, NSK Salai, Vadapalani
Chennai TN 600 026 India
tizil@cisco.com

ABSTRACT

An approach to integrating web-based network and device management applications is described which relies upon a centralized repository containing information about managed devices, management applications, users and the privileges which the users have to manipulate these devices and applications. This approach facilitates the creation of a secure, easy-to-use set of cooperating applications, which enable the user to conveniently manage large groups of network infrastructure devices such as routers and switches from one single point of access.

Keywords

- *Management Service* – An independently launchable (via URL, API, or both) network or device management task, readily understood by operations personnel.
- *Credentials* – The information to be used to authenticate access to a device or a Management Service. In the case of a device, this includes SNMP community strings, and passwords used to access the device. In the case of a Management Service, it would be the username and password or any other information, which would enable a user to invoke this service.

INTRODUCTION

The network management environment of today is becoming increasingly heterogeneous with diverse applications managing different network elements. Complex networks often have applications from the same or different vendors attending to a host of management functions. Though these applications operate in the same network, often collecting and using the same data, only minimal data is exchanged between them. These applications often operate with separate user identities and security model, resulting in a highly fragmented environment in which even the most basic management information, such as list of devices and associated credentials, are sparingly shared between applications.

Management Intranet is a paradigm for the integration of the web-based network and device management applications to operate in a common operational environment with a higher level of interaction provided through the use of web technologies and Internet standards. This white paper explains the concept of Management Intranet, its principles and the essential components needed to build a Management Intranet.

PRINCIPLES OF MANAGEMENT INTRANET INTEGRATION

The key to the concept of Management Intranet is the level of interaction and not so much the physical presence. Two products installed on the same network would not be part of the same Management Intranet if they do not interact with each other.

The following are the key principles of Management Intranet integration:

- Management Intranet uses open standards as the basis of its design, so that applications from multiple vendors can be part of the same Management Intranet
- All participating applications and functions present a web-based user interface
- All participating applications share a common user identity and security model
- All participating applications share a common device list
- Management services of the participating applications independently invoked as URL-linked tasks from either a product workflow or a customer workflow under a common user interface and point of access (the Management Intranet Home Page)
- A minimal set of Common Management Data (CMD) securely shared among all participating applications to enable a high degree of integration
- Network management applications register to join the management intranet over the network

MULTIPLE LEVELS OF MANAGEMENT INTRANET

The principles of management intranet integration are devised to allow a wide range of applications, possibly developed at different times and by different development groups from the same or different vendors, to work together so that it adds value to the user, much the same way as corporate Intranet services like Human Resources, Travel, etc. However, different levels of integration can exist between network management products and applications. The different levels of integration proposed are:

- **Level 1**: Management Intranet Integration through web hyperlinks, where applications can be cross-launched from each other. Here, the applications do not share data.
- **Level 2**: Full Management Intranet Integration where the basic management information is shared between applications residing on separate/same servers. This basic management information, shared between applications, is termed as Common Management Data.
- **Level 3**: Intelligent integration between applications, where the applications based on the context, launch other applications specialized in a particular network management area, passing the relevant analyzed data, to help the customer perform the specific management task. Here, the applications work in complete harmony, sharing data, sequencing their operations based on individual capabilities to accomplish a management task.

COMMON MANAGEMENT DATA (CMD)

CMD is the most basic set of management information shared by all management applications participating in a Management Intranet. It includes the list of devices and associated credentials, the location of distributed Management Services, such as the instances of management applications and the device domains over which those applications range and administrative grouping information about users and devices. The goal is to keep this data to the absolute minimum, to allow new applications to integrate with the Management Intranet easily, at the same time allowing different applications to evolve on their own time-lines without being locked together by the need to conform to an ever-changing set of shared data. This paper refers to the mechanism for sharing this data as 'CMD replication'. This does not necessarily mean that the CMD should be physically replicated to the participating applications. The primary purpose of the CMD and its replication mechanism is to facilitate broad distribution and sharing among trusted applications and operators the information about:

- Devices
- Location and type of Management Services
- Who is authorized to use the services, and against which devices
- Administrative grouping information of all the above.

The CMD schema must be rigidly controlled to ensure upward compatibility. The objective is to identify the minimum set of objects and attributes that will satisfy the bulk of common application requirements in respect to users, device lists, service location and authorization. Thus, the CMD may include device lists and credentials but not detailed inventory information. The access to CMD must be authenticated and made through a secure channel. Access control to the CMD elements should be possible at a fine-grained level based upon the requester's identity. An administrator should be able to control access to each data element individually.

COMPONENTS OF THE MANAGEMENT INTRANET

User Interface (UI)

The management intranet UI should focus on providing users flexibility to quickly access information and tasks that matters most to them in a single place, while retaining access to the product's original workflow to find relevant information and perform common tasks. It should give launch points to all Management Services from the participating applications. Ideally, all the applications participating in the Management Intranet should have the same look and feel. Since it would not be an easily achievable goal to get the same basic look and feel for applications from multiple vendors, attempt should be made to have the same basic look and feel for applications from the same vendor. An alternative approach is for the applications to allow API access to all their functionalities, so that the users can build a standardized UI over all the applications. Here the user gets the freedom to design and implement custom UI to access the Management Services offered by the product.

The Management Intranet should provide the user with a *Management Intranet Portal*, which would be the common web-based UI for all participating applications. This should provide launch points for all the Management Services registered with the Management Service Repository. Thus, the *Management Intranet Portal* would serve as a single point of access for the users to perform any network/device management operation on the network/devices managed by the Management Intranet. The *Application UI,* which is the default user interface workflow/logic-flow provided by the application vendor, should still continue to be available to the user, if he accesses the application directly, without coming through the Management Intranet Portal.

CMD Repository

The CMD Repository is the data store for the Common Management Data. It includes the replication mechanism for the data set as well. CMD Repository consists of two major data stores. They are:

Common Credentials Repository (CCR)

A common principle of intranets is that the user has the same user id and password across all the functions supported by the intranet. The same principle must hold good for the management intranet. A desirable feature would be to have the user to login only once into the management intranet to access any Management Service from the participating applications.

In a management intranet, you would need the following information to be available from the CCR:

- The list of authorized Management Intranet users and their credentials. These credentials should be valid for all devices and applications in the Management Intranet.
- The list of devices to be managed by the Management Intranet and the associated credentials.
- The list of Management Services each user is authorized to perform. One more level of granularity desired would be to allow configuration of the list of devices in which the user is allowed to perform a Management Service.
- The information on currently logged in users, with their client machine details. This can be used to implement the single-point login to the whole management intranet,

discussed earlier. The applications should also be allowed to update this information into the CCR.

- It would be desirable to have the security framework provide APIs to record 'who does what' in this repository. All applications can push their audit information into this central store from where all actions performed on the network can be traced. The device traps and syslog messages can also be analyzed to identify user actions on the device and then logged into this central repository. This provides the users with one single place to see 'who did what'.
- It is desirable to store the information about proxy servers for various network protocols in the CCR. Whenever the application goes out of the customer network (for example, to the support site of the vendor) for a backend operation, this information can be used.
- Other details, which may be stored in the CCR, are the credentials to reach a vendor's support site and the 'Inactivity timeout' for auto logout from the Management Intranet.

Note: The exact definition of the APIs to update or fetch the information in the CCR should be standardized. The APIs should use a secure channel, since it involves user credentials. Lightweight Directory Access Protocol (LDAP) and Secured Hyper Text Transfer Protocol (HTTPS) seem to be good candidates for the implementation.

Management Services Repository (MSR)

The Management Services Repository would hold the information about the Management Services available in the Management Intranet. The information stored about each Management Service should include the following:

- The Management Service name and version
- Vendor name and contact info for support
- The complete URL to launch the service
- The list of URL parameters it can take. It should contain the name of the parameter and the type of the parameter. A list of permitted values for the type should be standardized with a provision to extend it as and when new requirement comes in. The service URL should not assume this parameter to be present always. In the absence of this parameter, the Management Service should prompt the user for the required data
- The category of network management (Fault, Configuration, Accounting, Performance, Security - here after referred as FCAPS) to which this Management Service cater to. This information can be used by applications to make context sensitive selection of the Management Services to be presented to the user and also used by the Management Intranet UI to group the links
- A sub-category that specifies the area within the major network management domains (FCAPS), where this Management Service belongs. For example, *Voice Fault Analyzer* can be a sub-category within the Fault domain that specializes in analysis of voice faults. The list of sub-categories needs to be standardized with a provision to extend in the future
- To enable the level 3 management intranet integration discussed earlier, applications need to specify an *Integration Tag* value. Let us take an example to illustrate this concept. The vendor of a generic fault analyzer application XYZ specifies that,

  `"If any URL based application takes its processed fault data as a URL POST parameter with the name FAULT_DATA and help the user to analyze the fault further in voice related faults, application XYZ would provide a launch point to this application in its fault monitor window, whenever a voice related fault occurs."`

  The vendor of XYZ also defines an *Integration Tag* named XYZ VOICE FAULT CLIENT to identify any application satisfying this criterion. Now, any application that has the capability to analyze the fault data from XYZ and process the fault further would specify the value of *Integration Tag* as XYZ VOICE FAULT CLIENT. The application XYZ queries the Management Service Repository to get the list of applications that have specified XYZ VOICE FAULT CLIENT in its *Integration Tag* field and provides launch points in the fault monitor window. Applications can build intelligent integrations in this fashion.

The format of the Management Service Repository information and the APIs to interact with this registry should be standardized. It would be preferable to use URL based APIs. For the Management Service Repository format, XML-based registration would be a good candidate. Cisco has already implemented XML-based registration of Management Services in the CiscoWorks2000 family of products, under the name Cisco Management Connection [1].

World Wide Web Consortium (W3C) is working towards standardizing the format for describing a web service using XML. Web Services Description Language (WSDL) is an example [2].

Management Intranet Proxy

A desirable feature in Management Intranet would be to have the user to sign in only once to access all Management Services from the participating applications. Since all participating applications of the Management Intranet might not be using the same technology for detecting a valid user

session, it would be difficult to implement this, unless we have a Management Intranet Proxy. As the name indicates, the Management Intranet Proxy acts as a gateway server to the services hosted by the Management Intranet. The initial user access authentication is done with this server. It verifies the credentials provided by the user with the information stored in the Common Credentials Repository (CCR) and validates the login. After a successful login, it would perform a server side redirect of all browser requests to the actual service provider hosting the service. The results would also be collected and sent back to the server. Here the browser's interaction would be only with the server hosting the Management Intranet Portal. This server would pass the user credentials to the backend servers to get an automatic login. So the user would not need to login when he moves from one participating application to another.

The Management Intranet Proxy neither aggregates the results from the Management Services behind it nor intelligently routes the requests to the appropriate participating application based on a context data. This is not the intended functionality of Management Intranet Proxy, but comes under the domain of specialized applications, generally referred to as *Application Aggregators*. A possible use-case for *Application Aggregators* in the Management Intranet scenario is discussed in the section "Scalability Considerations in Management Intranet".

BUILDING THE MANAGEMENT INTRANET

Building the Management Intranet involves two steps - Implementation of the Management Intranet components discussed above, and the implementation of the CMD Repository Access Protocol by the participating applications. The application, which implements all the basic components of the Management Intranet, together with the features to administer the CMD Repository, is called the Administration Module.

Administration Module (AM)

The Administration Module (AM) application hosts a CMD Repository comprising of the Common Credentials Repository and Management Service Repository. It also provides the management intranet home page and acts as the Management Intranet Proxy, if required. All the administrative tasks of the management intranet can be performed from here. The user through AM, populates the Common Credentials Repository. The participating applications register the details of their Management Services over the network with the AM. The participating applications will read the CMD from the centralized data store of the AM through standardized APIs. Any modification to the data store is permitted from the AM only. AM would use the information available in its data store to construct the Management Intranet Portal. This model is simple to implement, but the disadvantage of this approach is that it has a single point of failure, as the data is not replicated. To overcome this, AM implementation may choose to replicate the CMD Repository to the data store of participating applications as well.

Network administrators may also choose to implement the standardized APIs over their existing credential store (AAA server) to make this act as the management intranet CMD Repository.

CMD Repository Access Protocol

The Management Intranet implementation will not be complete with out the support from participating applications. The applications need to implement the CMD Repository Access Protocol and fetch information from the AM, when it is configured to operate in a Management Intranet environment.

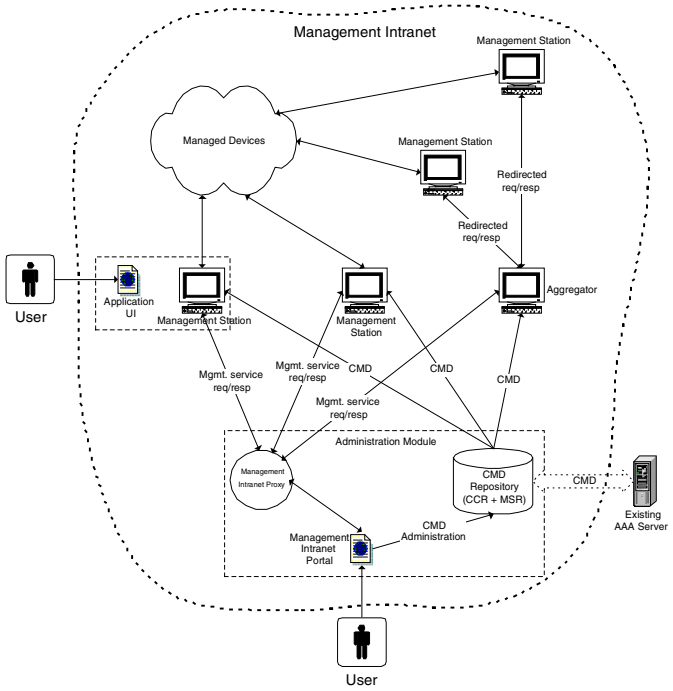

Figure 1: Management Intranet Model

If the AM has chosen the implementation of replicating the CMD Repository, the application needs to build support for that as well. It is desirable to have the applications implement a common standard API to check whether a device/set of devices is managed and supported by an instance of the application. This is needed for level-3 management intranet integration, where the calling application needs to check whether the target application has the device/devices in context managed.

Scalability Considerations in Management Intranet

Management Intranet integrates the various applications operating on a common device list and provides a unified view to the customer. If the number of devices exceeds more than the capability of a single instance of the application, then the user would need to deploy more than one instance of the application. In this case the Management Intranet would not aggregate them and present a single view. The Management Intranet Portal would display links to Management Services from all instances of this application. The user would need to invoke the appropriate instance based on the devices to operate on. In this case, the alternative is to use *Application Aggregators* and register this *Aggregator* application with the Management Intranet. *Application Aggregator*s are specially designed applications to take care of aggregating the data from identical servers and forwarding the Management Service launch requests to the appropriate server instance based on the device list in context. In this case, the *Aggregator* would hide the farm of servers behind it from the user. More details on *Application Aggregators* are beyond the scope of this paper.

CONCLUSIONS

The success of Management Intranet concept lies in the standardization of the protocols and data formats involved in its architecture. Summarizing the above discussion, the following sets of actions need to be completed before Management Intranet implementation starts off in a big way.

- The basic set of management information to be shared as CMD is to be identified. This should include the details to be stored in the Common Credentials Repository and the Management Service Repository. Desktop Management Task Force (DMTF) is already driving an effort to define the Common Management Data based on the initial work done by Jim Turner (co-author of this paper) and Andrea Westerinen of Cisco Systems Inc.
- *The CMD Repository Access Protocol*, the API definitions to access CMD in a Management Intranet, needs to be standardized. Separate sets of API should be defined for accessing Common Credentials Repository and Management Service Repository. A possible implementation of CMD schema could be around the Common Information Model (CIM) defined by DMTF. [3] DMTF has already defined an XML encoding of CIM data (CIM-XML). [4] HTTP and LDAP based access protocols for CIM-XML have also been defined.
- A firm process to be put in place, for extending CMD components and the CMD Repository Access Protocol in future.
- A process needs to be in place for coordinating the usage of *Integration Tag* and to ensure uniqueness of the *Integration Tag* definitions. This is critical to the achievement of Level 3 integration in Management Intranet.

ACKNOWLEDGEMENTS

The authors would like to acknowledge the work of Anders Viden and the ED & NM UI development group of Cisco Systems Inc. in the area of User Interface requirements for Management Intranet, work of ED & NM Security development group of Cisco Systems Inc. in the area of Security requirements for Management Intranet.

The authors would like to thank Hareesh Kumar K. N., of Cisco Offshore Development Center (HCL Technologies Ltd), for his invaluable assistance in editing the paper, and for the graphics.

REFERENCES

[1] “Cisco Management Connection”, <http://wwwin.cisco.com/cmc/cc/pd/wr2k/rsmn/prodlit/cmcnw_ds.htm>, (1 May, 2002)

[2] Christensen, Erik., Curbera, Francisco., Meredith, Greg., Weerawarana, Sanjiva. "Web Services Description Language (WSDL) 1.1". < http://www.w3.org/TR/wsdl>. (1 May 2002).

[3] "DMTF - Common Information Model (CIM) Standards". <http://www.dmtf.org/standards/standard_cim.php>. (1 May, 2002).

[4] "DMTF - Specification for the Representation of CIM in XML". <http://www.dmtf.org/download/spec/xmls/CIM_XML_Mapping20.php>. (1 May, 2002).