WebNMS White Paper

WebNMS Framework: A Complete EMS Framework

Introduction

During the past decade, the Telecom industry has seen a significant surge in the complexity of network infrastructures with widespread proliferation of a new generation of network elements and services.

The raw capacity of the network infrastructure is on a growth curve. To harness the underlying performance capabilities, the operations infrastructure—Operation Support Systems (OSS), Network Management Systems (NMS), and Element Management Systems (EMS)—have to evolve and scale accordingly. The legacy OSSs will continue to operate for the foreseeable future and are unlikely to receive significant enhancements since OSS forklifts are too expensive and disruptive.

The majority of the innovation is likely to be pioneered from the Network and Element management layer functions. Some of the desirable characteristics of EMS/NMS systems are comprehensive functionality, fine-grain customizability to specific domains, extensibility across network elements and services, and ease of integration into service providers’ infrastructures.

A complete EMS/NMS platform will support a powerful combination of Fault, Configuration, Accounting, Performance, and Security (FCAPS) functions, Service Level Management, Provisioning, and Topology, with extensive customization capabilities and extensibility across domains. The resulting solution should integrate seamlessly with OSS/BSS systems, databases, and other management applications in the OSS infrastructure.

While several management applications vendors focus on specific functional areas like Fault, Performance or Provisioning, integrating best-of-breed solutions will likely result in serious issues:

- Lack of tight integration
- Difficult to deploy and manage (multi-vendor interoperability issues with a common set of OSSs, databases, and management applications)
- Difficult to support (need to work with multiple organizations which may have different support infrastructures)
- Not cost effective
WebNMS Framework™

WebNMS Framework is an open, standards-based, comprehensive EMS/NMS platform. Its feature-rich support for FCAPS functions, Service Level Management, and Provisioning yield to a high degree of customization with minimal development.

The graphical development environment enhances developer productivity while exposing the powerful functionality underneath. XML-based configuration files allow baseline customization of the feature set without any coding.

For more sophisticated customization, filters and policy rules can be incorporated into the system through scripts, written in Java or other scripting languages. WebNMS Framework is packed with illustrative examples for a variety of application domains. The API is available in Java, CORBA, and XML. The customizability and extensibility features are available for the core data model, the back-end transactional components, and the front-end session components, as well as for the Java and HTML user interfaces.
WebNMS White Paper
WebNMS Framework: A Complete EMS Framework

WebNMS Framework Fault Management

With today’s complex and heterogeneous network infrastructure and the frequency and intricacy of events, alarms and faults are significantly higher than they were a decade ago. Failures include hard failures, soft failures, misconfiguration, performance bottlenecks, loss of resilience, and more. Mission-critical applications rely increasingly on reliability and availability of the underlying network infrastructure. As a result, dynamic bandwidth demands of applications have surged with a profound impact on the importance of managing network health.

Predictive analysis of events with intelligent association of complex cause-and-effect relationships offers valuable foresight into service-affecting conditions. Preemptive automation can then remedy service degradation with minimal human intervention. Redundant events and trouble tickets should be consolidated and streamlined before they are acted upon. Routine outages have to be distinguished from genuine faults. Network infrastructures with multi-layer physical and logical topologies and provisions for redundancy make fault isolation very complex.

One of the key challenges in fault management is encapsulating these different constituencies in an intuitive fashion with capability to configure intelligence and reasoning beyond the obvious.

WebNMS Framework offers sophisticated fault management capabilities for proactive management of network reliability and availability. Its architecture can be distributed for high scale and performance. WebNMS is capable of managing millions of Managed Objects, and data collection can be over 10,000 events and alarms per second.

Application-specific attributes for different deployment scenarios can be modeled rapidly through intuitive GUI and instrumentation. Custom rules and logic can be easily plugged in. Illustrative examples demonstrate how easy it is to leverage the powerful underlying fault management capabilities offered by Web NMS.

<table>
<thead>
<tr>
<th>Business Benefits</th>
<th>Key Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Manage reliability and availability in real time</td>
<td>• Support for TL1, SNMP, CLI, CORBA, XML, RMI ...</td>
</tr>
<tr>
<td>• Proactive monitoring of events, traps, and alarms</td>
<td>• Event/Trap processing – filtering, suppression, propagation</td>
</tr>
<tr>
<td>• Minimize service personnel dispatch and Mean Time to Repair</td>
<td>• Intelligent correlation and root-cause color-coded alarm views</td>
</tr>
<tr>
<td>• Isolate and resolve service affecting conditions before service degradation</td>
<td>• Alarm drill down categorized by severity, managed resource type, time of day</td>
</tr>
<tr>
<td>• Maximize network uptime</td>
<td>• Fault remedy work flow management, categorize alarms as open, acknowledged, closed</td>
</tr>
<tr>
<td>• Eliminate potential event/alarm storms</td>
<td></td>
</tr>
</tbody>
</table>
WebNMS Framework Fault Management Architecture

WebNMS Framework offers sophisticated fault management capabilities for proactive management of network reliability and availability. The architecture allows for easy integration of events from diverse sources, and provides ways to process, correlate, and present alarms and events based on domain-specific requirements. Illustrative examples demonstrate how easy it is to leverage the powerful underlying fault management capabilities offered by WebNMS. The comprehensive features already available ensure quick results for any fault management solution need.

Event and Alarm Management

Agents resident in network equipment typically generate events under specific conditions without any management request, i.e., asynchronously. Management Systems register for these events and receive them from these agents. Events can also be generated from polling network devices in case of failures, threshold crossings, or other significant events. Events can be SNMP traps, TL1 autonomous messages, and other forms of notifications resulting from discovery and polling. These events can report alarmed or non-alarmed conditions, performance counters, results of diagnostics, etc. WebNMS Framework supports collection of events from a wide range of network devices and other sources.
WebNMS White Paper
WebNMS Framework: A Complete EMS Framework

WebNMS Framework uses events as the basic unit of management information. Events are modeled as objects that define the basic attributes and actions needed for effective processing. Input events are first parsed as needed to determine how they are to be processed. Parsers convert raw data to WebNMS Framework events that can be passed to users and other applications in an actionable manner. They are normalized into WebNMS Framework events for streamlined parsing, filtering, and propagation. Events are deciphered and correlated with other relevant events and conditions to determine if it is an alarmed condition. Events can be suppressed or escalated to alarms after correlation.

Alarms are processed events to alert users of problems that need attention. Alarms are either propagated, or suppressed when the alarm needs to be ignored. Every alarm is associated with a unique entity which can be defined depending on the application. Duplicate alarms for a specific entity are suppressed. Propagated alarms are processed using root cause analysis. However, it makes sense to have only one outstanding alarm against a failure. In addition to suppression and correlation, alarm aging is supported to detect and suppress deliberate and ephemeral alarm conditions. Certain events like a loss of frame should not result in an alarm condition since this happens in every network. However, a threshold may need to be set to trigger an alarm if a certain number of frames are lost in a specific time interval. WebNMS Framework supports all the necessary fault management capabilities needed for diverse networks, systems, and applications.

**Customizability**
Every aspect of the event and alarm processing can be configured, fine-tuned, and customized. Domain-specific rules can be applied to filter events based on thresholds, time, count, etc. User-written event monitors can register with Fault Management to be notified as and when events are received. Policies can be used to manage the event and alarm databases. The rich set of services offered by the fault management module are exposed through a set of well defined and easy to use APIs, using which scalable, secure and reliable applications can be developed with ease. With an open architecture and hooks for extensive customization, WebNMS Framework lends itself to building custom applications quickly and easily.

**Security and Scalability**
A relational database is used as the persistence layer to facilitate high availability, scalability, performance, concurrence, and atomicity of all the operations. All the persistent information is stored in the database and the information is used for generating reports and for auditing. An internal caching mechanism helps to speed up transactions.

The WebNMS Framework server consists of three tiers of server components, separate from the RDBMS. These include

- the management server (mediation) tier, which provides protocol mediation with the managed systems;
- the back-end server tier, which provides database transaction and encapsulation service; and
- the front-end server tier, which provides client session management services.

The front-end tiers support a large number of clients. Management servers support management of large networks, achieving scalability on both the size of the networks and systems that can be managed as well as the number of clients that connect to the servers. There can be more than one deployment of each one of these servers to achieve massive scalability. The servers can be deployed with some redundancy so that any failure results in failover to other server components. This feature is crucial in high-availability applications.
WebNMS Framework Configuration Management

For many management solution deployments, what customers desire and need most are effective configuration tools. Unfortunately, building configuration tools is one of the most effort-intensive and time-consuming aspects of management solutions. Traditional approaches to configuration have relied on handcrafting configuration screens bundled with configuration management code in a two-tier architecture. With this approach, a typical configuration application is a complex set of inter-dependent screens which makes the application hard to build, maintain, or change when new requirements inevitably roll in.

WebNMS Framework Configuration Management is designed to make the process of building complex configuration management applications much more productive. With its approach of using discrete parameterized XML documents to drive configuration activity, WebNMS Framework configuration management makes it easier and less risky to develop, extend, and modify configuration applications.

The key benefits of AdventNet's configuration management approach derive from some important design elements, including XML and a server-based architecture:

- It uses an XML task-oriented architecture that ensures configuration functions can be added, removed, and modified without impacting the rest of the system.
- Built on a powerful XML mediation engine support SNMP, TL1, CLI, and other management protocols, it drives network configuration through XML documents.
- It uses a three-tier architecture, where the server stores, activates, and otherwise processes configuration tasks, allowing re-use, audits, scheduling, and a number of other functions.

WebNMS Framework Configuration Management enables a high-productivity model for delivering configuration management solutions for a wide range of domains. The extensibility and flexibility offered by the framework and tools allows for adding support for any configuration management need, including configuring devices using proprietary protocols. This discrete task approach permits incremental development, as well as development by large teams without excessive coordination. These benefits of WebNMS Framework configuration management will cost-effectively ensure the success of your applications.

<table>
<thead>
<tr>
<th>Business Benefits</th>
<th>Key Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lowers the time and cost of delivering configuration</td>
<td>Open, Standards-based architecture with XML, J2EE, SNMP, TL1, and CORBA</td>
</tr>
<tr>
<td>management solutions</td>
<td>support for NE and OSS integration</td>
</tr>
<tr>
<td>High productivity model using XML for building</td>
<td>Three-tier architecture with server support for storing, scheduling, auditing,</td>
</tr>
<tr>
<td>configuration management</td>
<td>etc.</td>
</tr>
<tr>
<td>Supports a wide range of configuration management needs</td>
<td>Supports parameterized XML tasks for easy management of configuration</td>
</tr>
<tr>
<td>in various domains</td>
<td>functions</td>
</tr>
<tr>
<td>Integrates easily into existing environments</td>
<td>Supports rollback capability for recovery from failures</td>
</tr>
<tr>
<td>Supports secure and audited configuration activity</td>
<td></td>
</tr>
</tbody>
</table>
WebNMS Framework Configuration Management Architecture

WebNMS Framework Configuration Management makes building complex configuration management applications easy. The configuration management design and architecture help the process of creating configuration functions through an XML document model. Using this common model, functions spanning multiple devices supporting multiple management protocols can be created through these paramaterized XML documents known as tasks. These configuration tasks can be made available for a wide range of configuration activities, and can be easily used to set up and execute complex configuration operations.

The server-based architecture, unlike traditional two-tier approaches, allows the server to store configuration tasks as templates and perform a number of useful functions. These include securing access to these tasks to authorized users, providing audit trails for all configuration activity, scheduling configuration tasks, enabling periodic retry of failed configuration tasks, and updating server databases upon completion.

WebNMS Framework Configuration Management architecture is built on the XML mediation engine in WebNMS. This mediation engine converts to and from XML into specific management protocols like SNMP, TL1, CLI, TFTP, and other protocols.
WebNMS White Paper
WebNMS Framework: A Complete EMS Framework

The core of the WebNMS Framework configuration management is the configuration server that is responsible for managing the configuration activities for the network. The configuration server ensures that configuration activities commence operation in the desired states, possess the proper parameter values, and form the desired relationship with other components. To achieve this, it interacts with various components like the task processing, configuration providers, as well as WebNMS Framework modules like security and topology.

Configuration management functions are represented as discrete User Configuration Tasks. Task processing involves identifying attributes and devices, performing authorization checks, storing attributes and device lists separately, task manipulation, task execution and result generation. The storage of attributes and devices in a persistence data store, i.e., the relational database, allows reusing tasks already created for additional configuration operations. Before the task gets applied to the device, task manipulation rules can be applied so that the task is modified to suit a particular device or application need. This may include user specific task manipulation rules. Rollback is incorporated into result processing with support for a few different kinds of roll back mechanisms.

WebNMS Framework Configuration management makes it easy to configure network elements using diverse management protocols. To communicate with the network element, protocol-specific configuration providers can be associated with the configuration server. The configuration server provides seamless integration of multiple configuration providers, including standard providers for SNMP, TL1, etc.

To address the security threats that could arise, authorization checks are performed before the task is configured. The WebNMS Framework security management service is used for this purpose. When vast networks are involved, logging configuration details becomes mandatory. The configuration management supports a customizable audit facility.

Thus, the configuration management module provides a customizable framework with a rich set of APIs which can be used to develop complex configuration applications in a timely efficient manner.
WebNMS White Paper
WebNMS Framework: A Complete EMS Framework

WebNMS Framework Provisioning
Automated, flow-through provisioning is a necessary capability for service providers to keep their operations costs under control. Service provisioning costs and customer dissatisfaction can spiral out of control if manual, error-prone processes have to be used. Complete flow-through provisioning capability involves complex integration across a number of different databases, network elements, management systems, and workflow systems, as well as interaction with the operations staff.

The key to meeting this requirement is a service activation system that allows high-productivity incremental development and can easily integrate with the diverse systems and interfaces in a typical operations systems infrastructure.

AdventNet's provisioning module is designed to meet the challenges of complex provisioning applications. It leverages a number of recent software industry developments and standards, including XML, GUI rendering, messaging, J2EE technologies, XML scripting, etc. Using these technologies, it provides a powerful, dynamic, template-driven provisioning module that makes delivering provisioning applications an order of magnitude simpler.

AdventNet provisioning brings together data from diverse databases, systems, network elements, management systems, and other sources, and makes it easy for operations to activate complex services that depend on this disparate data. With the template model, service activation capabilities can be developed, added, removed, and modified without affecting any other service activation capability or even restarting the server.

<table>
<thead>
<tr>
<th>Business Benefits</th>
<th>Key Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Lowers costs of service Provisioning and activation solutions</td>
<td></td>
</tr>
<tr>
<td>• Reduces operations costs with automation of service activation</td>
<td></td>
</tr>
<tr>
<td>• Enhance customer satisfaction with reduced provisioning delays and errors</td>
<td></td>
</tr>
<tr>
<td>• Improve inventory efficiency with improved synchronization across network and operations databases</td>
<td>• Open, Standards-based architecture with XML, J2EE, SNMP, TL1, CLI, and CORBA support for NE and OSS integration</td>
</tr>
<tr>
<td></td>
<td>• Template model for rapid delivery of service activation capabilities</td>
</tr>
<tr>
<td></td>
<td>• Business-rules capability for dynamic control and customization</td>
</tr>
<tr>
<td></td>
<td>• APIs for building applications and interactive provisioning functions</td>
</tr>
<tr>
<td></td>
<td>• Integration with physical inventory databases including updates</td>
</tr>
</tbody>
</table>
WebNMS Framework Provisioning Architecture

AdventNet’s provisioning module is designed to meet the challenges of complex provisioning applications. It integrates many functions into a common module, which can easily be extended for domain and vendor-specific solutions. The essential function of service activation involves a number of elements, including gathering data from network devices, logical and physical inventory databases, and operators or other OSS systems. The provisioning module integrates these diverse sources of data and makes it easy to gather the required information prior to activating a provisioning change.

The provisioning framework is a module of WebNMS and is built using the WebNMS framework service. It uses many of the FCAPS functions to implement provisioning, including reading and updating physical inventory. The XML-based configuration module of WebNMS provides a powerful way of accessing data from network elements and activating services on these network elements. It provides an XML document model for interacting with the network elements, making it possible to set up complex service activation functions in XML provisioning templates. It can support a wide range of devices using SNMP, TL1, CLI, Netconf, and other management protocols.

The provisioning templates are XML documents that have parameterized provisioning profiles for configuring multiple devices in the network. The operator will provide the parameters while doing the provisioning operation. The templates also contain information for automatically rendering the forms that request these parameters from the operator.
during provisioning. With complete support for complex provisioning scenarios and scripting capability to extend and customize further, these provisioning templates can cover almost every service activation requirement.

The business rules are Java filters that operate on the templates before they are presented to the user or applied to the network. Rules can also be applied after the completion of a provisioning operation. Some configurable filters that are driven by XML rule files need to be provided to minimize the coding to achieve simple tasks, e.g., allowing certain provisioning templates only on off-peak hours. The WebNMS Policy engine is integrated with the provisioning module.

A major goal of the provisioning framework is to allow equipment manufacturers or service providers to build their provisioning extensions very easily. They can build provisioning module extensions and expose that functionality via the provisioning APIs. The module will provide a common provisioning API, but applications can supplement these APIs using one or more domain and vendor-specific provisioning APIs.

With the WebNMS provisioning module, service providers and equipment vendors have a powerful tool for automating service activation and integrating it with their OSS infrastructure. This high-productivity, dynamic provisioning capability can save significant costs in building, deploying, and operating service activation systems.
WebNMS White Paper
WebNMS Framework: A Complete EMS Framework

WebNMS Framework Performance Management

Performance management offers a foundation for proactive management of efficient network resource utilization, capacity planning and prediction, and impact analysis. It can optimize ROI of network infrastructure investment by providing a deep insight into cost/performance tradeoffs at various levels of network resources. The core functions of performance management are data collection and mining, analysis, and reporting. Data collection involves deciphering and filtering raw performance data from network elements received through SNMP, TL1, CLI, CORBA, etc. Analysis usually involves thresholding, trending, and hysteresis. Reporting the analyzed data or Key Performance Indicators (KPI) in actionable manner to the concerned applications or operations personnel will drive proactive capacity planning, network optimization tasks, network readiness assessments for new application roll outs, etc. This information is also critical for other applications like fault management.

Changes in critical performance management metrics can be subtle and occur over extended periods of time or short-lived and conspicuous. A discerning eye for subtle deviations from the norm is essential to identifying the former scenario. The latter, though easy to visualize, requires an insight beyond just the quantitative metrics since transient conditions may not always be service affecting.

WebNMS Framework Performance Management provides a powerful module for data collection, analysis, and reporting of network, system, and application performance information. It supports a range of data collection mechanisms, including SNMP, TL1, CLI, etc., as well as the ability to add new mechanisms as needed by specific environments. It uses a scalable architecture to support large carrier network needs. Using a back-end relational database and a multi-tier architecture with support for distributed collection servers (probes), WebNMS Framework supports large-scale data collection and reporting needs. Its open, standards-based architecture allows solution providers to extend the performance management capabilities to any specific need.

<table>
<thead>
<tr>
<th>Business Benefits</th>
<th>Key Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Realtime view of network performance</td>
<td>• Support for TL1, SNMP, CLI, CORBA, XML, RMI ...</td>
</tr>
<tr>
<td>• Optimal usage of network resources</td>
<td>• Analyze, filter, and aggregate network performance data</td>
</tr>
<tr>
<td>• Identify bottlenecks and unleash idle capacity</td>
<td>• Customize aggregation policies</td>
</tr>
<tr>
<td>• Foresight into service affecting conditions</td>
<td>• Multi-level thresholding and hysteresis</td>
</tr>
<tr>
<td></td>
<td>• Intuitive and graphical performance management reports</td>
</tr>
</tbody>
</table>
WebNMS Framework Performance Management Architecture

WebNMS Framework offers sophisticated performance management capabilities for managing efficient utilization of network resources, proactive planning, and prediction and impact analysis. The performance management module provides data collection, analysis, threshold alerting, and reporting capabilities for a diverse set of network, system, and application management needs.

Data Collection, Analysis, and Mining

WebNMS Framework supports PM data collection from a wide variety of interfaces like TL1, SNMP (RMON2, MIB2), CLI, and NetFlow, thereby streamlining network instrumentation for performance management. Bandwidth analysis can be based on aggregate volume and/or rate measurements. Intelligent grouping by sub-networks, business units, geography, users, and applications allows operators and administrators to surface abnormal conditions. PM data-gathering time periods can be customized to support workday reporting to enable operators and administrators to schedule routine maintenance tasks to off-peak hours or to analyze traffic patterns for a given set of usage hours.

Custom filters and exception events can be used to channel and export such conditions to fault management and other applications. RDBMS is used as the persistence layer to facilitate high availability, scalability, performance, concurrence, and atomicity of all the operations. All the persistent information is stored in the database and the information is used for generating reports and for auditing. An internal caching mechanism helps to speed up transactions.

Scalability

The WebNMS Framework server consists of three tiers of server components, separate from the RDBMS. These include the management server tier which provides protocol mediation with the managed systems, the back-end server tier which provides database transaction and encapsulation service, and the front-end server tier which provides client session management services. The front-end tiers are supported to have a large number of clients and the management servers support management of large networks, achieving scalability on both the size of the networks and systems that can be managed as well as the number of clients that connect to the servers. There can be more than one deployment of each one of these servers to achieve massive scalability. The servers can be deployed with some redundancy so that any failure results in failover to other server components. This feature is crucial in high-availability applications.

Reporting

WebNMS Framework supports intuitive graphs that support both granular and aggregated reporting. Highlighting specific abnormalities is relatively straightforward with color-coded graphs. Overall trending can be performed by analyzing aggregate reports and charts over extended periods of time. Some of these tasks can be easily automated. PM data can also be exported to other applications. For example, threshold violations can be sent as events to WebNMS Framework fault management module.
WebNMS Framework Security Service

Managing secure access and user privileges is essential to the successful deployment of management solutions. Service providers and enterprises need to be able to define classes of users and easily set up users and permissions for each of these users based on roles and tasks they need to perform. A good security management solution should permit fine-grained access control to individual operations, as well as individual objects within the network or management system. Furthermore, it should permit easy integration with existing security solutions for single sign-on and other enterprise-wide security applications.

WebNMS Framework Security Management is designed to meet the needs of service providers and enterprises that need comprehensive and efficient control of authentication and authorization services. Based on J2EE security, it includes sophisticated authorization control and authentication services, in addition to plug-able authentication and authorization for easy integration with existing infrastructure.

This powerful security management module ensures your ability to meet the needs of advanced management solution deployments. The built-in authentication service uses the relational database to provide login services. In addition, it supports using external authentication sources like ActiveDirectory when required. Using a plug-able authentication model, it can support a wide range of external authentication services including RADIUS.
WebNMS White Paper
WebNMS Framework: A Complete EMS Framework

The authorization service enables detailed control of all operations each user is allowed to perform, and controls what set of objects these operations can be performed on. One of the challenges with a fine-grained authorization service is the performance impact of complex authorization functions. Using the relational database and property tables, WebNMS Framework security allows common functions to be authorized very quickly and eliminates the performance bottlenecks that would otherwise arise.

<table>
<thead>
<tr>
<th>Business Benefits</th>
<th>Key Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Provides security for all functions enabled by the management solution</td>
<td>• Open, Standards-based architecture with XML, J2EE, SNMP, TL1, and CORBA support for NE and OSS integration</td>
</tr>
<tr>
<td>• Ensures authorized and audited use of management solution capabilities</td>
<td>• Fine-grained and extensible authorization including support for users, groups, roles, operations, and object views</td>
</tr>
<tr>
<td>• Integrates easily into existing environments</td>
<td>• Secure authentication and access control, including SSL and Secure RMI server access control</td>
</tr>
<tr>
<td>• Supports secure access for a large staff with varied skill levels</td>
<td>• J2EE Security model support</td>
</tr>
<tr>
<td>• Protects investment through extensibility and plug-ability</td>
<td></td>
</tr>
</tbody>
</table>
WebNMS Framework: A Complete EMS Framework

WebNMS Framework Security Architecture

WebNMS Framework security management is built on the WebNMS Framework multi-tier architecture, and spans the database, back-end, front-end, and client tiers. It supports authentication and authorization for the complete set of interfaces and applications in WebNMS. In addition, it can be easily extended to support new applications and interfaces when required. With support for plug-able logic, the security module and applications using it can support ActiveDirectory, RADIUS, and other third-party security services for authentication and authorization.

The WebNMS Framework security module provides API interfaces for applications to query the security server for authentication information. It includes client tools for administering users and permissions, as well as setting up roles and views as needed during deployment. The security data is stored in the relational database, or may optionally use LDAP directories or third-party data stores.

WebNMS Framework applications, e.g., Fault Management, as well as other applications access the security management module as a service. The security service provides authentication of...
users, as well as information on whether a particular user is authorized to perform a specific operation on a specific object or set of objects. The user authentication information, e.g., name and password, as well as authorization information, e.g., operations permitted, is stored by the security service and used to provide security services to applications.

Setup of user authentication and authorization information and policies is required to allow the security service to enforce these policies. Administrators use the tools provided by the security module to setup these policies.

Applications like Configuration, Fault, Provisioning, etc., use security management as part of their functions. With the applications built on the WebNMS Framework platform being user centric, the authentication mechanism is used to verify the credentials before allowing access to the platform. Support for user groups provides a mechanism to collectively associate access rights to a set of users.

The applications built on WebNMS Framework define their own objects based on the entities they need to deal with. It is not sufficient to just tie up the access rights of a user with the operation performed. It becomes necessary to have a framework where the permissions are associated with the subsets of objects associated with the application. This requires fine-grained access control. WebNMS Framework Security Management achieves this by defining authorized views and associating operations with the authorized views.

The security module provides a seamless service for external as well as WebNMS Framework applications. Since the security logic is completely separate from the management application's business logic, the WebNMS Framework platform provides an environment where other applications can seamlessly integrate into the platform to utilize the security services.

Contact
Eric Wegner
Business Development Manager
WebNMS, a division of ZOHO Corporation
eric.wegner@zohocorp.com
732-801-9083