www.datamobilitygroup.com

Storage Management and the Continued Importance of CIM

White Paper

January 2004

 76 Northeastern Blvd. Suite 29A Nashua, NH 03062 Phone: 603.881.9031 Fax: 603.881.9035

There can be little doubt at this point in time that storage administrators are generally behind the efforts of the Storage Networking Industry Association (SNIA) to develop a common, open storage management interface based on the Distributed Management Task Force's (DMTF's) Common Information Model (CIM), now known as the Storage Management Initiative Standard (SMI-S).

Proof points include a session on interoperability at TechTarget's Storage Management 2003 conference during which nearly all 600 attendees indicated, by a show of hands, their plans to implement SMI-S in the future. More recently, users representing the IT departments at AOL-Time Warner, Nielsen Ratings, and Wells Fargo outlined their SMIS-S adoption plans to more than 1,000+ attendees at the Fall Storage Networking World (SNW) 2003 conference. Storage vendors now report seeing an increasing number of RFPs mandating SMI-S compliant storage devices and management applications.

Storage networking users understand that the value of open standards—for both users and vendors—stems from a user's ability to select the most suitable products for a particular application or set of applications *and* then easily integrate those products into their computing environments. SMI-S provides an open, and industry-accepted, way of integrating storage devices with storage management applications, as well as storage management applications to each other.

Today, device and management applications vendors are building, testing, and introducing SMI-S-compliant storage products, allowing users to begin matching SMI-S-compliant arrays, for example, with SMI-S-compliant management applications. However, it is important to note that not all SMI-S-compliant solutions are created equal. There are key differences in the way the standard is being implemented—distinctions that will become important as time goes on. To fully appreciate the significance of such differences, users must be able discern between solutions that are CIM-compliant and those that are truly CIM-built.

But before delving into the advanced benefits of CIM-built solutions, let's take a look at some SMI-S basics.

SMI-S — The Basics

SMI-S is a vendor-neutral application programming interface (API) specification that can be built into networked storage devices and management applications. SMI-S is based on the DMTF Web-Based Enterprise Management (WBEM) architecture, an existing, accepted, and published set of standards for the management of multiple disparate devices residing on an enterprise computing network. The WBEM "pyramid" is supported by three open standards:

- Common Information Model—the DMTF data model commonly referred to as CIM.
- xmlCIM—an XML coding specification created specifically for CIM.
- HTTP—a transport mechanism to enable open, interoperable communications among management applications and managed devices that conform to CIM.

However, the WBEM architecture has been adapted to the networked storage environment. In its first iteration, SMI-S will support a number of essential SAN management functions in an open, device-neutral way:

- Discovery—The automatic identification and registration of devices (HBAs, switches, arrays, tape drives) that exist on a SAN at start-up and as devices are added to the SAN.
- Monitoring—A continuous observation by a management application of SAN fabric health and the operation of each device within a SAN.
- Device management—The active control, configuration, and reconfiguration of SAN-attached devices.

While individual device control can be performed manually by IT administrators today, one of the goals of SMI-S is to automate this management in a heterogeneous storage area network (SAN) SAN fabric. Another goal of SMI-S is to establish a Web-based interface that is common to all devices within a SAN. Without SMI-S, this type of communication and management is only possible through a variety of different and proprietary interfaces.

Users looking at SMI-S for the first time should keep the following basic concepts in mind:

1. SMI-S is a model. SMI-S is a guide to building systems using modules that "plug" together. SMI-S-compliant storage modules *interoperate* in a system (in this case a storage fabric) regardless of who built them, provided that the modules use CIM "language" and adhere to sets of specifications called CIM schema.

2. SMI-S is object-oriented. Virtually anything storage-related—physical or abstract (from complex device management applications to bad sectors on a disk platter)—can be defined as a CIM object. A system (e.g., a storage fabric) is modeled using objects that have defined attributes. In addition, any object known now or *yet to be developed* can be defined and encompassed within the model. The SMI-S object orientation allows storage fabrics to scale and adapt to changes in technology over time.

3. SMI-S is command and control oriented. Unlike SNMP that is now commonly used to integrate storage management applications at a basic level, SMI-S is both passive (like SNMP) and active, allowing management applications to not only monitor devices within a storage fabric, but also dynamically configure/reconfigure devices. Command and control of both devices and fabrics can be automated using SMI-S-enabled storage management applications. SMI-S helps to unlock some of the unrealized benefits of intelligent storage fabrics in ways that SNMP cannot.

4. SMS-provides a single unified view of a SAN. SANs are often thought of as single entities, commonly depicted as clouds or referred to as fabrics. SMI-S allows developers to model a SAN as a single, abstracted entity.

Providers, Clients, and Agents

At a very basic level, SMI-S entities are divided into two categories: clients and providers. Clients are management software applications that can reside virtually anywhere within a network provided they have a physical link either within the data path or outside the data path to providers. Providers are the devices under management within the storage fabric. Clients can be host-based management applications (e.g., SRM), enterprise management applications (e.g., frameworks), or SAN appliance-based management applications (e.g., virtualization engines). Providers can be disk arrays, HBAs, switches, tape drives, etc.

Providers communicate with clients via agents that perform a number of functions:

- Act as a repository for information about a specific device
- Enable the discovery of the device by management applications
- Communicate events and perform status monitoring
- Enable interoperable configuration and control

The SMI-S model also describes two additional entities: object managers and lock managers. Object managers can be used as extensible agents by aggregating multiple devices under the control of a single agent. Object managers can also be used as persistent repositories for information about multiple devices. Lock managers allow multiple clients to concurrently communicate with a single provider. The lock manager reserves a provider-associated agent at a given moment in time so that a particular client can perform an operation (e.g., a query or a command); it then releases the agent to another client.

For Providers—A Direct Route

The SNIA's Storage Management Forum, which is tasked with development of the SMI standard, has provided two different routes for vendors to achieve SMI-S compliance. One route allows vendors to attach a "proxy" interface that "translates" an existing product interface into an SMI-S-compliant interface. The proxy approach is used by vendors to make existing products SMI-S-compliant without significant reengineering of the product's management interface.

The more direct route to SMI-S is through the creation of a "native" SMI-S-compliant interface. In this case, the product's management interface is both SMI-S- and CIM-compliant by design. Because a native implementation is by nature likely a more robust implementation of the CIM model, it is in a better position to take advantage of future releases of SMI-S as they are rolled out by SNIA. Advantages of native providers include:

- **Faster delivery of new products and features.** Vendors that have invested in native SMI-S support should have shorter development cycles going forward because they do not have to maintain proprietary interfaces.
- **Extensibility.** Native SMI-S providers can be easily extended to support new SMI-S features.

- **Easier manageability and lower TCO.** Native SMI-S providers do not have to be mapped to a proprietary data model, making them far less complex and less costly to maintain over time.

The two approaches are shown below in Figure 1.

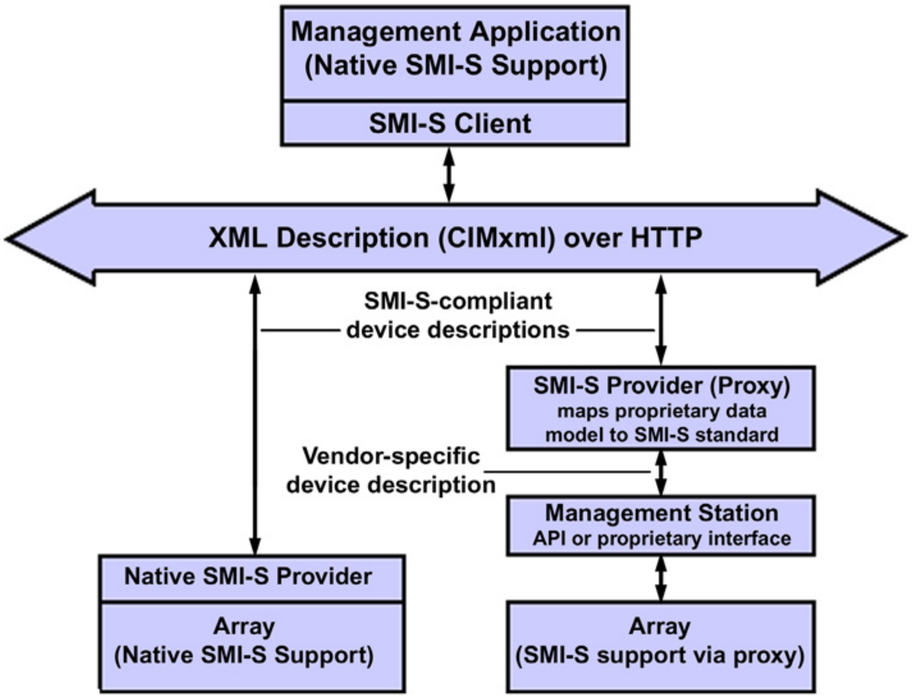

Figure 1

For Management Applications—Flexibility at All Functional Layers

Constructs defined by CIM can be applied to three different storage management layers: the device layer (a layer that includes all devices under management), the management data layer (a layer that contains management-related data that can be used by any storage management application), and the management application layer (the management application layer itself).

SMI-S can adapt CIM constructs to all three layers such that storage management applications from multiple vendors can use all three layers in common ways. This allows vendors to integrate applications without swapping proprietary APIs.

Device Layer

The SNIA Storage Management Forum has done an excellent job of developing and propagating CIM-based device management interfaces described in the previous section as "providers." Providers present a standard interface to the outside management world that is the same for all SMI-S-compliant management applications.

Management Data Layer

Data relating to infrastructure, configuration, policy, etc., can be collected in a CIM-compliant repository. That repository can then be used by any management application capable of supporting the repository to store and retrieve management data. The data within the repository is presented to the management application layer in a format that all management applications can understand.

Management Application Layer

Storage management applications interface directly with devices under management (the device layer) and use management data contained within a common repository (the management data layer) to perform management tasks. CIM can be used to create a common management application to application interface allowing the services of one application to be "exported" to another. (An example would be an archiving application that uses a separate data mover application to transfer data from one device to another.)

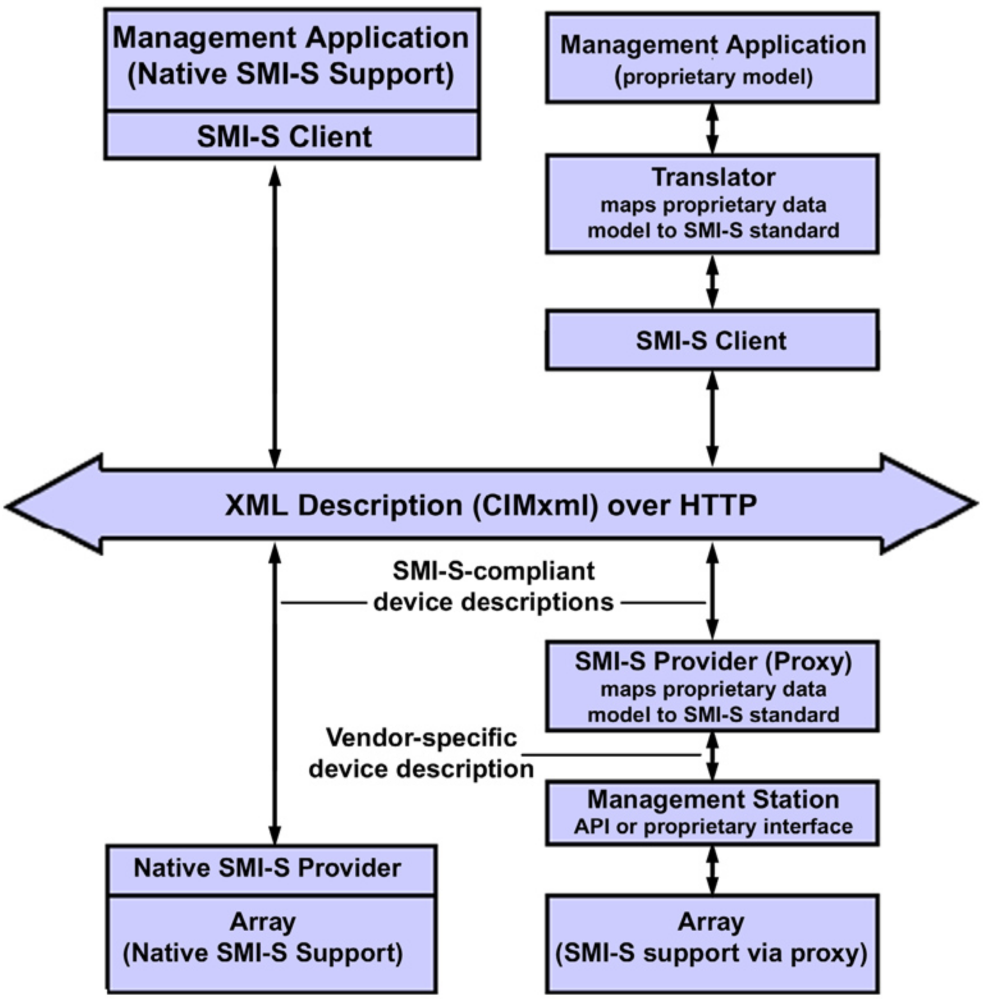

Figure 2

CIM-compliant vs. CIM-built

To achieve these services "export" capabilities, the management application must be CIM-built, not just CIM-compliant.

As shown in the above Figure 2, the CIM-built management application on the left writes data from the SMI-S client directly into a CIM-compliant data source. This enables other CIM/SMI-S management applications to work directly with its data source and enables its "services" to be driven by other management applications. The CIM-compliant management application on the right translates the data from its SMI-S client into its own proprietary data source. Other CIM/SMI-S management applications are not able to use this data source and are not able to access its services because these services are not modeled after CIM.

End users should expect the following benefits from CIM-built management applications that use a pure CIM model:

- **Greater flexibility.** The pure CIM data source of the CIM-built Management Application can be accessed by any other CIM-built management application, enabling management architectures such as ILM and the unique features of best-of-breed solutions to be easily leveraged. At the same time, support for both SMI-S enabled and non SMI-S enabled devices can be accomplished.
- **Less risk of vendor lock-in.** New management applications can be substituted easily without losing any capacity, performance, history, or discovery information.
- **Greater extensibility and scalability.** CIM-built management applications can be easily extended to support new SMI-S and CIM features, without requiring a significant translation effort to map these new features to proprietary formats.
- **Faster time to value.** CIM-built management applications model all physical and logical storage resources in a normalized way, resulting in lower training costs, faster ramp-up by storage administrators, and lower ongoing ownership costs.

Is it CIM-built?

IT administrators looking to adopt SMI-S as their standard storage management interface will want to build a solid SMI-S foundation. This means that early on in the process of product selection, they will need to identify devices that are SMI-S-compliant and management applications that are both SMI-S- and CIM-compliant on all three of the levels outlined above. Doing so will allow them to integrate "best-of-breed" management applications for a particular set of tasks, regardless of the vendor.

Integration of disparate but related storage management applications will likely be required by IT administrators wishing to architect future storage environments to incorporate Information Lifecycle Management (ILM) concepts and processes. Users will be reluctant at best to adopt an ILM strategy that forces them to rip out existing but disparate data and infrastructure management applications simply to replace them with ones that can be integrated only if they come from a single vendor.

Rather, IT administrators are likely to implement ILM processes from products that can be functionally integrated—and having CIM-built management applications makes this possible. It also addresses other important requirements that are near and dear to storage administrators including:

- The ability to manage all devices on a SAN with a single, consistent, web-based user interface.
- The ability to add devices to a SAN in a "plug and play" fashion—to continue to scale the storage infrastructure and add diverse technologies without breaking the existing storage management framework.
- The ability to step beyond passively monitoring a SAN to an environment where devices can be actively controlled and configured by automated management agents.
- Further integration of SAN management functions with enterprise management frameworks.
- The ability to manage FC SANs and IP SANs with the same management application.

In addition SMI-S promises to reduce SAN costs both directly and indirectly. Providing an open, standard API model greatly reduces management application development costs. Management software vendors are relieved of the significant financial burden resulting from sourcing, adapting, and supporting a collection of proprietary APIs. The reduction in development and support costs can be passed on directly to enterprise IT buyers. SMI-S also promises to greatly reduce the storage management burden weighing down on the shoulders of enterprise IT administrators—a burden that, for many, gets heavier with each passing day as more data accumulates within databases, more emails are added to email folders, and more web-facing applications are brought on-line. In short, an intelligent storage fabric allows enterprise IT to manage more data with fewer staff.

Adoption by the storage industry of SMI-S technology could also result in a "knock-on" reduction in SAN costs. It is no secret that storage vendors are trying to accelerate SAN penetration within enterprise IT. While large enterprises have been the primary SAN adopters, small-to-medium-sized shops represent a huge untapped opportunity for SAN vendors. By reducing, if not eliminating, some of the barriers to SAN adoption, SMI-S makes SANs significantly easier for small-to-medium-scale IT shops to deploy and support over time. It's simple: more SANs mean greater SAN ubiquity. Greater SAN ubiquity means higher SAN component volumes yielding, in turn, lower costs for individual SAN components. SMI-S helps SANs become more "mainstream."

 76 Northeastern Blvd. Suite 29A Nashua, NH 03062 Phone: 603.881.9031 Fax: 603.881.9035

for example...

AppIQ StorageAuthority Suite

An example of a pure CIM-built Management Application is the StorageAuthority Suite from AppIQ. Founded in 2001, AppIQ has built an integrated family of storage resource management (SRM), SAN management, and storage provisioning products that support both CIM and SMI-S.

AppIQ StorageAuthority Suite models the entire application storage infrastructure (Oracle, Microsoft Exchange, file servers, hosts, HBAs, fabric switches, and disk subsystems), which enables these resources to be discovered, monitored, provisioned, automated, reported on, and charged back in the same way, regardless of the manufacturer.

Realizing widespread adoption of SMI-S-compliant hardware and software is not likely until sometime next year, AppIQ has done two things to ease the transition to CIM/SMI-S:

1. **Its suite translates vendors' proprietary interfaces into CIM and SMI-S, rather than vice versa.** This allows the suite of software to support proprietary vendor interfaces and special "hidden features" today, while still delivering a pure CIM database and CIM-built Management Application. When vendors replace proprietary APIs with native SMI-S providers down the road, AppIQ's native SMI-S and CIM support will take over, obviating the need for translators.

2. **It has created a CIMIQ software development platform.** AppIQ has packaged up the software tools, testing methodologies, and experience developing SMI-S providers into a program designed to help leading storage vendors accelerate their support for SMI-S. A number of vendors, including Brocade, Hewlett-Packard, Hitachi Data Systems, LSI Logic, Network Appliance, McData, and Sun Microsystems, have partnered with storage resource management and SAN management vendor AppIQ to streamline their SMI-S provider development efforts.

Recommendations for Users

For users interested in building a solid SMI-S-based management foundation—one that will allow a number of storage management applications to be integrated into a coherent SMI-S-based framework—we recommend asking the following questions before selecting storage infrastructure and management applications:

For Storage Devices

- Is the SMI-S provider native or based on the proxy model?
- If a proxy is used, when will a native version be available?
- Can the proxy version be replaced by a native version? How easily?

 76 Northeastern Blvd. Suite 29A Nashua, NH 03062 Phone: 603.881.9031 Fax: 603.881.9035

For Management Applications

- Does an SMI-S client write directly to the management application database using the CIM data model?
- If so, can other CIM-built applications use the database without exporting it to other formats or modifying the data?
- Are hosts, HBAs, fabric switches, and disk subsystems from different vendors represented in exactly the same way, according to SMI-S, for common, normalized management?
- Is SMI-S support included as a standard product capability, or is it priced separately as an option?
- Has the management application been used to demonstrate SMI-S compliance at interoperability plugfests?

For Both

- To what degree has the vendor implemented the functionality called for in SMI-S?
- Has the product passed SNIA's ICTP certification process?

Data Mobility Group