Systems Management Architecture for Server Hardware (SMASH) Command Line Protocol (CLP) Architecture White Paper

Version 1.0.1

Status: Informational

Publication Date: October 20, 2006

DSP2001

Version 1.0.1
Publication Date: November 24, 2005
DSP2001
Status: Informational

Abstract

The Systems Management Architecture for Server Hardware (SMASH) is an initiative that represents a suite of specifications which standardize the manageability interfaces for server hardware. The suite of specifications lay out an architectural framework, interfaces in the form of protocols, addressing and profiles for server hardware.

This document is an architectural white paper describes the concepts used in SMASH CLP.

Acknowledgments

The following persons were instrumental in the development of this specification:
Bob Blair, Newisys; Greg Dake, IBM; Jon Hass, Dell; Jeff Hilland, HP (editor); Steffen Hulegaard, OSA Technologies; Arvind Kumar, Intel; Jeff Lynch, IBM; Aaron Merkin, IBM; Christina Shaw, HP; Enoch Suen, Dell; Michael Tehranian, Sun; Perry Vincent, Intel, John Leung, Intel; Khachatur Papanyan, Dell; Reddy Dasari, Dell.

Table of Contents

Abstract 3
Acknowledgments 3
1 Introduction 8
1.1 Target Audience 8
1.2 Related Documents 8
1.3 Terminology 9
1.4 Acronyms and Abbreviations 9
2 Architecture Overview 11
2.1 Principal Goals 11
2.2 Service Model 11
2.2.1 In-Band vs. Out-of-Band 12
2.2.2 In-Service vs. Out-of-Service 12
2.2.3 Combined Service Model 12
3 Server Management CLP Architecture 14
3.1 Architectural Model 14
3.2 Client 15
3.2.1 User 16
3.2.2 Transport Client 16
3.3 MAP 16
3.3.1 Management Service Infrastructure 17
3.3.2 Client Object Manager Adapter 17
3.3.3 External Authentication, Authorization, Audit Service 18
3.4 Managed System 18
3.4.1 Managed Element 19
4 Server Management Models ..... **Error! Bookmark not defined.**
4.1 Operation Model 20
4.1.1 MAP Responsibilities 20
4.1.2 Operation Handoff 21
4.1.3 Operation Queue 21
4.1.4 Multi-session capabilities 22
4.1.5 Resource Handling 22
4.2 Boot Model ..... **Error! Bookmark not defined.**
4.2.1 Boot Configuration ..... **Error! Bookmark not defined.**
4.2.2 Boot Source ..... **Error! Bookmark not defined.**
4.2.3 Boot Configuration Management ..... **Error! Bookmark not defined.**
4.3 Firmware Update Model ..... **Error! Bookmark not defined.**
4.3.1 Firmware update mechanism ..... **Error! Bookmark not defined.**
4.3.2 Firmware Update properties ..... **Error! Bookmark not defined.**
4.3.3 Firmware Update Support for Multiple Firmware Versions **Error! Bookmark not defined.**
4.3.4 Firmware Update Operation ..... **Error! Bookmark not defined.**
4.4 Discovery ..... **Error! Bookmark not defined.**
5 Profiles 23
6 Target Addressing 25
6.1 Addressing Architecture 25
6.2 UFcTs and UFiTs 25
6.3 Target Addressing in the CLP 25
7 Security 26

7.1 Transport Considerations 26
7.2 User Account Management 26
7.3 Audit 27
7.4 CLP Service & MAP Management 27
8 Conclusion 29

List of Figures

Figure 1 Service Model 13
Figure 2 SM CLP Architecture Model 14
Figure 3 Example MAP Implementation Architecture 15

1 Introduction

This document is an introduction into the architectural framework required for managing server hardware in the data center today. This document lays forth the basic principles required for understanding and implementing the Systems Management Command Line Protocol (SM CLP) as specified by the DMTF. Specifically, this group of documents includes the SMASH CLP Architecture White Paper (this document), Server Management Command Line Protocol Specification [4], Server Management Managed Element Addressing Specification [2], SMASH Implementation Requirements [3], and the Server Management CLP to CIM Mapping Specification [5].

The focus of the SMASH architecture is to enable the management of the server resources in a standard manner across any Manageability Access Point implementation, regardless of operating system state.

1.1 Target Audience

The intended target audience for this document is readers interested in understanding the Server Management Command Line Protocol (SM CLP) Specification, the Server Management Managed Element Addressing Specification or Server Management Architecture in general.

1.2 Related Documents

[1] Common Information Model (CIM) Schema, V2.14, December, 2006 - Downloadable from http://www.dmtf.org/spec/cim.html

[2] "SM Managed Element Addressing Specification", V1.0.0, DSP0215, 2005, DMTF SMASH – Downloadable from http://www.dmtf.org/standards/smash

[3] "SMASH Implementation Requirements", DSP0217 V1.0.0, 2006, DMTF SMASH – Downloadable from http://www.dmtf.org/standards/smash

[4] "Server Management Command Line Protocol Specification", V1.0.0, DSP0214, 2005, DMTF SMASH – Downloadable from http://www.dmtf.org/standards/smash

[5] "SM CLP to CIM Mapping Specification", V1.0.0, DSP0216, 2006, DMTF SMASH – Downloadable from http://www.dmtf.org/standards/smash

[6] "Posix Utility Conventions", The Open Group Base Specifications Issue 6, IEE Std 1003.1, 2004 Edition. Downloadable from http://www.opengroup.org/onlinepubs/009695399/basedefs/xbd_chap12.html

137 ## 1.3 Terminology

Term	Definition
Administrator	A person managing a system through interaction with management clients, transport clients and other policies and procedures.
Client	Any system that acts in the role of a client to a MAP.
Command Line Protocol (CLP)	The command line protocol defined by the Server Management Architecture for Server Hardware, used for managing systems.
Command Processor Engine	The logical entity within a MAP responsible for parsing incoming commands and returning responses.
In-Band	Management that operates with the support of hardware components that are critical to and used by the operating system
In-Service	Management that operates with the support of software components that run concurrently and are dependent on the operating system.
Manageability Access Point (MAP)	A collection of services of a system that provides management in accordance to specifications published under the DMTF Server Management Architecture for Server Hardware initiative.
Managed Element	The finest granularity of addressing which can be the target of commands or messages, or a collection thereof.
Managed Element Access Method	The method by which a Managed Element performs a unit of work.
Managed System	A collection of Managed Elements that comprise a Computer System for which a MAP has management responsibilities.
Out-of-Band	Management that operates with hardware resources and components that are independent of the operating systems control
Out-of-Service	Management that operates with the support of software components that require the operating environment to be put out-of-service and the system be placed into an alternate management environment. In this state, the operating system is not available
Target Address Scheme Resolution Service	The entity responsible for discovering, enumerating and determining the addresses of Managed Elements within the MAP.
Transport	The layers of the communication stack responsible for reliable transportation of commands and message from the Client to the MAP
User	The set of Administrators and Management Clients which interact with the Transport Client in order to manage a Managed System through a Manageability Access Point.

138 ## 1.4 Acronyms and Abbreviations

Term	Definition
CIM	Common Information Model
CIM Server	Common Information Model Server

CLP	Command Line Protocol
DMTF	Distributed Management Task Force
MAP	Manageability Access Point
ME	Managed Element
NIC	Network Interface Card
SSHv2	Secure Shell Version 2
SMASH	Systems Management Architecture for Server Hardware
SM CLP	Server Management Command Line Protocol
UFiP	User Friendly Instance Path
UFcT	User Friendly Class Tag
UFiT	User Friendly Instance Tag

2 Architecture Overview

Enterprise Server Management in today's data center is comprised of a rich set of tools and applications which administrators can use to manage the data center. In many cases, these tools are specialized and adapted to each individual environment, installation and product in the data center.

Currently, the richness of the CIM Schema provides a feature rich systems management environment. In its current form, it also places an additional burden on those vendors attempting to implement the CIM Schema & WBEM Protocols to support server hardware management in the Out-of-Band and Out-of-Service scenarios. This has resulted in lack of interoperability in the server hardware management arena, particularly in the out-of-band and out-of service cases. In addition, the resulting Out-of-Band and Out-of-Service management solutions are different from the operating system's representation and management of the server.

The Systems Management Architecture for Server Hardware initiative supports a suite of specifications which include architectural semantics, industry standard protocols, and profiles to unify the management of the data center. By creating industry standard protocols, interoperability is guaranteed over the network and the syntax and semantics of those protocols are guaranteed to be interoperable by compliant products which adhere to those standards. By basing it on the CIM Schema, the SM CLP leverages the richness of CIM. By creating industry standard profiles, the richness of the CIM Schema can be applied in a consistent manner so that systems offered by different vendors will be represented in similar ways.

Extra emphasis has been placed in the development of the SM CLP architecture to enable lightweight implementations which are architecturally consistent. This has been done to enable a full spectrum of server implementations without sacrificing the richness of the CIM heritage. This includes software only solutions and small footprint firmware solutions. Emphasis has been placed on ensuring that these implementations will be interoperable, regardless of implementation, CPU architecture, chipset solutions, vendor or operating environment.

2.1 Principal Goals

One goal of the Server Management Command Line Protocol (SM CLP) Architecture is to enable the same interfaces regardless of server state. To this end, a Service Model has been included in Section 2.2.3 to illustrate that, regardless of Service Access Point or Operating System Service state, the same protocols should be able to used for Systems Management.

Another goal of the SM CLP Architecture is to enable the same tools, syntax, semantics and interfaces to work across a full range of server products – stand alone systems, rack mounted servers, blades, Telco servers, partitionable as well as virtual and redundant servers. Therefore, we have encompassed considerations for these products in our initial architecture and will include support for them in the on-going profile development effort.

2.2 Service Model

Fundamental to the SM CLP Architecture is the underlying goal to unify the experience achieved through out-of-band mechanisms with those available via the operating system. To achieve this goal, the SM CLP Architecture contains a model to describe these terms (In-Band, Out-of-Band, In-Service, Out-of-Service) and to relate them to management today.

2.2.1 In-Band vs. Out-of-Band

A key concept in understanding the Service Model is an understanding of the terms In-Band and Out-of-Band and how they are used within the context of Server Management.

In-Band Management operates with the support of hardware components that are critical to and used by the operating system. An example would be a general purpose NIC available through the operating system.

Out-of-Band Management operates with hardware resources and components that are independent of the operating system. These resources are dedicated to systems management and allow management of system hardware components independent of their state. Typically, they are also available when the operating system is available & can interact with the operating system. An example would be a service processor or baseboard management controller.

2.2.2 In-Service vs. Out-of-Service

Dependency on the operating system service state is described by the terms "In-Service" and "Out-of-Service".

In-Service management operates with the support of software components that run concurrently and are dependent on the operating system. This is often provided through a service or process within the operating system.

Out-of-Service management operates with the support of software components that require the operating environment to be put out-of-service and the system be placed into an alternate management environment. In this state, the operating system is not available.

2.2.3 Combined Service Model

By combining the operating system service dependency with the management access method ("In-Band"/"Out-of-Band"), we can achieve the following Service Model matrix. This service model is useful in understanding what is meant by unifying the In-Service/Out-of-Service and In-Band/Out-of-Band management experience. This should help vendors of manageability components, software and solutions to understand the goal and deliverables encompassed by the SM CLP Architecture. Included in the Service Matrix are examples of solutions for that part of the matrix.

Below, in Figure 1, is the SM CLP Architecture Service Model. The horizontal axis is the OS-Dependency and refers to the state of the normal operating system environment on the management environment. The vertical axis represents the physical location of the Manageability Access Point. Note that Service Processor is terminologically equivalent to a firmware or software based management controller or service.

		OS-Dependency	
System HW dependency / MAP Location/ Access	*Main System Hardware*	**Out-of-Service Management** *Pre-boot BIOS/EFI* *Provisioning OS* *Diagnostic Environment*	**In-Service Management** *OS-Resident Agent*
	Auxiliary Service Hardware	**Out-of-Band Management** *Service Processor* *Chassis Management Module* *Shelf Manager*	

Figure 1 Service Model

3 Server Management CLP Architecture

In order to provide server management standardization, it is necessary to develop an abstract model that describes server management regardless of the actual implementation. This is necessary to provide a common vocabulary and to provide a common base of understanding. It is also used to illustrate the access points where interoperability is guaranteed as well as to show semantically visible components and interfaces.

The goal of the architecture is also to describe server management in abstract terms regardless of server type, topology & framework. This means it must be implementation agnostic as well as span the spectrum from small stand-alone servers, to large partitionable servers and encompass topologies such as blades and racks as well as unique segments such as industry standard servers, telecommunications and mission critical high-end servers.

3.1 Architectural Model

This section introduces the overall SM CLP Architecture Model (see Figure 2). The terms used in this model are defined in the following sections. The dotted lines in this model indicate the protocols and transports that are externally visible. These are the communication interfaces between the Manageability Access Point (MAP) and the Client and represent data that flows across the network, for example. The solid lines indicate semantically visible interfaces. The packets, transports, and interfaces are not externally visible but the fact that they are separate components with their own semantics is visible. The functional implications which are noticeable by the Client need to be accounted for in order to have a complete model.

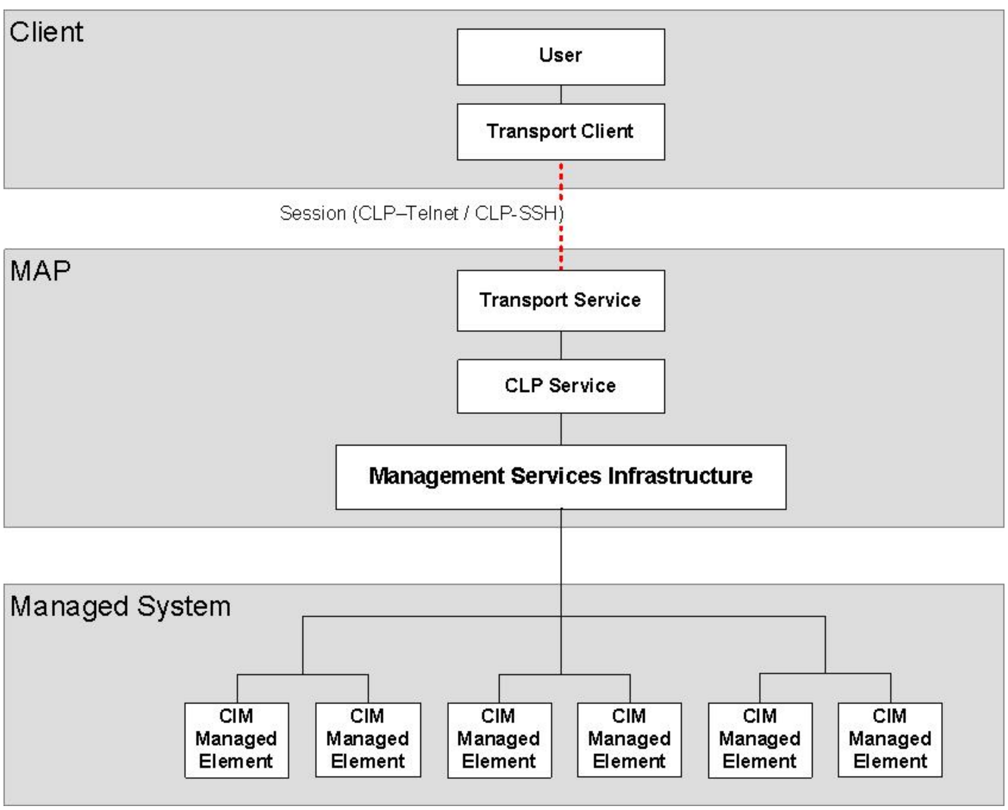

Figure 2 SM CLP Architecture Model

Figure 3 contains an example implementation that provides an emphasis on components within the MAP which are noticeable when implemented within a WBEM context. While the entities described are not required to exist as independent entities, their existence can be determined by the syntax and semantics of the interface between the MAP and the Client. This figure expands on the architecture model, exposing the detailed, identifiable portions of the Client and the MAP. This includes the Transports and a detailed User model to indicate support by the SM CLP of both a direct human Administrator and a Management Client. It also indicates that Authentication, Authorization and Audit components exist within the map and, therefore, are expected to be accessible through the protocols. In addition, Operation Invocation Engine and the Target Address Scheme Resolution Services indicate that both the operations within the MAP and the addressing & discovery within the MAP are distinct with their own operational semantics. Note that while only one Managed System is shown, managing multiple Managed Systems from one MAP is supported by the SM CLP architecture.

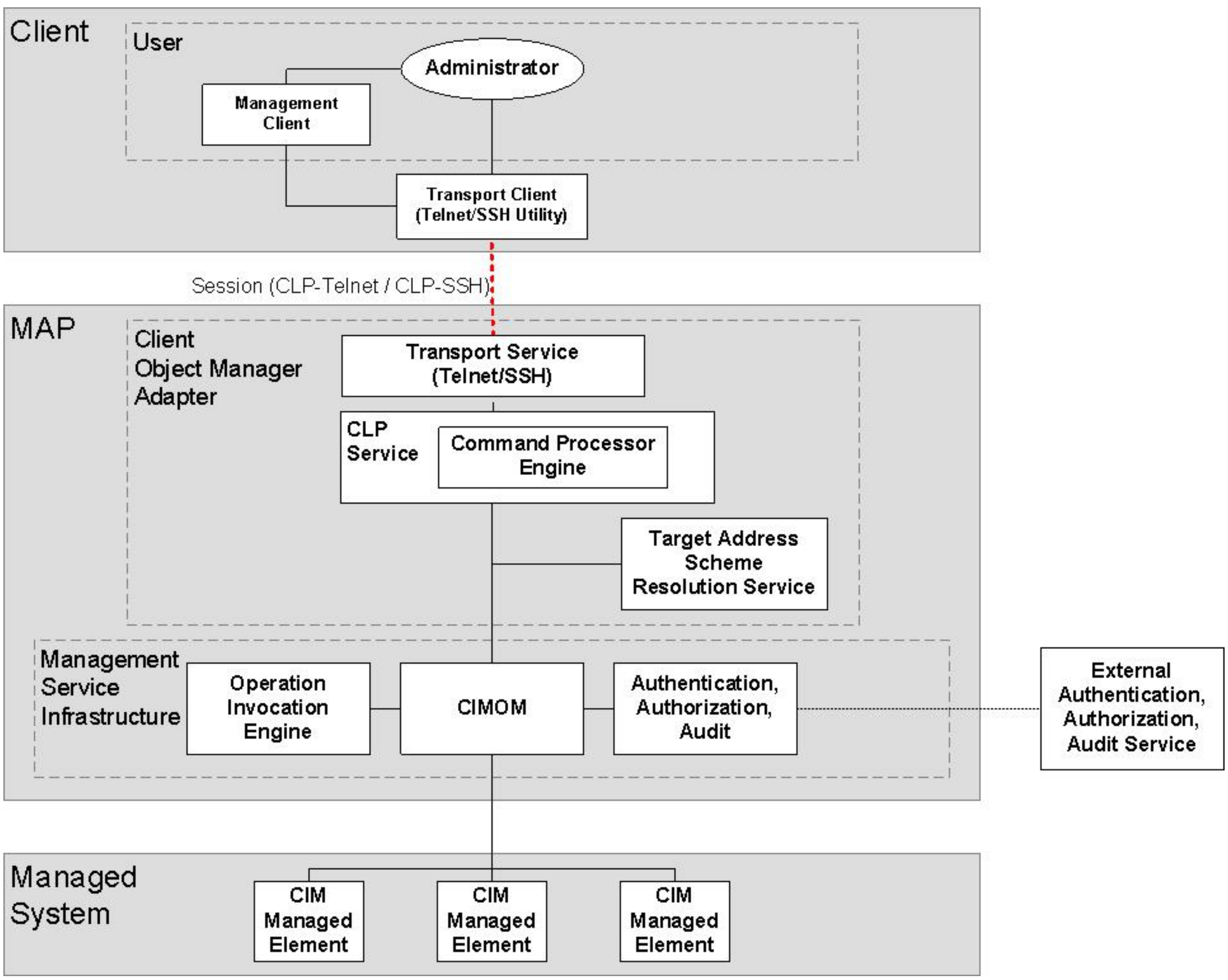

Figure 3 Example MAP Implementation Architecture

3.2 Client

A Client is a logical component that manages a system via a Manageability Access Point (MAP). A Client may run on a management station or other system.

A Client is responsible for:

- Providing an interface to the functionality provided by the MAP in a form consistent with the SM Architecture.
- Accessing a MAP using one of the SM CLP Architecture defined management protocol specifications. This entails interacting with the MAP through the following process:
  - Initiating a session with a MAP.
  - Transmitting protocol-specific messages to the MAP.
  - Receiving protocol-specific output messages from the MAP.

3.2.1 User

The Command Line Protocol (CLP) User in this model represents an instance of a Client which transmits and receives CLP compliant messages. The CLP is part of the SM CLP Architecture. It is intended to either be a human or script interacting with a terminal service such as telnet or sshv2. For more information on the CLP, see [6].

3.2.1.1 Management Client

A Management Client represents a program of some type, such as a script or application, that initiated management requests to the Transport Client and handles responses from the Transport Client. Interaction between the Management Client and the Transport Client is in the form of SM CLP messages. Interaction between the Administrator and the Management Client is outside the scope of this document.

3.2.1.2 Administrator

This represents the human interacting with either the Management Client or directly with the Transport Client. Interaction between the Administrator and the Transport Client is in the form of SM CLP messages.

3.2.2 Transport Client

The Transport Client represents the endpoint of the transport and lower layer protocols with which the User interacts. It initiates and maintains the transport session with the Transport Service in the MAP. This includes the transport session establishment and authorization. Authentication is expected to take place either during or after Transport session establishment but before CLP Session establishment, as indicated later in this specification.

The CLP specification contains mappings for SSHv2 and Telnet, but other transports are possible.

3.3 MAP

The Manageability Access Point ("MAP") is a network-accessible service for managing a Managed System. A MAP can be instantiated by a Management Process, a Management Processor, a Service Processor or a Service Process.

The MAP is responsible for:

- Managing the Session between the MAP and the Client. The MAP is considered the endpoint for the transport protocol.

- Interpreting the incoming protocol-specific messages and seeing that a response is transmitted.
- Returning protocol-specific output messages to the Client containing status and result data.

The MAP fulfils these responsibilities by utilizing components contained within the MAP. Note that the interface between the Managed Elements (ME) and the MAP is outside of the scope of the SM CLP Architecture. The interfaces within the MAP are outside of the scope of the SM CLP Architecture.

The MAP contains the following major components, which are discussed in the following sections:

- A Client Object Manager Adapter, provides adapts the CLP Messages into CIM operations that the Management Service Infrastructure can act upon.
- The Management Service Infrastructure, which provides management access to the instrumentation of the Managed Systems.

3.3.1 Management Service Infrastructure

The Management Service Infrastructure is a logical entity that contains the core services set of the MAP that implement a CIM Server. It is primarily comprised of the functions described below.

3.3.1.1 CIMOM

This is the components of the Management Service Infrastructure that handles the interaction between the Client Object Manager Adapter and the Providers. It supports services such as the Operation Invocation Engine & the Authentication, Authorization & Audit components.

3.3.1.2 Operation Invocation Engine

The Operation Invocation Engine is responsible for understanding the management requests and tracking the initiation, interim status and completion of operations resulting from those requests on Managed Elements. A major component of the Operation Invocation Engine is the Operation Queue. This is the queue of all of the operations submitted to the MAP. Operations are discussed in more detail in Section 4.

3.3.1.3 Authentication, Authorization, Audit

This entity is responsible for coordinating the authentication, authorization and auditing within the MAP. This includes coordination of transport session establishment, local account information and the access permission required for MAP operations. It also is responsible for coordination of audit information of the operations and tasks taking place within the MAP. Note that this is a service internal to the MAP and does not include any external service components or coordination.

3.3.2 Client Object Manager Adapter

This represents the collection of entities required to process the SM CLP commands and responses and, as required by the messages, interact with the Management Service Infrastructure to accomplish the requests and produce responses. It consists of the Transport Service, CLP Service, Command processor Engine and Target Address Scheme Resolution Service.

3.3.2.1 Transport Service

This represents the transports and lower layer protocols on which the CLP resides. This includes the transport session establishment and authorization. Authentication is expected to take place either during or after Transport session establishment but before CLP Session establishment, as indicated later in this specification.

It also represents the entity which encrypts/decrypts the data stream. This happens as part of the transport mechanism in this architecture. For instance, SSHv2 has encryption mechanisms.

The CLP specification contains mappings for SSHv2 and Telnet, but other transports are possible.

3.3.2.2 CLP Service

This represents the endpoint of the CLP within the MAP. Commands will be received here and turned into internal operations within the MAP. This entity is responsible for receiving messages and transmitting responses which are compliant with the SM-CLP Specification[4].

The interface between the CLP Service and the Management Service Infrastructure is implementation dependent and thus the interface itself is out-of-scope of the Systems Management Architecture for Server Hardware.

3.3.2.3 Command Processor Engine

This represents the entity which parses incoming commands and handles responses of the CLP. It is responsible for ensuring that the SM CLP messages are compliant with the grammar in the SM-CLP Specification[4].

3.3.2.4 Target Address Scheme Resolution Service

This entity is responsible for discovering and enumerating the Managed Elements within the local domain, for maintaining the addressing and naming structure of the local domain, and coordinating this information with the operation invocation engine. This Service is required to implement and adhere to the rules and grammar specified in the Server Management Managed Element Addressing Specification[2].

3.3.3 External Authentication, Authorization, Audit Service

The External Authentication, Authorization, Audit Service represents the entity which establishes and coordinates the authentication, authorization and auditing information outside of the MAP. Examples of services that it may coordinate are keys, certificates, user accounts, passwords and privileges. The instantiation of any global Authentication, Authorization, Audit Service is outside of the current scope of the SM CLP Architecture. In addition, the interface between the MAP and the Security Service is outside of the current scope of the SM CLP Architecture. Note that this is distinct from the Authentication, Authorization, Audit component of the MAP itself since (see Section 3.3.1.3) it is an external service and not contained within the MAP.

3.4 Managed System

A Managed System is a collection of Managed Elements that comprise a Computer System for which the MAP has management responsibilities. The Managed System may sometimes be referred to as a host, node, server, or platform. A Managed System could represent multiple types of systems, such as stand-alone, rack, blade or virtual systems.

There may be one or more Managed Element and/or Resources, or collections thereof, managed by a single MAP. Consequently, there may be multiple servers in a Managed System. There may be more than one Managed System within the domain of any MAP.

Each Managed Element within the Managed System could contain subcomponents, sub-targets or resources within that individual Managed Element.

3.4.1 Managed Element

Managed Elements are the targets, components, resources, collections or logical entities within a Managed System which the operations will manipulate.

Specific interfaces for Managed Element access are outside of the scope of the SM CLP Architecture.

4 Operation Model

This section contains information relevant to operation handling within the MAP. It will cover MAP responsibilities, operation handoff, queue depth issues, issues on multi-session support, operation visibility, communication between MAPs and resource handling.

It is important to understand that in the MAP operation model, the term operation is often used. The reader should understand CIM_Job (Core Schema), CIM_JobQueue and be familiar with them. The terms operation and job are synonymous with respect to this specification.

4.1 MAP Responsibilities

The Manageability Access Point (MAP) has several responsibilities to the Client. Some of these may appear intuitive to some readers, but for purposes of clarity they will included here.

MAPs are responsible for managing the elements for which they claim responsibility. This does not imply that they will actually execute the method or modify the property included in the operation, but MAPs are responsible for ensuring that they are the focal point of the interaction and responsible for tracking the operation.

The MAP is responsible for ensuring the command is syntactically correct. It may pass the parsing to further levels within the MAP or System, but it is the MAP that has the responsibility for ensuring that the implementation complies with the protocol.

The MAP is responsible for command, message and operation handling. It may delegate the actual operation but it is responsible for handling commands and messages, turning them into jobs or operations, tracking operations and manipulating the operations (including completing, canceling, removing, or logging).

The MAP is responsible for determining if the specified ME is in the scope of the MAP. Operations which target MEs which are not within the MAPs scope should result in the appropriate error syndrome.

The MAP is responsible for determining if access to the ME is allowed. This includes, but is not limited to, authorization determination (to ensure that the user account and access right combination will allow access to the ME) and determination that the ME is in a state where the operation can be initiated.

The MAP is also responsible for determining if the operation or property modification is valid for this Managed Element and if the operation or property modification is a valid request. It is the MAP's responsibility to ensure that any such request takes place as indicated. The MAP ensures that the request is properly formed and conveyed, but relies on the feedback from the ME for the assessment of operation validity.

The MAP is responsible for maintaining any session context required. Since the MAP contains the connection with the transport, any session related information, such as current default target, or option settings, such as language, locale or output format are required to be maintained by the MAP. For protocols that do not maintain session state or do not allow connections to persist, this is not required.

The MAP is responsible for maintaining the local UFiT address space. This includes any aliases or any OEM extensions. It is responsible for ensuring the creation of the address space of Managed Element instances and mediating commands and messages into operations on those elements.

4.2 Operation Handoff

Operations within the MAP are not directly visible to the Client. The fact that they exist, are initiated, can be cancelled, can complete and can be deleted is visible. In addition their status can be retrieved.

Operations can only be created using commands or messages. The MAP exposes one and only one identifiable, traceable operation for any single, valid command. If an implementation spawns multiple activities in order to process a single command or message, then all of the activities are related to the single job identifier created when the operation was initiated and it is the responsibility of the MAP to track the multiple activities and relate them to the single operation.

All operations have identifiers. The CIM_ConcreteJob class is used to represent operations, so the identifier is that of a CIM_ConcreteJob instance. The term Operation ID (OPID) or Job ID is used interchangeably to represent the identifier of that CIM_ConcreteJob instance. Note that OPIDs are returned when the operation is spawned, regardless of the duration of the operation. The status of the operation can be retrieved with a command or message using the OPID. The MAP must keep track of all active operations.

When an operation is complete, the settings for the operation will determine if that instance represented by the OPID will persist or will immediately be recycled. TimeBeforeRemoval from CIM_ConcreteJob is used to determine the amount of time that an operation will persist in the operation queue.

All operations must be able to handle a cancellation request. Sometimes the response to the cancellation will be an error, such as in the case of an operation that cannot be undone, such as an operation that has already taken place or that cannot be stopped part of the way through, such as turning the power off or resetting a system.

Any operation which is longer than the typical command-response time will be run asynchronously and an operation identifier will be returned. The Client can then determine the status of the operation and whether or not the operation is complete. This can be done through a query operation on the operation queue using the OPID. The operation queue can also be queried to find out the maximum operation queue depth, or if the queue is full.

4.3 Operation Queue

The architecture contains an operation Service within the MAP which logically contains an operation queue. This is a FIFO queue which contains all of the operations to be processed within the MAP. All current sessions submit operations to this single queue. The Operation Queue is modeled using CIM_JobQueue. The CLP [4] provides access to the capabilities of this queue and the SM Profiles [3] for the MAP indicate the properties available. The Properties of the Operation Queue are expected to vary from implementation.

Ordering is with respect to command initiation and is implied by the queue. Ordering of operation initiation is guaranteed within a session but no such guarantee is made between sessions.

The MAP's operation queue depth varies from MAP to MAP. The minimum acceptable operation queue depth is equal to one operation. Some implementations may support multiple outstanding operations on a single session; others may not. Should the queue become full, the MAP is responsible for communicating this resource constrained condition distinct from other error conditions. This is communicated through error codes. For instance, an error that indicates resource busy is distinct from one that indicates the job queue is full. For a complete description on the error semantics, see the SM CLP Specification [4].

The MAP must be able to indicate to the Client the maximum operation queue depth supported by the MAP as well as the number of current outstanding operations. This is done through the modeling of the Operation Queue within the MAP.

Detailed information of individual operations on the operation queue, such as is available through CIM_ConcreteJob, can be queried through the MAP by directing queries at individual operations.

4.4 Multi-session capabilities

An important aspect of MAP operations management is to be able to support simultaneous sessions through the MAP. Implementations are not required to support more than one session simultaneously. However, implementations are expected to exist that support many simultaneous sessions. Therefore, the SM CLP Architecture supports multiple concurrent sessions.

The number of ports offered to transports from the Management Services Core for each protocol supported must be at least one per protocol supported. The MAP utilizes the error syndromes of the transport and subsequent layers when handling out of resource conditions (such as no more ports available), attempting to connect to the wrong port, or not supporting the requested transport.

Another aspect of multi-session capabilities is the ability for operations to be visible regardless of the transport that initiated them. This implies that there is one global operations (job) queue per MAP. The MAP is responsible for routing the results of operations to the appropriate session. But if the command or message spawns an operation, then any session should be able to discover the details about the operation in question, by querying the operation using the OP ID. This is helpful for a number of reasons. For example, if an operation is spawned, the Client may disconnect and then query the status of that operation at a later time, provided the Client has retained or can discover the identifier for that operation.

4.5 Resource Handling

The SM CLP Architecture contains mechanisms that enable resource handling.

In this version of the SM CLP Architecture, the manipulation of resources in the server is limited to treating the server as a collection of Managed Elements. This allows the MAP to be able to create and modify configurations of the system and the establishment of boot order as well (see Section **Error! Reference source not found.**)

The administration and configuration of complex systems, such as those with shared resources, often requires the locking of a ME in order to manage the ME or to ensure that the ME is assigned to one and only one system. Direct support of these mechanisms is not included in this version of the architecture. Because direct support is not required, the mechanism for handling resource locking is outside of the scope of this specification.

5 Profiles

DMTF Management Profiles provide the information model definitions for manageability content and architecture models mapping computer hardware in a way that is consistent between different implementations. These profiles combine to ensure that implementations supporting the management of similar components provide a consistent representation of the components. Individual implementations support the profiles that are appropriate for the hardware and software configurations they manage.

CLP implementations are dependent on underlying modeling of system components. In order to achieve an interoperable CLP, the information models utilized are required to be consistent across implementations.

The SMASH Architecture identifies a subset of DMTF Management Profiles that are appropriate for its targeted management domain. The following is a list of DMTF Management Profiles that are included in the SMASH CLP Architecture with a brief description of the functionality provided by each. As noted above, implementations will select the DMTF Management Profiles that are appropriate for their environment and therefore not all profiles will be supported by all implementations.

- DSP1004, the *Base Server Profile* is a top-level profile providing the ability to manage server systems.
- DSP1012, the *Boot Control Profile* provides the ability to manage boot configurations of a system.
- DSP1018, the *Chassis Manager Profile* provides the ability to represent the chassis manager of a modular system.
- DSP1005, the *CLP Service Profile* provides the ability to manage an implementation of the SMASH CLP architecture.
- DSP1022, the *CPU Profile* provides inventory, status, and state information for processors of a managed system.
- DSP1019, the *Device Tray Profile* provides the ability to manage shared media trays in a modular system.
- DSP1037, the *DHCP Client Profile* provides the ability to manage the DHCP client configuration of a managed system.
- DSP1038, the *DNS Client Profile* provides the ability to manage the DNS client configuration of a managed system.
- DSP1014, the *Ethernet Port Profile* provides inventory, status, and state information for the Ethernet interfaces of a managed system.
- DSP1013, the *Fan Profile* provides inventory, status, and state information for fans of a managed system.
- DSP1036, the *IP Interface Profile* provides the ability to manage the configuration of IP interfaces of a managed system.
- DSP1008, the *Modular System Profile* provides the ability to manage modular enclosures and contained components.
- DSP1020, the *Pass-Through Module Profile* provides inventory, status, and state information for pass-through modules of a managed system.

- DSP1011, the *Physical Asset Profile* provides the ability to report physical asset information including capacity and FRU information for components installed in a managed system.
- DSP1027, the *Power State Management Profile* provides the ability to query and manage the power state on a managed system.
- DSP1015, the *Power Supply Profile* provides inventory, status, and state information for power supplies of a managed system.
- DSP1010, the *Record Log Profile* provides the ability to retrieve error and event log information for managed systems.
- DSP1039, the *Role Based Authorization Profile* provides the ability to manage rights granted to security principals through role membership.
- DSP1009, the *Sensors Profile* provides the ability to query sensor status and state information for component and system sensors.
- DSP1021, the *Shared Device Management Profile* provides the ability to control access to shared devices in a modular system.
- DSP1034, the *Simple Identity Management Profile* provides support for basic account management, including account creation and deletion.
- DSP1007, the *SM CLP Admin Domain Profile* is used to model the administrative domain of an SM CLP implementation.
- DSP1006, the *SMASH Collections Profile* provides support for collecting settings, capabilities, and other Managed Elements to simplify management access through an SM CLP implementation.
- DSP1023, the *Software Inventory Profile* provides the ability to view the firmware, device drivers, BIOS, and other software installed on a system and its components. It also provides the ability to view the software available for installation on a system and its components.
- DSP1025, the *Software Update Profile* provides the ability to perform software installation, upgrades, and downgrades on a system and its components.
- DSP1017, the *SSH Service Profile* provides the ability to manage the configuration of an SSH service and client sessions.
- DSP1026, the *System Memory Profile* provides inventory, status, and state information for the main system memory of a managed system.
- DSP1016, the *Telnet Service Profile* provides the ability to manage the configuration of a Telnet service and client sessions.
- DSP1024, the *Text Console Redirection Profile* provides the ability to start and stop text console redirection over the interfaces of a managed system.

6 Target Addressing

The primary goal of the target addressing scheme is to provide an easy-to-use way to accurately address CIM objects.

The target address term of the CLP syntax in this architecture is extensible. Addressing for version 1.0.0 is fully described in the Server Management Managed Element Addressing Specification [2].

The addressing scheme provides a unique target for CLP commands. The scheme is finite for parsing target names and unique for unambiguous access to associated instance information needed to support association traversal rooted at the MAP AdminDomain instance.

6.1 Addressing Architecture

The Addressing rules are applied to the CIM aggregation and association relationships to ensure that each fully qualified instance name is unique. This is accomplished by requiring that an instance name is unique within its immediate container. The exact containers which Managed Elements are allowed to be in is defined fully in the Server Management Managed Element Addressing Specification [2].

The addressing rules, specified in the Server Management Managed Element Addressing Specification [2] contain the detail necessary to fully understand the formulation of Addresses and valid Target names for the CLP. This section contains a brief overview of the Addressing architecture.

6.2 UFcTs and UFiTs

A User Friendly class Tag (UFcT) convention is defined to simplify long complex CIM class names without compromising object references, class properties, associations or behavior. This provides a more user friendly experience for the Client (human end user). UFcTs are simple synonyms of specific CIM classes used in Server Management Profiles.

A User Friendly instance Tag (UFiT) is formed by taking a User Friendly class Tag and combining it with a non-negative integer suffix.

UFcTs are used to represent CIM classes. UFiTs are used to represent a specific Managed Element.

UFiTs are then combined in a manner similar to a file directory structure to form a User Friendly instance Path (UFiP) - see Section 6.3 below. This structure is based on the collection of, associations between and aggregations of Managed Elements.

6.3 Target Addressing in the CLP

The Server Management Command Line Protocol will accept UFiTs which are formed into a UFiP. The SM CLP also accepts other target address constructs, such as those used to select all instances of a class . MAP's will support a number of standard, default UFiTs that are consistent with the SM CLP Architecture Addressing rules contained in the SM Managed Element Addressing Specification[2] and the Server Management Profiles[3].

7 Security

Security is an important consideration when providing server management. The In-Service/In-Band aspects of server management have been well explored through various standards and implementations, but the cross-section of Out-of-Band and Out-of-Service dimensions requires unique considerations.

While there are many aspects to security, it is important to focus on a finite but achievable list for the SMASH specifications. Specifically, these are transport considerations, logon, account properties, account management, credential management and the management of the MAP itself.

7.1 Transport Considerations

Implementations of the SM CLP Architecture may support Telnet or SSHv2 as the transport for the CLP. The detailed requirements for each transport protocol are detailed in the CLP specification [4]. Information on the exact specifications supported is contained in the SM CLP specification as well as any other information required to implement the CLP over these specific transports. Note that the Architectural Model described in Section 3.1 shows how these transports are included in the architecture.

Some transports contain their own authentication mechanisms, such as key-exchange in SSHv2. Others rely on an intermediate authentication mechanism. If the transport supplies an authentication mechanism, it should equate to a user configured in the MAP which will then be used for the session's authorization information. If another authentication mechanism is used, such as in the case of Telnet, the logon mechanism is expected to be user based, so the user name and password used to authenticate the Telnet session can be used to determine authorization of the commands of the CLP. For instance, key exchanges equate to user names and passwords. The user name and password used to authenticate the connection, or the user name and password associated with the key information, is the user name and password used to determine authorization of the commands of the CLP. Regardless, the CLP Service expects authentication to be performed before a session is established between the CLP and the Client. The CLP Session established is expected to pass an user account name as described in Section 7.2 to the MAP for use in authorizing commands.

For transports that do not contain an adequate encryption protocol, it is recommended that they be layered upon a protocol that supports strong encryption. It should be apparent to the reader that the vulnerability of the MAP is equivalent to the vulnerability of the transport protocols, thus in order to prevent intrusion the MAP should support secure transports. In the case of Telnet, any mapping of Telnet over a protocol such as TLS or SSL is outside the scope of this specification and the SM CLP Architecture. SSHv2 includes automatically negotiated encryption, so any layering is not required since encryption is inherent to the protocol.

7.2 User Account Management

User account management is an important aspect to the security of the SM CLP Architecture. Since the user account used for authentication is expected to be the same account used for authorization, it is important to understand the user account model.

User accounts can be created and assigned to a CLP user group.

There are three CLP user groups defined in the architecture. Implementations are required to support at the Read Only and Administrator groups. Implementations may support more groups or definable groups. If a user belongs to more than one group, the group with the most privileges is the group used for authorization of commands.

- Read Only - Members of this group are only able to perform read operations. This includes retrieval of data and the ability to perform non-invasive commands such as help, change default target and change session options.
- Operator – Members of this group are able to perform read, write and execute operations. Consequently, members of this group can query data. In addition, they can change the state of Managed Elements. They can change setting data or settings or collections. They cannot create, delete or instances or properties directly.
- Administration – Members of this group have read, write, create, delete and execute privileges. Members of this group have all access rights. Members of this group can create, delete or modify users and assign them to groups, unless prohibited by the Authentication, Authorization, Audit Service. Members of this group can also create and delete instances, such as log records.

At this time, there are no per target access control lists defined in the architecture.

The MAP must support the methods and properties to add accounts, remove accounts, show account information and modify accounts as follows:

- Add Account – Create accounts and set their initial state and conditions.
- Remove Account – remove the account completely.
- Show Account – retrieve information associated with the account. Access to other accounts is limited to Administration accounts. Passwords can never be retrieved.
- Modify Account – An account can change the password for that account. Accounts with Administration level can change the password or attributes for any account.

Note that all of these methods/properties are subject to the access rights granted to the user account under which the action is taking place.

7.3 Audit

There are several kinds of auditing supported in the SM CLP Architecture. The MAP itself has a log which can be set to record certain types of information. They exact type of information recorded is implementation dependent.

The MAP also supports access to any logs available within the system. This includes retrieval of the number and identifiers for logs in the server; insertion, retrieval and removal of records (called events) in the log; and in some cases modification of the type of information recorded in the log.

7.4 CLP Service & MAP Management

The CLP Service itself is represented a manageable service in the SM CLP Architecture. Consequently, it is manageable as any other Managed Element would be.

The CLP Service can be disabled completely. The method for re-enabling the MAP is outside the scope of the specifications and is therefore implementation dependent.

Some systems may have dependencies between the MAP and the Managed System. If the MAP is dependent on the Managed System, then resetting the Managed System may result in resetting the MAP. If the system does not have a dependency between the Managed System and the MAP, then resetting the Managed System will not result in resetting the MAP. Any such dependency is implementation dependent.

Each transport and service can be enabled and disabled individually. Each service can be managed independently, allowing for customizable feature and property changes for each service.

The hardware that realizes the interface into the MAP is individually manageable. For example, in the case of an Ethernet interface, the MAC address, IP address(es) and parameters and TCP ports and parameters may all be configured as well as enabled and disabled.

Because the MAP is a container for all of the services and protocols, there are some architectural considerations to keep in mind. The first of these is that if the MAP is reset, all other services are reset as well. This implies that all sessions will be dropped when the MAP is reset.

Security information is persistent across MAP resets. This includes, but is not limited to, user accounts, account groups, properties, transport information and settings and service settings and log information and records.

The initial state of the MAP and initial user account is outside the scope of the SM CLP Architecture.

8 Discovery

Discovery in the SM CLP Architecture can be divided into three categories. The first is the discovery of the Managed Elements which are managed by the MAP. The second is the discovery of the capabilities of the MAP. The third is discovery of the MAP's services. This section will discuss all three aspects of discovery.

The first aspect of discovery is how a Client discovers which Managed Elements are managed by this MAP. Fortunately, this is a capability that exists in the protocols in use today. The CLP has the profiles, addressing and verbs to determine the Managed Elements within the management domain of the MAP. These are well documented in their individual specifications.

The second aspect of discovery is the capabilities of the MAP itself. This has been handled in the SM CLP Architecture by modeling the MAP within the profiles. The base MAP profile contains the classes for the standard services available within the MAP, such as the CLP and operation services. To discover the capabilities of the MAP is to simply discover the properties and methods available for the services within the MAP, as well as the service access points and transports for the MAP. The CLP and other specifications indicate how to query and alter the values of the properties for the services within the MAP.

The final aspect of discovery is discovery of the Service Access Points of the MAP. This is service dependent. It is expected that each service will define it's own discovery methodology. The DMTF has defined an SLP template for WBEM. An SLP template for the CLP has not been defined.

9 Conclusion

The SM CLP Architecture contains the models, mechanisms and semantics necessary to manage servers in the data center, regardless of service state. This includes the architectural, service and operations models, and covers boot and firmware update as well as service discovery. The profiles contain the required classes, instances, properties and methods necessary to manage systems. The combination of the profiles with the addressing methodology determines the format of the target addressing convention for compliant systems. This delivers the syntax and semantics necessary to manage servers.

The SM CLP Architecture is one component in a suite of specifications which delivers the Architecture, Addressing, Profiles, Command Line Protocol and Discovery necessary to manage the full range of current and emerging servers in enterprise environments.