



December 3-6, 2007, Santa Clara Marriott, Santa Clara, CA

SMASH & DASH Overview

Jon Hass, Dell Inc



Presentation Outline

- Introduction
- SMASH & DASH
- Architecture Overview
- Protocol Support
- Profiles
- Discovery
- Security
- Indications



Disclaimer

- The information in this presentation represents a snapshot of work in progress within the DMTF.
- This information is subject to change. The Standard Specifications remain the normative reference for all information.
- For additional information, see the Distributed Management Task Force (DMTF) Web site.



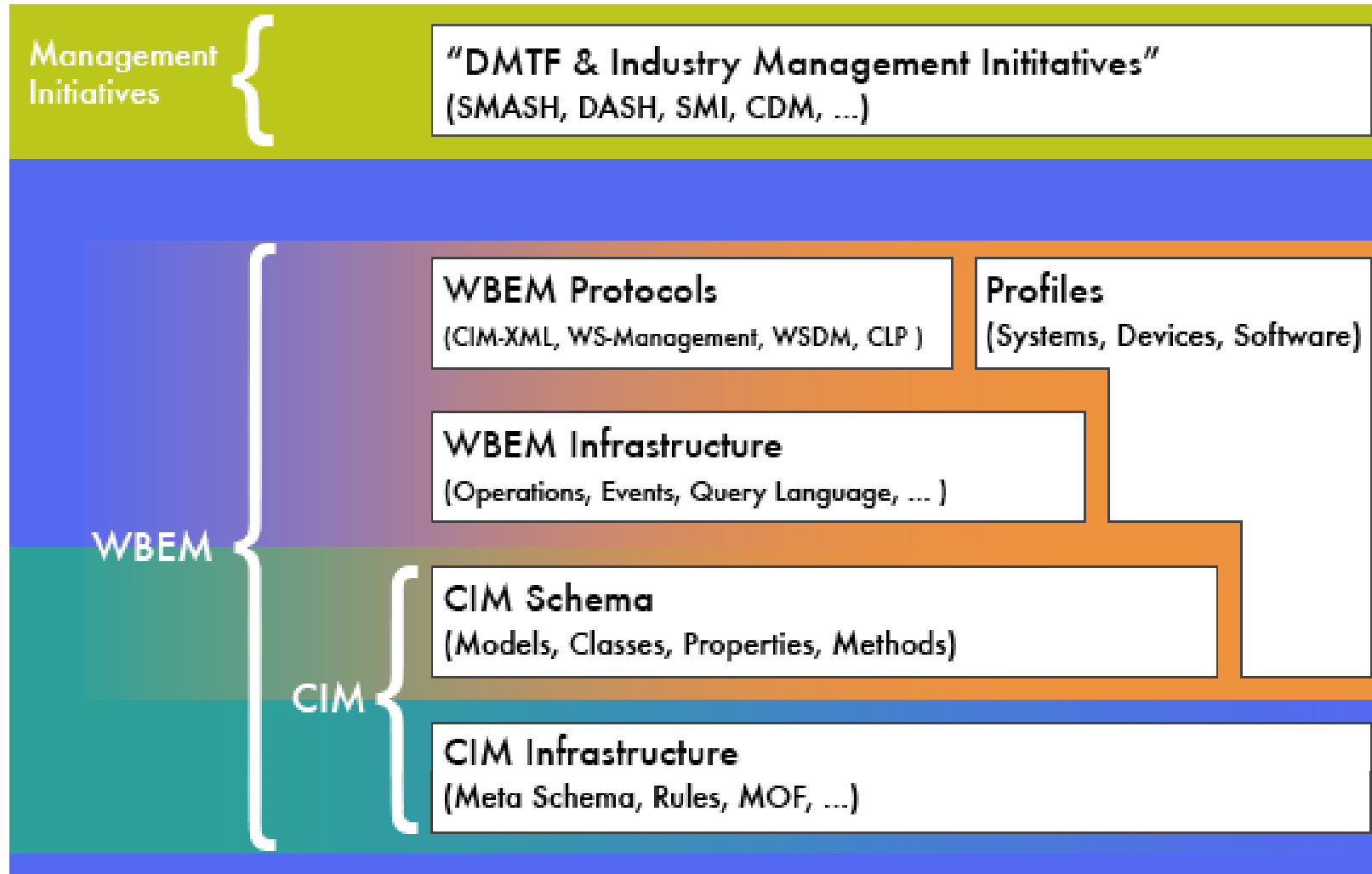
The DMTF was formed to lead the development, adoption and unification of management standards and initiatives for desktop, enterprise and internet environments



DMTF Management Initiatives

- DMTF currently has 3 Management Initiatives
 - SMASH – Systems Management Architecture for Server Hardware
 - DASH – Desktop and mobile Architecture for System Hardware
 - CDM – Common Diagnostics Model
- DMTF Recognizes SMI as a Management Initiative

DMTF Technology Diagram





Industry Standard Platform Manageability Alignment

- DMTF is driving a consistent interface and view, regardless of machine state or access method.

	In Service	Out Of Service
In Band	Host OS	Pre-OS Environment
Out of Band	iLO / BMC	iLO / BMC

WS-Management
 SMASH CLP
 SMASH/DASH Profiles
 Security

- Industry is aligning around key elements:
 - Protocols (Transport) – WS-Management & CLP
 - Profiles (Data Model) – SMASH, DASH & SMI-S Profiles



Architectural Models

- Management Models included in white papers
 - In-Band/In-Service, Out-of-band/Out-of-Service Model
 - Same as SMASH
 - Manageability Access Point Model
 - With different transport/protocol than SMASH has currently
 - Operational Model
 - Job oriented for certain functions (implementation dependent)
 - Session Capabilities
 - Concurrent session support (implementation dependent)
 - Resource Handling
 - Transient nature of resources
 - Security Model
 - Consistent across implementations



What is SMASH?

- SMASH Stands for Systems Management Architecture for Server Hardware
 - SMASH is a suite of specifications that deliver industry standard protocols and profiles to unify the management of the data center.
 - Vendor independent
 - Platform neutral
 - Independent of machine state
- The SMASH specifications utilize the **CIM data model** and industry standard transports and security mechanisms.
 - Align out-of-service with in-service manageability.
 - Align in-band with out-of-band manageability.
 - Customer Driven
- 1.0 Standard completed Dec, 2006
 - Made public at Manageability Developers Conference
- 2.0 Standard completed Sep 2007
 - Made public at Intel Developers Forum



What is DASH?

- DASH Stands for Desktop and mobile Architecture for System Hardware
 - Web services based programmatic interface for desktop to mobile environment, including bladed PCs.
 - Utilizes the **CIM Data Model**, leveraging the SMASH Profiles & Architecture gives this effort a head start.
 - Tackling tough issues like standardized Eventing.
 - First revision maps to ASF functionality plus inventory and account management.
- DASH consists of:
 - Architecture White Paper
 - **WS-Management**
 - DASH Implementation Requirements Specification
 - Profiles.
- 1.0 completed Apr, 2007
 - www.dmtf.org/standards/dash
 - Made public at Microsoft Management Summit (MMS), 2007
- 1.1 completed Dec 2007
 - Made public at MDC 2007.



State of the SMASH

- 1.0 Published Dec 2006
 - Architecture White Paper
 - SM CLP at 1.0 Final Standard
 - SM ME Addressing at 1.0 Preliminary Standard
 - Profiles & Mapping Specs at 1.0 Preliminary Standard
 - www.dmtf.org/standards/smash
- Interoperability Forum formed in the DMTF
 - SMASH 1.0 CLP: tester completed, tests 40% complete
 - DASH 1.0, SMASH 2.0: choosing platform to test through WS-Management
 - Infrastructure: developing certification repository
- 2.0 Published September 2007
 - Including WS-Management Support
 - Added Discovery
 - Additional Profiles
 - PCI, LED, KVM Redirection, Watchdog, OS Status, Indications
 - Added reference to SIM-S Host Hardware Raid Profile
 - Updated White Paper
- Planning on periodic “train” to add features/functions



State of DASH

- 1.0 Published April 2007
 - Architecture White Paper
 - Implementation Requirements Specification
 - Message Registry
 - www.dmtf.org/standards/smash
 - 1.0 Refresh published November 2007
- Interoperability Forum formed in the DMTF
 - DASH 1.0: choosing platform to test through WS-Management
 - Infrastructure: developing certification repository
- 1.1 Published December 2007
 - Additional Profiles
 - KVM Redirection, Media Redirection, USB Redirection, Text Console Redirection, OS Status, Battery, Opaque Data, BIOS Management, IP, Ethernet, Host Lan, DHCP Client, DNS Client, Software Update
 - Updated White Paper
- Planning on periodic “train” to add features/functions



Protocol Support

- Transport and Management Protocols are separate entities
 - Transport Stack (e.g. TCP/HTTP) is IP-based protocol layers below Management Protocol
 - Management Protocol is a WBEM Protocol (eg: WS-Management, WSDM, CLP and/or CIM/XML)
- Transport and Management Protocol Requirements
 - Leverage established Industry Standard Protocols to the extent practical:
 - Discovery
 - Transport Protocol Stack
 - Management Protocols
 - Non conflicting with well known or registered TCP/UDP port addresses
- Minimal effort required to operate through firewalls including NAT
 - (eg. avoid dynamic port assignment through negotiation)
- Support for transport protocol stack(s) meeting the following requirements
 - Consistent with (HTTP(S)/TCP) in-band/in-service instrumentation access
 - Suitable for embedded implementations



Management Protocol Support

- WS Management is the common programmatic interface leveraged by both SMASH and DASH standards
 - Normative references & mapping information in the implementation requirements specifications.
- SMASH also includes the SM CLP



What is the SM CLP?

- SM CLP (Server Management Command Line Protocol) is
 - Designed for a human (primary) or a script (secondary)
 - Working over, but not limited to, industry standard transports
 - Telnet & SSHv2
 - Exposes CIM data model in a “human friendly” fashion through simple commands
 - SM ME Addressing Spec turns CIM containment into command targets like “system1\fan1”
 - NOT a full featured programming interface
 - Because it is a lightweight communication mechanism with some semantics were intentionally left out.
 - Therefore, a programmatic interface is still required for some operations
 - But input and output are fully machine-parsable.
 - BUT all of the Hardware Operations (provisioning, allocation, configuration, inventory, state change, security) can be done with the CLP.
 - Either by a human, script or program
 - Because there is a grammar that defines input and XSD defined output.
 - Very light weight implementations can be done.



Profile Support

- A profile is a specified subset of the CIM Schema elements that describe a standard implementation for interoperability and conformance verification
 - The CIM Specification defines the language and methodology for describing management data.
 - Schemas provides the actual model descriptions.
- A profile contains
 - Required and Conditional CIM Element Properties and Methods
 - Class & Instance Diagrams
 - Profile Usage Guide and Profile Registration Profile compliance
- DMTF is producing Profiles
 - Strong desire to have common set of profiles to extent possible
 - Synergy with SMASH and SMI efforts
 - Definition of optional elements to support scaling from desktop and mobile platforms up to stand-alone, modular and partitionable servers



SMASH Profiles

High-level Profiles

1. **CLP Service**
2. **Base Server**
3. **Modular System**
4. **Chassis Manager**
5. Physical Asset
6. Boot Control
7. SM CLP Admin Domain
8. SMASH Collection
9. CPU
10. System Memory
11. Fan
12. LED
13. Power Supply
14. Power State Management
15. Record Log
16. Sensor
17. Watchdog
18. Host Hardware Raid (Reference)

19. OS Status
20. PCI Device
21. Software Update
22. Software Inventory
23. Host LAN Network Port
24. IP Interface
25. Ethernet Port
26. DHCP Client
27. DNS Client
28. SSH Service
29. Telnet Service
30. Role-Based Authorization
31. Simple Identity Management
32. Shared Device Management
33. Pass-Through Module
34. Device Tray
35. Text Console Redirection
36. KVM Redirection
37. Profile Registration
38. Computer System
39. Indications



DASH Profiles

High-level Profiles

1. **Base Desktop & Mobile**

Component Profiles

2. Physical Asset
3. Boot Control
4. CPU
5. System Memory
6. Fan
7. Power Supply
8. Power State Management
9. Sensor
10. Battery
11. BIOS Management
12. Opaque Data
13. OS Status
14. Software Update
15. Software Inventory
16. Host LAN Network Port
17. IP Interface
18. Ethernet Port
19. DHCP Client
20. DNS Client
21. Role-Based Authorization
22. Simple Identity Management
23. Text Console Redirection
24. KVM Redirection
25. Media Redirection
26. USB Redirection
27. Profile Registration
28. Computer System
29. Indications



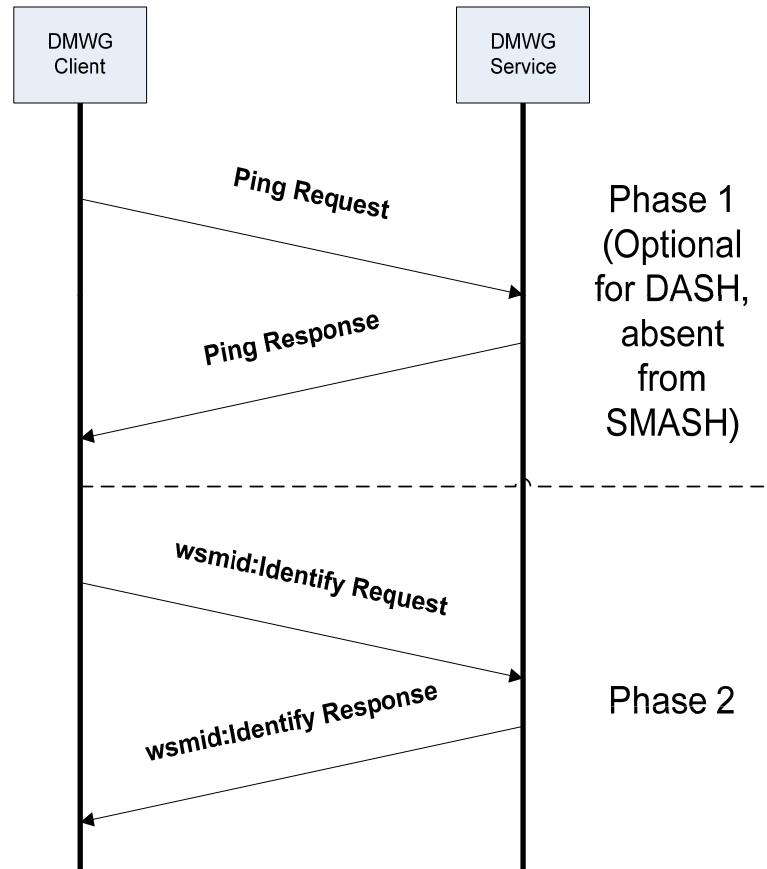
Web Services HW Management Ports

- Embedded DASH implementations operate on well known TCP Ports (repurposed ASF TCP Ports)
 - One for HTTP
 - Could be configured to support WS-Discovery only
 - One for HTTPS
 - We would support both TLS Security Profiles (see next page)
- Embedded SMASH implementations do not use the repurposed ASF ports.

Discovery Overview

- When discussing discovery it is important to divide the discussion into 3 broad groups, namely
 - Network Addressable End-Point Discovery
 - Classification (Type Discovery)
 - Service Discovery
- These broad groups can be further broken down with each layer of discovery providing more information including:
 - The existence of the Network Addressable End-Point.
 - The type of device (classification)
 - The services (capabilities) of the device as a whole
 - The device in the context of topology (e.g. a MAP in a Client Machine)

Two-Phase Discovery Message Flow





DMWG WS-Management Security Requirements

- HTTP 1.1 is the required transport
- Two classes of DASH defined WS-Management security levels: (See next slide for class details)
 - DMWG Class A – HTTP Only
 - DMWG Class B – HTTPS or IPSec
- A DASH implementation must be compliant with at least one of DASH Security Class levels
- A DASH implementation should be Class B compliant for privacy/confidentiality and additional security



SMASH & DASH Classes of WS-Management Security Profiles

- DASH Class A: HTTP Digest authentication (user authentication)
- DASH Class B: Support for at least one the security profiles below
 - HTTP_TLS_1
 - Two-level auth + encr: HTTP Digest auth. + TLS server/client certs (X.509) + TLS 1.0 (implementation of client cert is optional)
 - Cipher suites
 - TLS_RSA_WITH_AES_128_CBC_SHA
 - HTTP_TLS_2
 - Two-level auth + encr: HTTP basic auth. + TLS server/client certs (X.509) + TLS 1.0 (implementation of client cert is optional)
 - Cipher suites
 - TLS_RSA_WITH_AES_128_CBC_SHA
 - HTTP_IPSEC
 - Two-level auth + encr: HTTP 1.1 over IPsec with HTTP digest authentication
 - IPsec ESP transport mode – Authentication + Encryption
 - Cipher suites: One of the following
 - AES-GCM (key size: 128 bits, ICV or Digest len: 16 B)
 - AES-CBC (Key size: 128 bits) with HMAC-SHA1-96



SMASH & DASH Authentication Requirements

- Required User Account Management Profiles
 - Role Based Authorization Profile
 - Simple Identity Management Profile
- Three roles are defined for DASH & SMASH:
 - Administrator – Mandatory for SMASH & DASH
 - Operator – Optional for SMASH & DASH
 - Read Only – Mandatory for SMASH, Optional for DASH

Indications

- Two major categories of Indications for the CIM Model
 - Alert
 - Lifecycle
- Alert Indications
 - Message ID/string oriented class design
 - The underlying event and its data may or may not be modeled in the CIM class hierarchy
 - Includes handles pointing to the alerting Managed Element
 - Includes support for specifying RecommendedActions
- Lifecycle Indications
 - Generated based on changes in instantiated objects
 - Indication class includes the object instances and handles pointing to the objects
 - For changes in existing objects, the indication class also include the object instance before the change
 - Predominant approach used by SNIA – generally focused on the following:
 - Object creation and deletion
 - Value changes to the OperationalStatus and HealthState properties



Indications – Phase 1

Phase 1- Alert Indication Content Definition

- Platform Event Message Registry
 - Standardized message ID's and message strings
 - Publish recommended “Perceived Severity” mappings
- Publish Recommended Message Registry Mappings in progress
 - Recommended PET Frame Values Mapping
- Included in DASH 1.0 and SMASH 2.0 Specifications



Indications – Phase 2

Phase 2 – Update DMTF profiles with Indications definitions

- Profile update work targeted for CY 2008
- Will include specifying specific Platform Event and Operational Message Registry messages per profile
- Combining Message Registry messages with Lifecycle indications being considered



December 3-6, 2007, Santa Clara Marriott, Santa Clara, CA

Questions?