DM Scheduling Architecture

Candidate Version 1.0 – 26 Jun 2008

Open Mobile Alliance

OMA-AD-DM-Scheduling-V1_0-20080626-C

Contents

Figures

1. Scope (Informative)

This document describes the architecture of the DM Scheduling Framework. The architecture is based on the requirements and the use cases included in [DMSCHED-RD], specifying the functional components of the framework and the interactions among them. The dependency that DM Scheduling Enabler has upon other DM Enablers is also described.

2. References

2.1 Normative References

[RFC2119] Key words for use in RFCs to Indicate Requirement Levels, S. Bradner, March 1997, URL:http://www.ietf.org/rfc/rfc2119.txt

[DMSCHED-RD] OMA Device Management—DM Scheduling Requirements, Open Mobile Alliance™, OMA-RD-DM_Scheduling-V1_0, URL:http://www.openmobilealliance.org/

[DMSCHED] OMA Device Management Scheduling Management Objects, Version 1.0. Open Mobile Alliance™. OMA-TS-DM_Scheduling-V1_0, URL:http://www.openmobilealliance.org

[OMA-DM] OMA Device Management Protocol, Version 1.2. Open Mobile Alliance™. OMA-TS-DM-Protocol-V1_2, URL:http://www.openmobilealliance.org

[TRAP] OMA Device Management—Trap Management Objects, Version 1.0. Open Mobile Alliance™. OMA-TS-DiagMonTrapMO-V1_0, URL:http://www.openmobilealliance.org

2.2 Informative References

[ARCH-PRINC] OMA Architecture Principles V1.2, OMA-ArchitecturePrinciples-V1_2, Open Mobile Alliance™, URL:http://www.openmobilealliance.org/

[FUMO] Firmware Update Manegement Object, Version 1.0,

Open Mobile Alliance™, OMA-DM-FUMO-V1_0, URL:http://www.openmobilealliance.org/

[OMA-DICT] “Dictionary for OMA Specifications”, Version 2.6, Open Mobile Alliance™, OMA-ORG-Dictionary-V2_6, URL:http://www.openmobilealliance.org/

3. Terminology and Conventions

3.1 Conventions

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in [RFC2119].

All sections and appendixes, except “Scope” and “Introduction”, are normative, unless they are explicitly indicated to be informative.

3.2 Definitions

Device	See [OMA-DICT].
Device Description Framework	A markup language used to describe OMA DM object schema; may be used in a standardized specification to describe the characteristics of conformant implementations or published by a vendor to describe a particular device implementation.
Device Management Authority	Any legal entity authorized, either directly or through delegation, to perform management operations on a terminal using the OMA Device Management protocol through a set of management objects.
DM Client	An abstract software component in a Device implementation that conforms to the OMA Device Management Enabler static conformance requirements specified for DM Clients. It servers as an end-point of the DM Client-Server Protocols including the one described in this architecture document.
DM Server	An abstract software component in a deployed Device Management infrastructure that conforms to the OMA Device Management Enabler static conformance requirements specified for DM Servers. It servers as an end-point of the DM Client-Server Protocols including the one described in this architecture document.
Management Object	A schema for configuration settings that an OMA DM client exposes to OMA DM servers for management operations defined in the OMA DM Enabler [OMA-DM].
Schedule	A plan for performing management operation that can be embodied as a piece of configuration to be consumed by the DM Scheduling Agent.
Status Reporting	It is the sending of Generic Alert to the DM server to inform the interested events occurred in the Device with regards to the Schedules under the control of the server.
Trap	A mechanism employed by a management authority to enable the Device to capture and report events and other relevant information generated from various components of the Device, such as a protocol stack, device drivers, or applications.

3.3 Abbreviations

DDF	Device Description Framework
DM	Device Management
DMA	Device Management Authority
DMEC	OMA DM Enabler Client
DMES	OMA DM Enabler Server
DMS	Device Management Server
DMSEC	DM Scheduling Enabler Client
DMSES	DM Scheduling Enabler Server
MO	Management Object
OMA	Open Mobile Alliance

4. Introduction (Informative)

The DM Scheduling Enabler, as a Management Object specification, builds on top of the OMA DM v1.2 to seamlessly add the common scheduling capability to existing OMA DM based management infrastructure.

With the added capability the management operations, which the underlying OMA DM system is able to perform over the DM session, can be scheduled on the Device and executed offline when schedule matches: time-based or event-based.

The design of the architecture does not require other OMA DM Enablers to be dependent on the DM Scheduling Enabler, allowing such enablers to evolve independently from the DM Scheduling Enabler.

4.1 Planned Phases

The architecture of the Scheduling Framework described in this document is to cover all the requirements and use cases for the DM Scheduling Enabler Release 1.0 included in [DMSCHED-RD]. At the moment of writing, there is no plan to develop future releases of DM Scheduling Enabler.

5. Architectural Model

5.1 Dependencies

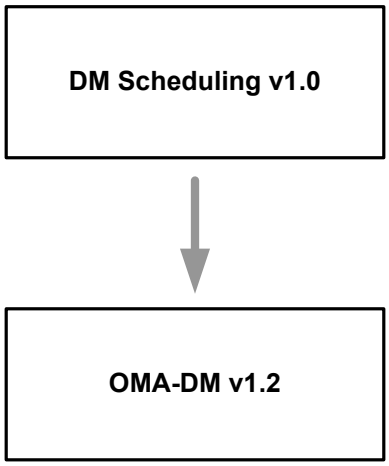

Figure 1: Dependency between DM Scheduling v1.0 Enabler and others.

OMA Device Management V1.2 [OMA-DM]

Since the DM Scheduling Enabler V1.0 is the Management Object Enabler, it depends upon the OMA Device Management Enabler V1.2 the same way as other Management Object Enablers do. In addition, the DM Scheduling Enabler V1.0 leverages OMA Device Management Enabler also for the execution of scheduled management tasks.

5.2 Architectural Diagram

5.2.1 Conceptual Diagram

The Scheduling Framework builds on top of the OMA DM Enabler [OMA-DM] the same way as other Management Object Enabler does, to enable the offline processing of the management commands according to the Schedules given by the DM Server. The commands may be processed repeatedly if the Schedule is of recurring type, or may never be processed if schedule match doesn't occur. When the commands are executed, the result is reported back to the DM Server by default if not specified otherwise.

Once the Schedule is installed on the Device and activated through the DM Sessions, the DM Scheduling Agent runs the Schedule starting from setting up the triggers and then waiting for them to fire the execution of the scheduled tasks. For time-based Schedules the agent sets an alarm clock or timer, and similarly for event-based Schedules the agent registers at the designated Trap for receiving its triggers as soon as the related events occur.

When executing the scheduled tasks after being triggered, the DM Scheduling Agent simply transfers rather than process the OMA DM message to the underlying DM Client asking it to process the message. Subsequently, the DM Client processes it and returns the responses to the agent, which would deliver them up to the DM Server or silently discard them.

In this context, the DM Scheduling Agent can be seen as an entity that resides on the Device interacting with other functional entities on the same Device to perform the scheduled tasks on behalf of the DM server.

5.2.2 Functional Architecture

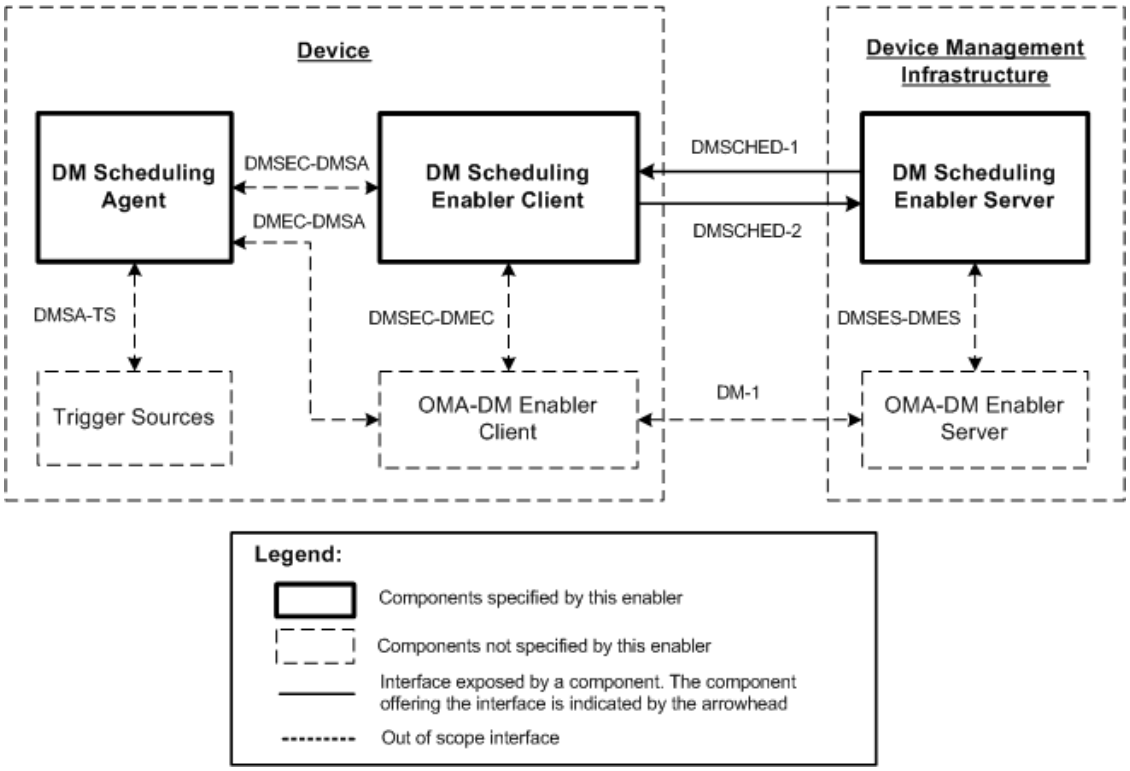

Figure 2: Device Management Scheduling Framework Architecture

5.3 Functional Components and Interfaces

5.3.1 DM Scheduling Enabler Server

The DM Scheduling Enabler Server is an abstract software component in the device management infrastructure that implements and conforms to the DM Scheduling Enabler [DMSCHED]. The DM Scheduling Enabler Server creates Schedules, transfers them to the Device, and activates them. It can also reconfigure, delete, or deactivate the Schedules. In addition, it also interprets and processes the asynchronous messages called the Status Reporting sent from the Device reporting the status of the Schedules on the Device, e.g. the run-time errors encountered. The DM Scheduling Enabler Server heavily relies on the underlying OMA-DM Enabler in performing those tasks mentioned above.

5.3.2 DM Scheduling Enabler Client

The DM Scheduling Enabler Client is the abstract software component in the Device implementation that serves as an end-point of the DMSCHED-1/2 interfaces conforming to the DM Scheduling Enabler specifications. It is a logical entity with its functionalities clearly distinguished from other entities shown in figure 2. However, in actual implementation, the DM Scheduling Enabler Client may or may not be physically bundled into the OMA-DM Enabler Client and/or the DM Scheduling Agent.

The DM Scheduling Enabler Client is responsible for communicating the DM commands with the DM Scheduling Enabler Server for the installation of new Schedules in the Device, reconfiguration of the existing Schedules, and removal of the Schedules out of the Device. Usually, such operations begin from the server, but it may also start from the User requests. Note that normally only one Management Authority is given the ownership of the Schedule.

The DM Scheduling Enabler Client is also the initiator of the Status Reporting messages, which are sent proactively to the Management Authority. See section 5.3.1.6 for more description of the Status Reporting.

The DM Scheduling Enabler Client heavily relies on the services provided by the underlying OMA-DM Enabler in doing such tasks mentioned above as well as in providing User Interactions functions during the installation, reconfiguration, and removal of the Schedules.

5.3.3 DM Scheduling Agent

The DM Scheduling Agent is a logical entity that actually runs the Schedules conforming to the DM Scheduling Enabler specifications [DMSCHED]. More specifically, it continuously performs the Scheduling Operations for each active Schedule on the Device. The Scheduling Operation consists of the following stages: Initialization, Trigger Waiting, User Interaction, Task Execution, Gating, and Status Reporting. It is, after successful initialization, a basically straight forward operation – execution of the scheduled operations triggered by the external sources such as timer (alarm clock) or Trap.

See the following subsections for more descriptions for each of the stages.

5.3.3.1 Scheduling Operation Initialization

Whenever a Schedule gets activated, the DM Scheduling Agent starts a Scheduling Operation in this stage, where the Agent initializes the Scheduling Operation to make sure that the following stages will run free of error, for instance checking the integrity of the Schedule, detecting wrong configuration, and configuring the trigger sources - timer or Traps - to send triggers according to the related Schedule. Or the integrity of the Schedule may be checked against possible modification during the course of the delivery and the storage. The mechanisms used in this stage are implementation specific.

5.3.3.2 Trigger Waiting

In this stage, after successful initialization of Scheduling Operation, the Agent waits for the triggers. When it receives the triggers, the Agent verifies the security information attached to the triggers to make sure they originated from the valid sources before moving on to the rest of stages.

There are two possible types of triggers:

- **Timer:** the given points in time, duration, periods, or intervals.
- **Trap Events:** the indication as well as the related data received from the source of the Trap, which includes but not limited to the protocol stacks developed by other standard body, other OMA Enablers, and device driver software. The mechanisms for receiving the Traps triggers must conform to the Trap framework [TRAP].

5.3.3.3 User Interaction

This is OPTIONAL stage between the Trigger Waiting and Task Execution stage, which will be skipped if there is no user interaction rules specified in the Schedule. In this stage, the client-driven user interaction functions are provided. These functions are similar to the User Interaction specified in [OMA-DM] but distinguished from it in that these interactions occurs only between the User and the DM Scheduling Agent. That is, DM Server is not directly involved, and DM sessions are not required in the course of interactions.

A few types of user interaction functions should be supported to meet the usability and privacy requirements in [DMSCHED-RD]. For example, when there is a trigger a short text may be displayed to notify the user about the task to run, and/or the user may be asked to confirm or cancel the task.

5.3.3.4 Task Execution

In this stage, the DM Scheduling Agent executes the scheduled tasks. If the OMA DM message is specified in the Schedule, the Agent simply forwards the message to the underlying OMA-DM Enabler Client, and then the OMA-DM Enabler Client will actually process the message as though it is received through the DM session.

5.3.3.5 Gating

This OPTIONAL Gating function would be useful especially in wireless environment where millions of end-user devices should be managed. As a result of OMA DM message processing in the previous stage the responses are generated, which are, by default, delivered to the server via the Status Reporting mechanism. This is where the Gating comes into play. If Gating is used, such responses that are gated-off will not be delivered to server but silently discarded in the Device.

5.3.3.6 Status Reporting Function

Once the Schedules are installed in the Device and running, status changes and events occurred to them are reported to the server. The reports are contained in the properly formatted message, and delivered to the server using the Generic Alert mechanism defined in [OMA-DM]. For example, the results obtained from processing of the scheduled tasks may be reported back to the server, or the errors occurred throughout the Scheduling Operation could be reported. In addition, the status changes resulting from the User's request to modify, cancel, or terminate the Schedule may also be reported.

5.3.4 Trigger Sources

These are abstract software components in the Device that provide the triggers to the DM Scheduling Agent. Every Trigger Sources may fall into two types: Timer and Trap Source. The Trap Source is the component that is compliant to the OMA Trap Enabler [TRAP]. The Trigger Source is out of scope of the DM Scheduling Enabler.

5.3.5 OMA-DM Enabler Server

The OMA-DM Enabler Server shown in figure 2 is an abstract software component in the deployed network infrastructure that acts as the network side end points of the DM-1 interface complying with the [OMA-DM]. The OMA-DM Enabler Server provides services based on [OMA-DM] to the DM Scheduling Enabler Server when it performs its tasks mentioned in section 5.3.1.

5.3.6 OMA-DM Enabler Client

The OMA-DM Enabler Client shown in figure 2 is the abstract software component in a Device that conforms to [OMA–DM]. In order to execute the scheduled tasks the OMA-DM Enabler Client will be leveraged as described in section 5.3.1.4. Through DMSEC-DMEC interface depicted in figure 2, the OMA-DM Enabler Client receives OMA DM messages. Then it processes them, and returns the responses to the DM Scheduling Enabler Client. In addition, the OMA-DM Enabler Client will also be leveraged in communicating the DM commands for installing, modifying, and removing the Schedules, and in sending Status Reporting messages.

5.3.7 Interfaces

5.3.7.1 DMSCHED-1

The DMSCHED-1 is the logical interface between the DM Scheduling Enabler Client and the DM Scheduling Enabler Server to install, reconfigure, and remove the Schedule in the device. And this interface specifies on the activation and deactivation of the Scheduling Operation. The interface is also used by the DM Scheduling Enabler Client to send the Status Reporting messages to the DM Scheduling Enabler Server. The interface is logical in a sense that it is layered on top of the physical DM-1 interface defined in [OMA-DM], as shown in figure 2.

5.3.7.2 DMSCHED-2

The DMSCHED-2 is second logical interface between the DM Scheduling Enabler Client and the DM Scheduling Enabler Server. Through the DMSCHED-2 interface, the DM Scheduling Enabler Client provides information on the Scheduling Operation on the Device such as the registered status changes and the results from scheduled task executions, and the DM Scheduling Enabler Server receives such information or reports. Note that this interface is associated with the Status Reporting described in section 5.3.1.6.

5.3.7.3 DMSEC-DMSA

This interface allows the logical separation between DM Scheduling Enabler Client and DM Scheduling Agent in such a way that DM Scheduling Enabler Client only takes care of the protocols in communication with the server, letting DM Scheduling Agent focus on running Schedules. In other words, this interface denotes a logical link so that the two functionally separate entities can collaborate with each other. This interface is also implementation specific. Depending on the implementation, this interface may explicitly implemented, or exist implicitly or event not exist.

At high level in a DM layered architecture model, DM Scheduling Enabler Client acts as an interface (or proxy) specifically to the DM Scheduling Agent, through which the Management Authority can access and use the services provided by DM Scheduling Agent. In this context, this interface should be able to map the requests that DM Scheduling Enabler Client receives from the server, e.g. installing the Schedule, activating the Schedules, etc. Also through this interface, DMSA should be able to tell the DM Scheduling Enabler Client what happened while it is running the Schedule, e.g. run-time errors, so that it can be communicated back to the server.

5.3.7.4 DMSEC-DMEC

This purely implementation specific interface allows the logical separation between the two functional components, DM Scheduling Enabler Client and the DM Enabler Client, only from the architectural perspectives. This interface should adopt very similar mechanism that is supposed to be used between other MO Enabler Clients and the underlying DM Enabler Client.

5.3.7.5 DMEC-DMSA

Through this interface, the DM Scheduling Agent is able to use the OMA DM message processing capability of the underlying OMA DM Client Enabler.

5.3.7.6 DMSES-DMES

The DMSES-DMES is the interface between the DM Scheduling Enabler Server and the OMA-DM Enabler Server. The DM Scheduling Enabler is layered on top of the functions provided by the OMA-DM Enabler and, through this interface, the

functionality provided by the OMA-DM Enabler Server is used by the DM Scheduling Enabler Server. This interface is implementation specific.

5.3.7.7 DMSA-TS

The DMSA-TS is an implementation specific interface through which the DM Scheduling Agent configures the Trigger Sources and receive the triggers or Notifications.

5.4 Flows (Informative)

5.4.1 Schedule Installation

When it receives a request from the External Management Infrastructure to setup and run a certain Schedule on the Device (step 1), the DM Scheduling Enabler Server creates and sends a Schedule to the Device (step 2). Although not explicitly shown in the diagram, this request is transported through the underlying OMA-DM Enabler. Receiving the request, the Device authenticates the server and possibly verifies the request to the extent practical (step 3). Note that, in between step 2 and step 3, the on-session user interaction may optionally be performed. After successful authentication and verification, the Schedule is installed in the DM Tree of the Device (step 4), and the DM Scheduling Enabler Server then receives a confirmation that the Schedule it sent is successfully installed (step 5). Similar to step 2, this confirmation is transported through the underlying OMA-DM Enabler. Finally, through the proper means of communication, the confirmation is relayed back to the External Management Infrastructure (step 6).

In figure 3, the Scheduling Context Installation flow is shown.

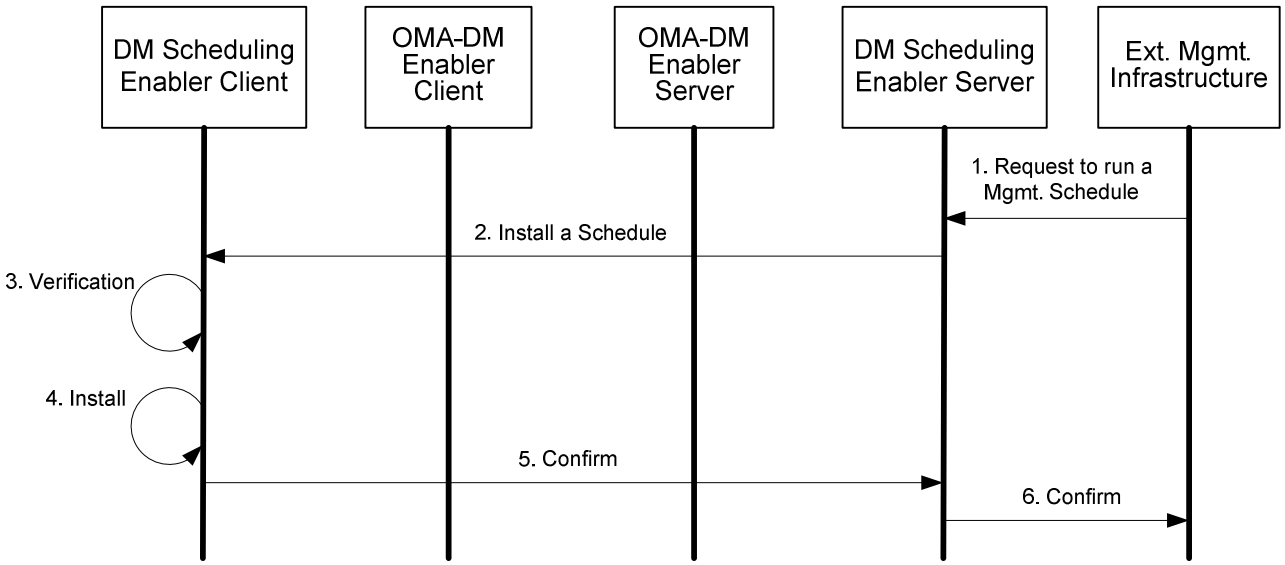

Figure 3: Schedule Installation Flow

[Note: The DM Scheduling Enabler Client and the OMA-DM Enabler Client may be bundled together and interface between them may be different according to the various implementations. The same is true between the DM Scheduling Enabler Server and the OMA-DM Enabler Server.]

5.4.2 Scheduling Operation

This subsection presents the typical Scheduling Operation flow as an example. The server activates a Schedule by sending commands over DM-1 interface (step 1), the request is eventually served by the DM Scheduling Agent by starting the Scheduling Operation for the designated Schedule (step 2). As already explained in the preceding sections, if the Schedule looks fine the agent configures the External Triggers so that they start sending triggers at the scheduled times or whenever the events are detected (step 3). Now, the agent waits for the trigger to come. At last a trigger is received (step 4), and it fires the rest of the Scheduling Operations. The scheduled tasks are executed (step 5) and the responses generated as a result of the execution are forwarded to the DM Scheduling Enabler Client (step 6), which subsequently communicates them up to the server via Status Reporting (step 7). If the Schedule is recurring, the flow is repeated from receiving other triggers until the Schedule is expired, or halted by the server or User.

Note that although the OMA DM Enabler Client and Server are not shown in the diagram, they actually exist between the DM Scheduling Enabler Client and the Server. They are not shown in the diagram for the sake of simplicity.

Figure 4 below depicts this flow.

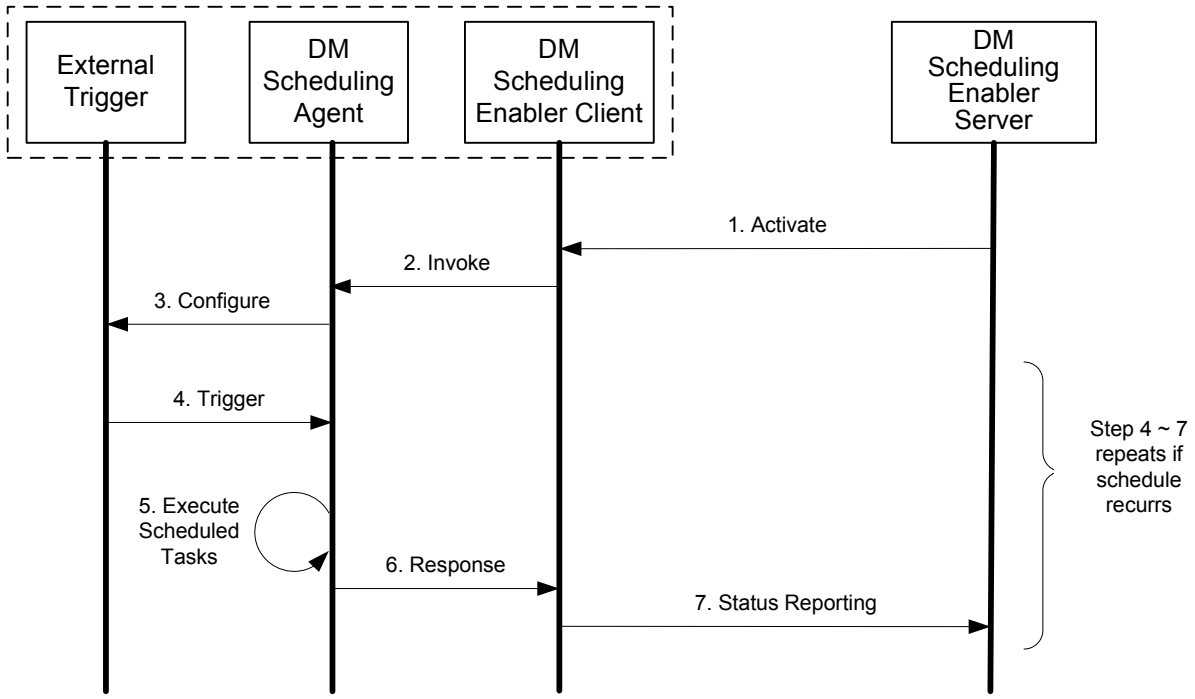

Figure 4: Scheduling Operation Flow (example)

5.5 Security Considerations

Based on the DM layered architecture model, the DM Scheduling Enabler, like other MO Enablers, fully takes advantage of the security mechanisms provided by the underlying OMA DM Enabler. The mechanisms for authentication, integrity protection and data confidentiality of the DM v1.2 can be leveraged to protect the Schedules in transit.

As important as the protection of the Schedule in transit is ensuring the data integrity after that because, depending on the device platform, the Schedules in storage could be left vulnerable to the attacker or accidental modifications by the user, for example. Such content protection can be achieved by using one of the out-of-band content protection mechanism, such as [XMLSIGN] or [XMLENC], as is recommended in [DMSEC].

Finally, additional care should be taken to the DMSA-TS interface, the interface between the DM Scheduling Agent and the Trigger Sources. The DM Scheduling Agent receives triggers from the external components as a signal that it is time to execute the scheduled management tasks. Considering that such triggers may be implemented by different vendors from those of the DM Scheduling Agent, there has to be some mechanism for the DM Scheduling Agent to ensure that received trigger is not modified en route and originated from the trusted source. A platform may provide sufficient mechanism to secure this interface or this may not be an issue at all depending on the devices. It is also possible that simple password-based access control mechanisms be devised during the Technical Specifications development stage.

Appendix A. Change History (Informative)

A.1 Approved Version History

Reference	Date	Description
n/a	n/a	No prior version

A.2 Draft/Candidate Version 1.0 History

Document Identifier	Date	Sections	Description
Draft Versions: OMA-AD-DM_Scheduling-V1_0	07 Oct 2005	All	Initial Draft Version
	07 Feb 2006		ARC comments addressed.
	09 Mar 2006		Editorial Updates.
	19 Apr 2006		Updates on references, flows. Marked some references as informative to allow different implementations.
	29 May 2006		Layered Architecture is captured and explained.
	13 June 2007	All sections	Restructured to reflect the changes in DM Scheduling RD.
	10 Sep 2007	5.3.6, 5.5	New architecture diagram in place, and security section added.
	06 Nov 2007	5.3.7.2	Document prepared for ARC re-review with change bars. Section 5.3.7.2, which was mistakenly left out, is brought back in.
	21 Jan 2008	2.2, 3.2	Replace the definition for the Device with that from the OMA Dictionary. And OMA Dictionary was added to the references.
Candidate Version: OMA-AD-DM_Scheduling-B1-0	26 Jun 2006	n/a	Status changed to Candidate by TP TP ref#: OMA-TP-2008-0239-INP_DM_Scheduling_V1_0_ERP_for_Candidate_Approval