ManageableDevice:2 Device Template Version 1.01

For UPnP Version 1.0
Status: *Standardized DCP (SDCP)*
Date: *February 16th, 2012*

THE UPNP FORUM TAKES NO POSITION AS TO WHETHER ANY INTELLECTUAL PROPERTY RIGHTS EXIST IN THE STANDARDIZED DCPS. THE STANDARDIZED DCPS ARE PROVIDED "AS IS" AND "WITH ALL FAULTS". THE UPNP FORUM MAKES NO WARRANTIES, EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE STANDARDIZED DCPS, INCLUDING BUT NOT LIMITED TO ALL IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT AND FITNESS FOR A PARTICULAR PURPOSE, OF REASONABLE CARE OR WORKMANLIKE EFFORT, OR RESULTS OR OF LACK OF NEGLIGENCE.

Authors	Company
Enrico Grosso	Telecom Italia
Jooyeol Lee	Samsung Electronics
William Lupton	2Wire / Pace
Davide Moreo	Telecom Italia
Xavier Roubaud	France Telecom
Kiran Vedula	Samsung Electronics

* Note: The UPnP Forum in no way guarantees the accuracy or completeness of this author list and in no way implies any rights for or support from those members listed. This list is not the specifications' contributor list that is kept on the UPnP Forum's website.

Contents

List of Tables

List of Figures

1. Overview and Scope

1.1. Introduction

This device specification is compliant with the UPnP Device Architecture version 1.0. It defines a device type referred to herein as *ManageableDevice*, and a set of UPnP services that provide UPnP Device Management (DM) functions.

UPnP DM services can be used to add management operations to any UPnP device. Management includes such functions as troubleshooting, diagnostic, configuration and software/firmware image management. Section 2.3 defines the general architecture for deployment of these services, which provide functionality to:

- Perform basic management operations on a device using the *BasicManagement* service (refer to the *BasicManagement* service specification [BMS] for details).
- Configure a device using the *ConfigurationManagement* service (refer to the *ConfigurationManagement* service specification [CMS] for details).
- Manage software components on a device using the *SoftwareManagement* service (refer to the *SoftwareManagement* service specification [SMS] for details).

In addition, a set of generic configuration/status parameters, referred to as the UPnP DM *Common Objects*, is defined in *ConfigurationManagement* service [CMS], and are accessible via the actions defined in the *ConfigurationManagement* service. A *ManageableDevice* has to support these parameters (some are OPTIONAL) via the *ConfigurationManagement* service.

A full-featured UPnP DM device provides Control Points with the following capabilities:

- *BasicManagement*:
    - Reboot and/or reset the device;
    - Perform IP (Internet Protocol) layer, self-test diagnostics and bandwidth tests;
    - Retrieve status and content of device logs.
- *ConfigurationManagement*:
    - Check the configuration and status of a device;
    - Provision or configure devices.
- *SoftwareManagement*:
    - Manage software lifecycle on a device;
    - Update software components and firmware images.

The UPnP DM service specifications do not specify or restrict format and content of log files nor protocols for downloading software or firmware.

Regarding UPnP DM services deployment, there are two main methods:

- either by deploying a UPnP DM device with at least *BasicManagement* service and the *ConfigurationManagement* service (*SoftwareManagement* service is OPTIONAL),

- or by deploying independently UPnP DM services ([BMS], [CMS] and [SMS]) as additional features of other types of UPnP device.

At last, Device Management operations can be protected by an OPTIONAL *Security* feature based on *DeviceProtection:1* [DPS]. Actions that do not return sensitive information, change the device configuration, or affect normal device operation can always be invoked by all Control Points. If the *Security* feature is supported, other actions can only be invoked if the Control Point is appropriately authorized.

1.2. References

[BMS] *UPnP BasicManagement:2 Service Document*, UPnP Forum, February 16, 2012.
Available at: http://www.upnp.org/specs/dm/UPnP-dm-BasicManagement-v2-Service.pdf

[CMS] *UPnP ConfigurationManagement:2 Service Document*, UPnP Forum, February 16, 2012.
Available at: http://www.upnp.org/specs/dm/UPnP-dm-ConfigurationManagement-v2-Service.pdf

[DPS] *UPnP DeviceProtection:1 Service Document*, UPnP Forum, February 24, 2011.
Available at : http://www.upnp.org/specs/gw/UPnP-gw-DeviceProtection-v1-Service.pdf

[RFC 2119] *RFC 2119, Key words for use in RFCs to Indicate Requirement Levels*, IETF, March 1997.
Available at: http://tools.ietf.org/html/rfc2119

[SMS] *UPnP SoftwareManagement:2 Service Document*, UPnP Forum, February 16, 2012.
Available at: http://www.upnp.org/specs/dm/UPnP-dm-SoftwareManagement-v2-Service.pdf

[SOAP] *Simple Object Access Protocol (SOAP) 1.1*
Available at: http://www.w3.org/TR/2000/NOTE-SOAP-20000508

[UDA] *UPnP Device Architecture, version 1.0*, UPnP Forum, July 20, 2006.
Available at: http://www.upnp.org/specs/arch/UPnP-arch-DeviceArchitecture-v1.0.pdf

1.3. Glossary

ACL	Access Control List
BMS	*BasicManagement* Service
CMS	*ConfigurationManagement* Service
CP	Control Point
DDD	Device Description Document
DM	Device Management
DPS	*DeviceProtection* Service
EE	Execution Environment

MD	*ManageableDevice*
SDO	Standards Development Organization
SMS	*SoftwareManagement* Service
UDA	UPnP Device Architecture
VM	Virtual Machine

1.4. Notation

- In this document, features are described as Required, Recommended, or OPTIONAL as follows:

  The key words “MUST,” “MUST NOT,” “REQUIRED,” “SHALL,” “SHALL NOT,” “SHOULD,” “SHOULD NOT,” “RECOMMENDED,” “MAY,” and “OPTIONAL” in this specification are to be interpreted as described in [RFC 2119].

  These keywords are thus capitalized when used to unambiguously specify requirements over protocol and application features and behaviour that affect the interoperability and security of implementations. When these words are not capitalized, they are meant in their natural-language sense.
- Strings that are to be taken literally are enclosed in “double quotes”.
- Words that are emphasized are printed in *italic*.
- *Data Model* names and values, and literal XML, are printed using the `data` character style.
- Keywords that are defined by the UPnP DM Working Committee are printed using the *forum* character style.
- Keywords that are defined by the UPnP Device Architecture are printed using the **arch** character style.
- A double colon delimiter, “::”, signifies a hierarchical parent-child (parent::child) relationship between the two objects separated by the double colon. This delimiter is used in multiple contexts, for example: Service::Action(), Action()::Argument.

2. Device Definitions

The requirements in this section apply only when *ManageableDevice* is used. Section 2.3 defines the general architecture for deployment of the UPnP DM services.

2.1. Device Type

The following device type identifies a device that is compliant with this specification:

urn:**schemas-upnp-org:device**: *ManageableDevice:2*

2.2. Device Model

A *ManageableDevice:2* MUST implement minimum version numbers of all REQUIRED embedded devices and services specified in the table below. A *ManageableDevice:2* device can be either a *Root* device or can be *Embedded* within another UPnP device. A *ManageableDevice:2* device can contain other standard or non-standard *Embedded* UPnP devices as well.

Table 2-1: Device Requirements

DeviceType	Root	R/O[1]	ServiceType	R/O[1]	Service ID[2]
ManageableDevice:2	*Root* or *Embedded*	*R*	*BasicManagement:2*	*R*	*BasicManagement*
			ConfigurationManagement:2[3]	*R*	*ConfigurationManagement*
			SoftwareManagement:2	*O*	*SoftwareManagement*
			Standard non-DM services defined by UPnP (QoS, Security etc.) go here.	*X*	*To be defined*
			Non-standard services embedded by an UPnP vendor go here.	*X*	*To be defined*
			DeviceProtection:1	*CR*[4]	*DeviceProtection*
Standard devices embedded by a UPnP vendor go here.	*Embedded*	*O*	*Services as defined by the corresponding standard UPnP Device Definition go here.*		
Non-standard devices embedded by a UPnP vendor go here.	*To be defined*	*X*	*To be defined*	*To be defined*	*To be defined*

[1] *R* = Required, *CR*= Conditionally required, *O* = OPTIONAL, *X* = Non-standard.
[2] Prefixed by urn:**upnp-org**:**serviceId**: .
[3] MUST support the UPnP DM *Common Objects* as defined in *ConfigurationManagement* service.
[4] REQUIRED if the *Security* feature is supported.

2.3. Architecture

This section provides the general architectural concepts for the deployment of UPnP DM services, within *ManageableDevice* or within any other UPnP device.

Figure 2-1 shows the dependency diagram for *ManageableDevice* and for several other example UPnP devices. It can be seen that this diagram implies most of the *ManageableDevice* requirements of Table 2-1:

- *BasicManagement* [BMS] is required.
- *ConfigurationManagement* [CMS] is required (in order to access the required UPnP DM *Common Objects*).
- *SoftwareManagement* [SMS] is OPTIONAL.
- *DeviceProtection* [DPS] is OPTIONAL.

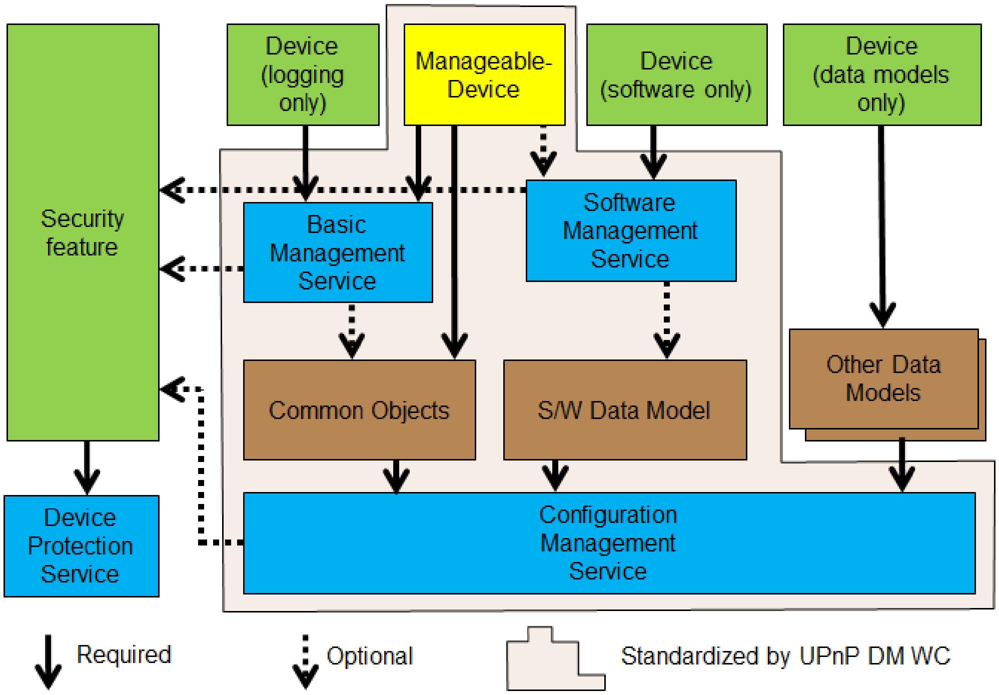

Figure 2-1: Architecture Diagram

Figure 2-1illustrates the architecture of a *ManageableDevice* and also of three other UPnP devices:

- The first one implements only logging actions and so requires only *BasicManagement*. *BasicManagement* has an OPTIONAL dependence on the UPnP DM *Common Objects* (see [BMS] for details).
- The second one implements only software life-cycle actions and so requires only *SoftwareManagement*. *SoftwareManagement* has an OPTIONAL dependence on the *Software Data Model*.

- The third one requires access to some additional data models, and so requires only *ConfigurationManagement*.

The UPnP DM *Common Objects* define baseline configuration information for the device, which is accessible via the *ConfigurationManagement* service. When the *ConfigurationManagement* service is included in a UPnP device other than *ManageableDevice*, the UPnP DM *Common Objects* will not necessarily be supported; instead other specific data models might be provided by the UPnP device. If this is a standard UPnP device type, additional data models might also be defined by the corresponding UPnP Working Committee.

Since many other definitions of sets of parameters for device management (data models) are widely available in the industry, a mechanism for importing other data model definitions into the UPnP DM *Data Model* is also defined. Details about this mechanism can be found in [CMS]. This mechanism also specifies how data models from other **UPnP working committees should be handled by the *ConfigurationManagement* service.**

In addition to UPnP DM services, security feature may also be OPTIONALly integrated to provide control over which actions can be invoked and which resources can be manipulated. Each device, including *ManageableDevice*, can choose to support or not this feature. In any case, the security feature MUST support *DeviceProtection* service [DPS] (see chapter 2.4). To conclude, security feature is applicable independently to any service, for example only to *ConfigurationManagement* service in a full feature *ManageableDevice*.

2.4. Security considerations

2.4.1. Scope

The main security concern of Device Management is to protect against attack from malicious Control Points. Moreover, without *Security*, Control Points can invoke any actions that are exposed by UPnP devices.

More specifically the *Security* feature addresses the following needs of different stakeholders:

- For the End Users
    - To prevent unauthorized Control Points from invoking device management actions;
    - To avoid Man In the Middle attacks by providing secure communication channels;
    - To be able to provide fine-grained management control by providing *Roles* and Access Control Lists;
    - To feel comfortable delegating home network management to a third party.
- For the Device Manufacturers
    - To prevent unauthorized Control Points from managing UPnP devices;
    - To allow manufacturers to have fine-grained specific management operations.
- For the Service Providers
    - To define rules in order to ensure that some specific actions and resources will only be managed by an authorized Control Point;
    - To allow multiple service providers each to manage their own area of interest, thereby minimizing the possible side effects.

2.4.2. Overview

Security feature is OPTIONAL. It provides authorization mechanism to permit invocation of actions and to allow access to resources. Indeed *BasicManagement:2*, *ConfigurationManagement:2* and *SoftwareManagement:2* actions can change the behaviour of the UPnP device:

- Reboot operation when the device is doing some important transaction may impact user experience;
- Firmware update when the device is performing some configuration changes or some other transactions may affect the device operation.
- Some Execution Units may not be started and stopped by any Control Points;
- Some subtrees of the device *Data Model* may not be modified by any Control Points.

To enable the OPTIONAL *Security* feature, a UPnP service MUST support *DeviceProtection:1* service . [DPS] specification defines additional security procedures and methods that have to be implemented by a UPnP service.

2.4.3. Parent device and embedded devices

Security feature applies to services supported by a device and not the device itself. In order to protect a Service, at least one of the ancestor devices MUST contain a *DeviceProtection:1* instance. The Service is protected by the *DeviceProtection:1* that is in its nearest ancestor device

2.4.4. Roles

DeviceProtection:1 defines three *Roles*: *Public*, *Basic* and *Admin*. For the purposes of securing a given UPnP DM service, a minimum set of two *Roles* MUST be implemented with the following definitions based on [DPS]:

- *Public*: this is the less restrictive *Role*. It provides open access and it is associated to non-secured sessions: any Control Point is implicitly considered to have this *Role* in addition to any other *Roles*that are explicitly assigned to it. If a *Public Role* is recommended for an action, it implicitly means that any other *Role* is also allowed to run the action. Note that it explicitly means that it allows access to any Control Point without the need to support security and *DeviceProtection* service.

- *Admin*: this is the most restrictive *Role*. It provides secured access to any action and argument values and to entire *Data Model*. Any Control Point with such *Role* MUST have the access rights to all the resources of the UPnP device. This *Role* MUST NOT be included in the restricted *Role List*.

The *Role* name *Public* and *Admin* MUST be implemented by UPnP DM services whenever OPTIONAL *Security* feature is enabled.

For the purpose of specifications, the three *Role*s described in *DeviceProtection:1* will be used to illustrate the different security levels. The name of the *Basic Role* MAY be changed but the following SHOULD apply:

- *Basic Role* provides some level of security (more restrictive than *Public*) by at least requiring secured sessions. It is associated to some actions/argument values. The privileges assigned to this *Role* depend on the implementation and deployment choices. When OPTIONAL *Basic Role* is implemented, it SHOULD comply with the recommendation herein specified.

Specifies additional OPTIONAL *Role*s MAY be implemented to support other device management *Security* use cases:

- *dm:ThirdPartyAdmin*: if specified, this *Role* MUST be assigned to a third party (instead of *Admin*) that provides the device for management on behalf of the end user. The privileges assigned to this *Role* MUST be a subset of privileges of the *Admin Role* (possibly coincident).*dm:UserAdmin*: if specified, this *Role* MUST be assigned to the end user (instead of *Admin*) for local administration of the device. The privileges assigned to this *Role* MUST be a subset of privileges of the Admin *Role* (possibly coincident).

The decision to implement *dm:ThirdPartyAdmin*, *dm:UserAdmin* and a deployment scenario in terms of *Roles* for actions depends on the usage context or business model. For instance implementation choice may be linked to fact that a device is rented or owned by a user. Consequently there are different possible deployment models. For example:

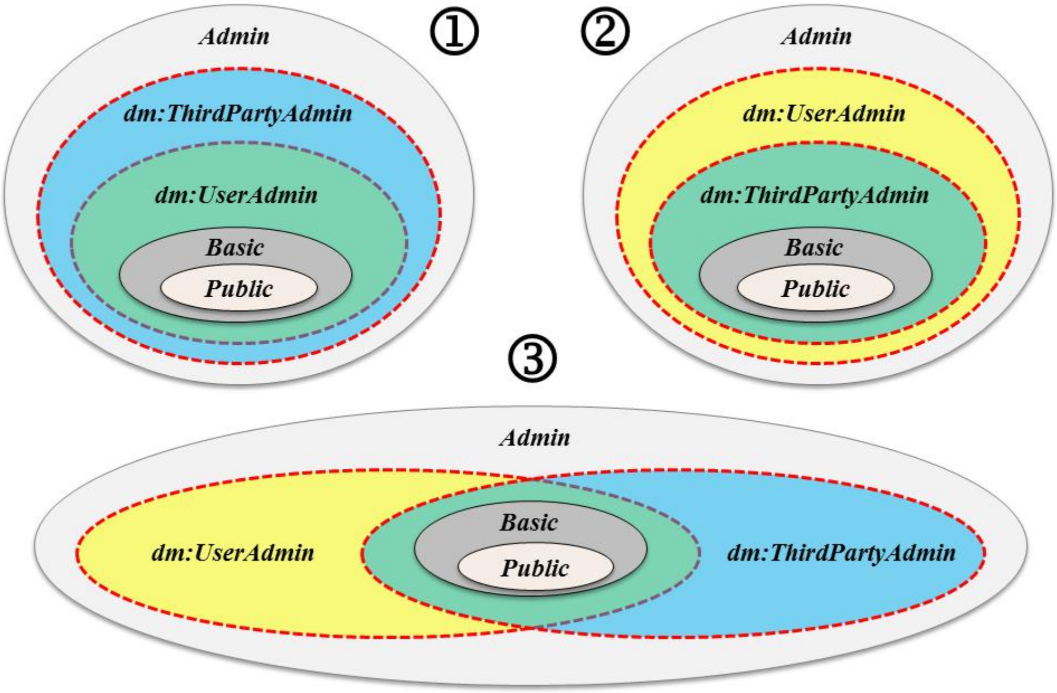

Figure 2-2: Different scenarios for *Role* deployment

The figure above shows different scenarios:

1. This illustrates a service provider(*dm:ThirdPartyAdmin*) rights to manage a rented device such as a home gateway, where
   - The *dm:ThirdPartyAdmin* privileges are less or equal to the *Admin* privileges.
   - The *dm:UserAdmin* privileges are less or equal to the *dm:ThirdPartyAdmin* privileges.
   - The *Basic* privileges are less or equal to *dm:UserAdmin* priviledges.
2. This illustrates a user (*dm:UserAdmin*) rights to manage an owned devices, where
   - The *dm.Userdmin* privileges are less or equal to the *Admin* privileges.
   - The *dm:ThirdPartyAdmin* privileges are less or equal to the *dm.3UserAdmin* privileges.
   - The *Basic* privileges are less or equal to *dm:ThirdPartyAdmin* priviledges.
3. This illustrate a case where a third party and a user have the right to manage sub-sets of a device, possibly with common sub-sets, where
   - Both The *dm:UserAdmin* and *dm:ThirdPartyAdmin* privileges are less or equal to the *Admin* privileges.
   - The *dm:UserAdmin* has some privileges that may be shared with the *dm:ThirdPartyAdmin*. Both *Roles* may have privileges that are not assigned to the other.
   - The *Basic* privileges are less or equal to *dm:ThirdPartyAdmin* and/or *dm.3UserAdmin* priviledges.

Other deployment models can be defined with only two administration *Roles*, either the couple {*Admin*; *dm:ThirdPartyAdmin*} or the couple {*Admin*; *dm:UserAdmin*}. In fact implementers are free to decide whenever to add OPTIONAL *Role* for recommended *Role*s or recommended *Restricted Roles*:

- For example, in the *GetSupportedParameters()* action ([CMS]), the *dm:ThirdPartyAdmin* and *dm:UserAdmin Roles* could be recommended in the *Restricted Role List* (see chapter 2.4.5) because there might be part of the *Data Model* which must be hidden to both in a mutually exclusive way.
- There could be a set of parameters that is accessible only to *dm:ThirdPartyAdmin* (and *Admin*), whereas another set of parameters is accessible only to *dm:UserAdmin* (and *Admin*). But, for some simpler implementation, a less restrictive requirement could be enough, having both the *dm:ThirdPartyAdmin* and *dm:UserAdmin Roles* in the *Role List*: therefore they both have complete access to all the supported parameters.

Due to the heterogeneous scenarios, [BMS], [CMS] and [SMS] specifications will not list OPTIONAL *Roles* such as *dm:ThirdPartyAdmin* and *dm:UserAdmin* for actions and will limit recommendations to the three mandatory *Roles*, *Public*, *Basic* and *Admin,* as defined by Device Protection.

In case *dm:ThirdPartyAdmin* and/or *dm:UserAdmin* are implemented, the hierarchy between the different *Roles* from the most to the less privileged MUST be:

1. *Admin*
2. *dm:ThirdPartyAdmin* or *dm:UserAdmin*
3. *Basic*
4. *Public*

To conclude, implementers are free to decide which *Roles* should be implemented when supporting *Security* feature. Only *Public* and *Admin* are MANDATORY, other intermediate OPTIONAL *Role* names depends on *Security* implementation requirements and there should usually not be a clear hierarchy interpreted from the *Role* names. Each UPnP DM service will provide the necessary actions to list unconditional and conditional actions that each *Role* is allowed to invoke.

Note that it is strongly suggested to respect the recommended propositions (e.g. security level to run actions) as they represent an appropriate implementation of a secured device management.

2.4.5. Role List and Restricted Role List

Device Protection:1 defines *Roles List* and *Restricted Role List*. As this is a key concept for UPnP DM *Security* feature, the following is a reminder of its usage as defined in [DPS] specification.

Actions that do not return sensitive information, change the device configuration, or affect normal device operation are referred to as *Non-Restrictable* actions and can always be invoked by all control points. For example, diagnostic actions that can facilitate interoperability, or actions that event a state variable that cannot be protected by *DeviceProtection:1*. All other actions are referred to as *Restrictable* actions.

If the OPTIONAL *Security* feature is not supported, all actions can be invoked by all control points. If the *Security* feature is supported, *Restrictable* actions can only be invoked if the control point is appropriately authorized

The terms *Role List* and *Restricted Role List* are defined by *DeviceProtection:1*. Each action has an associated *Role List*; a control point that possesses a *Role* in the *Role List* can unconditionally invoke the action. The *Role* names in *Role List* have OR semantics (e.g. "*Admin Basic*"). This means that a Control Point only needs to be authorized with one of the *Roles* in *Role List* to use the action. The *Roles* in *Role List* are granted access for all argument values.

Some actions also have a *Restricted Role List*; a control point that does not possess a *Role* in the *Role List* but does possess a *Role* in the *Restricted Role List* might be able to invoke the action (it's up to the action definition to specify this).

The *Public Role* is defined by *DeviceProtection:1*. All control points automatically possess the *Public Role*, and all control points can unconditionally invoke all actions that have a *Role List* of "*Public*". Therefore:

- If the *Security* feature is not supported, behavior is the same as if the feature was supported and all actions had a *Role List* of "*Public*" and an empty *Restricted Role List*.

- Regardless of whether or not the *Security* feature is supported, all *Non-Restrictable* actions have a *Role List* of "*Public*" and an empty *Restricted Role List*.

For *Restrictable* actions, this specification defines RECOMMENDED values for the *Role Lists* and *Restricted Role Lists*. Device manufacturers are permitted to choose different values.

2.4.6. Access Control List

The *DeviceProtection* access control model is based on an access control list (ACL) that assigns Device-specific and service-specific *Roles* to Control Point and User Identities. Each Device maintains its own ACL, and there is no explicit support for automatically sharing or synchronizing ACLs across multiple Devices. *Roles* correspond to permissions to perform specific SOAP actions or to access resources (*Data Model*, deployment and execution units).

UPnP DM service specifications include RECOMMENDED *Roles* to perform actions, but Devices MAY ignore those recommendations. Therefore *BasicManagement:2*, *ConfigurationManagement:2* and *SoftwareManagement:2* services provide the ability to query a service to discover the *Roles* required to perform specific actions or access specific resource.

Note that ACL cannot be changed dynamically; consequently any ACL *Data Model* modification in an existing UPnP DM device will be a consequence of a software update or an implementation specific mechanism.

2.4.7. General Security Policies

If *Public* is the MANDATORY *Role List*, then TLS tunnel is OPTIONAL when invoking the action and the action is non-restrictable. This allows unsecured Control Point to access to *Public* actions and resources of a device.

2.4.8. Backward compatibility

Any UPnP DM version 2 device that supports OPTIONAL *Security* feature might reject an action invocation that would have been permitted by a given UPnP DM version 1device..
This is not a backward compatibility issue since returned error codes are already defined in UDA (see [UDA]). UPnP DM version 2 and UPnP DM version 1 are based on UDA.
Security error codes are used when (list is not exhaustive):

- action does not return information and CP role isn't appropriate;
- information returned is "all or nothing" and CP role isn't appropriate;
- information requested is supposed to be hidden to the CP role used to invoke the action.

2.5. Theory of Operation

After a Control Point discovers UPnP device implementing UPnP DM device or services within the home network, it can invoke various actions defined within services listed in Table 2-1: Device Requirements. In the following subsections, a set of basic usage scenarios are introduced.

All examples in this section are provided with reference to the *ManageableDevice* device type, but they apply also to other types of UPnP device that include the corresponding UPnP DM services.

2.5.1. Option 1: UPnP DM services contained within ManageableDevice

The following two figures illustrate the two main options for deploying *ManageableDevice* as a root device,

Figure 2-2 outlines the minimum deployment option, where only required UPnP DM services are provided:

- The *BasicManagement* service (BMS) provides basic features for administration and diagnostics;
- The *ConfigurationManagement* service (CMS) provides management of UPnP DM *Common Objects*.

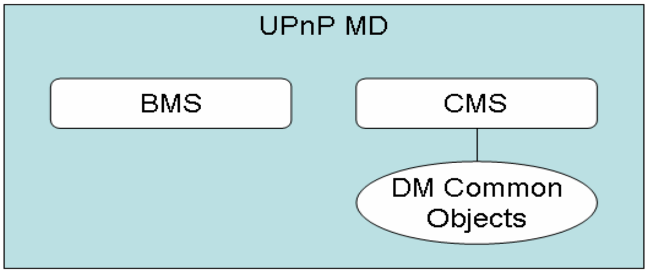

Figure 2-3: UPnP *ManageableDevice* (minimum implementation)

Figure 2-3 outlines the extended deployment option, where all UPnP DM services are provided:

- The *SoftwareManagement* service (SMS) targets an EE and adds management functions for firmware upgrade and software management.

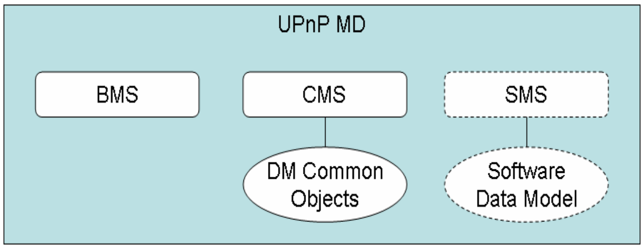

Figure 2-4: UPnP *ManageableDevice* (extended implementation)

2.5.2. Option 2: UPnP DM services included in other types of UPnP device

UPnP DM services are designed to be included in other types of UPnP device. The following figures illustrate some possible deployment options for UPnP DM services inside an arbitrary UPnP device.

Figure 2-4 below outlines the most general deployment option of the complete set of UPnP DM services within UPnP Device X (which also provides its own services):

- BMS, CMS and SMS provide the corresponding management features for UPnP Device X;

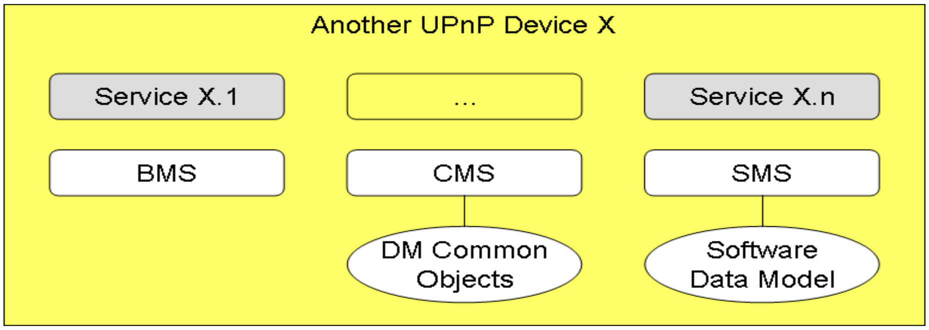

Figure 2-5: Example UPnP device including UPnP DM services

Figure 2-5 outlines a minimal deployment option, where only one UPnP DM service (BMS) is included within UPnP Device X (which also provides its own services):

- BMS provides basic management features for the targeted UPnP device.

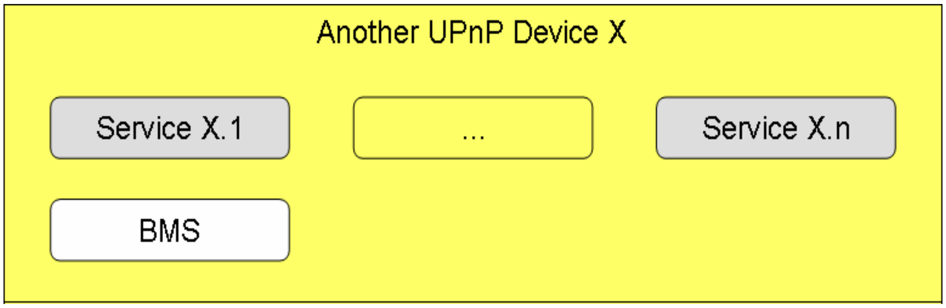

Figure 2-6: Example UPnP device including BMS only

Figure 2-6 outlines another minimal deployment option, where only one UPnP DM service (CMS) is included within a UPnP Device X (which also provides its own services):

- CMS provides configuration management features for specific parameters defined for the targeted device (Device X *Data Model*);

- UPnP DM *Common Objects* can be OPTIONALly supported.

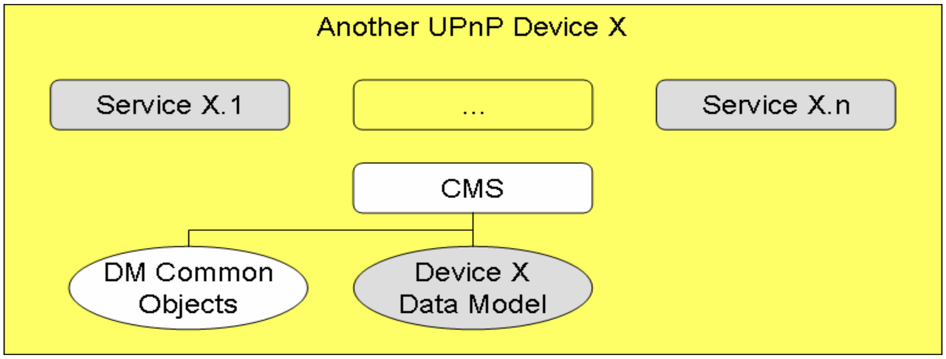

Figure 2-7: Example UPnP device including CMS only

If a standard UPnP device adopts this option, it can also define its own *Data Model* to be manipulated by CMS.

Figure 2-7 outlines another minimal deployment option, where only one UPnP DM service (SMS) is included within a UPnP Device X (which also provides its own services):

- SMS provides software management features for the targeted device;
- The UPnP DM *Software Data Model as defined in SoftwareManagement* service can be OPTIONALly supported (not shown in figure). In that case CMS will also be supported (not shown in figure).

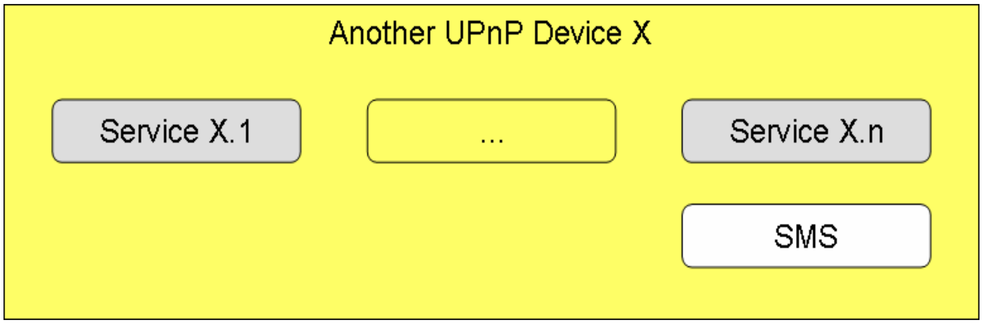

Figure 2-8: Example UPnP device including SMS only

2.5.3. Option 3: Multiple root devices

Figure 2-8 illustrates the deployment option where multiple root UPnP devices are available in a single physical device (product), and one of these is the *ManageableDevice*:

- UPnP Device X and Device Y provide their own functionality;
- UPnP *ManageableDevice* provides management features for either the physical device or the other UPnP root device(s): this choice depends upon the implementation or can be specified by the UPnP Working Committee defining such architecture for these UPnP devices.

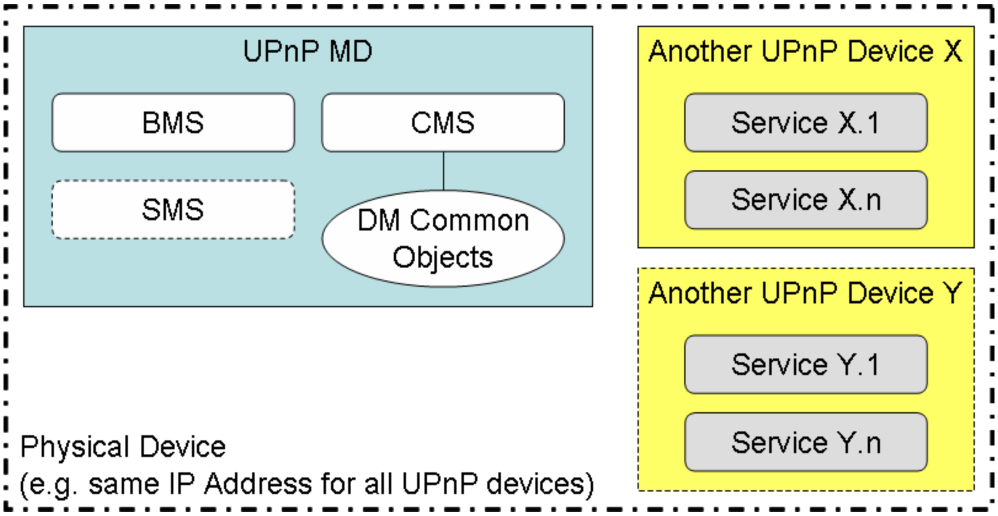

Figure 2-9: Multiple UPnP Root Devices

2.5.4. Option 4: Embedded *ManageableDevice*

Figure 2-9 illustrates the deployment option where the *ManageableDevice* is embedded within another UPnP device:

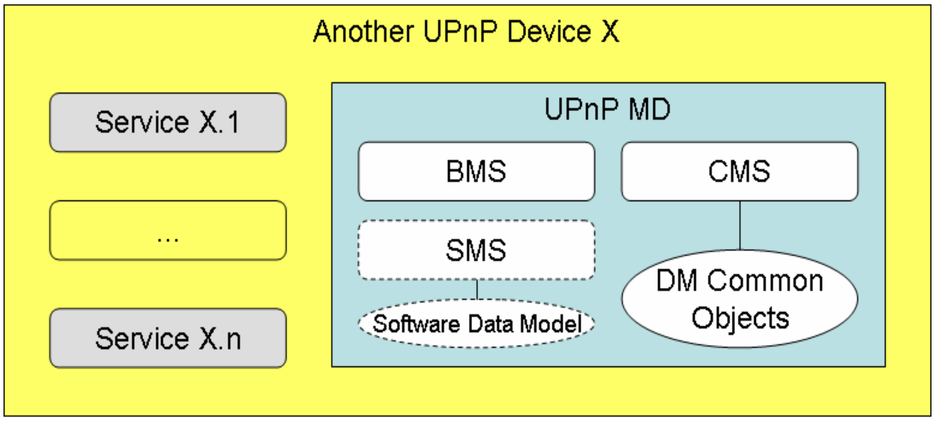

Figure 2-10: UPnP *ManageableDevice* as embedded device

2.5.5. Example: Deployment of Device Protection in the root device

Figure 2-13 illustrates deployment scenario of Device Protection [DPS] in a root device:

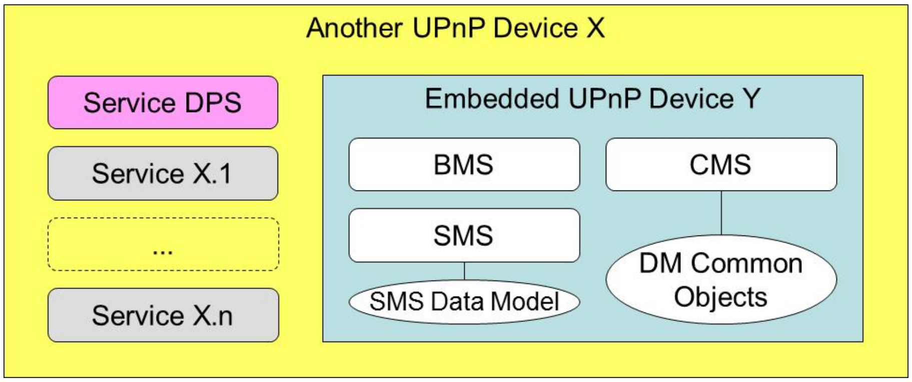

Figure 2-11: An example of *Security* feature implementation in a root device

- Root device X supports a *Security* feature (*DeviceProtection* service) defined at root level. This *Security* feature (Service DPS) has a scope of the entire root device since there are no other DPS instances in embedded devices.

- Embedded device Y is using the *Security* feature defined at root level to secure its UPnP Device Management:2 actions.

2.5.6. Example: Deployment of Device Protection in an embedded device

Figure 2-14 illustrates deployment scenario of Device Protection [DPS] in an embedded device:

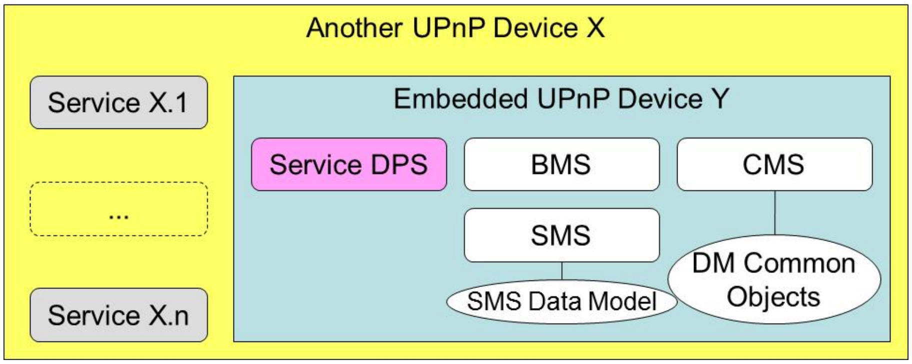

Figure 2-12: An example of *Security* feature implementation in an embedded device

- Root device X does not implement a *Security* feature and Service X.1…Service X.n cannot use the Service DPS (*DeviceProtection* service) defined in the embedded device Y level.
- Embedded device Y is using the ACL (Service DPS.1) defined at its level to secure its UPnP Device Management services (BMS, CMS and SMS).

2.5.7. Option 5: *ManageableDevice* embedding other UPnP devices

Figure 2-10 illustrates the deployment option where the *ManageableDevice* is embedding other UPnP devices:

- The UPnP *ManageableDevice* actions affect also the embedded UPnP devices;
- Indeed additional data models (not shown in the figure) can be added for addressing specific management requirements of the embedded devices.

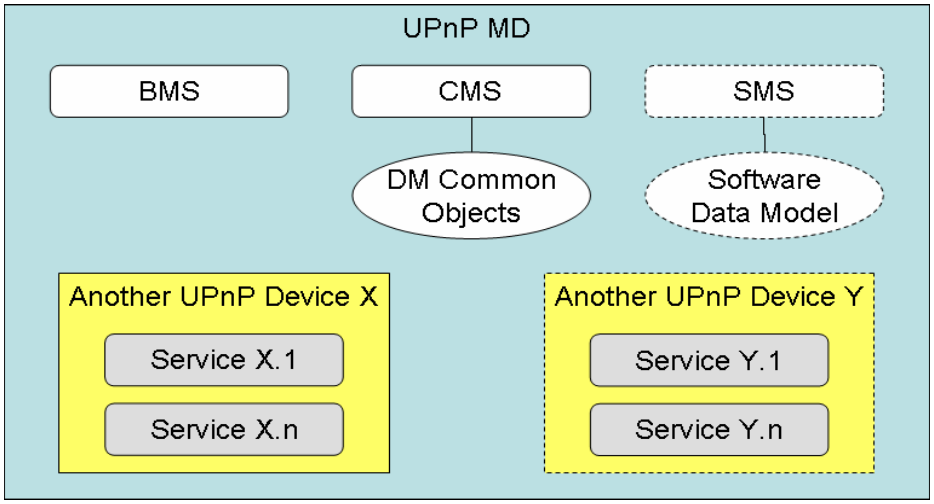

Figure 2-13: UPnP *ManageableDevice* embedding other UPnP devices

2.5.8. Example: A Complex Deployment

Figure 2-11 is an example of a complex deployment where some of the previous deployment options are mixed, in order to get the desired combination and scope for the different device management domains:

- Each UPnP DM service targets its parent UPnP device.

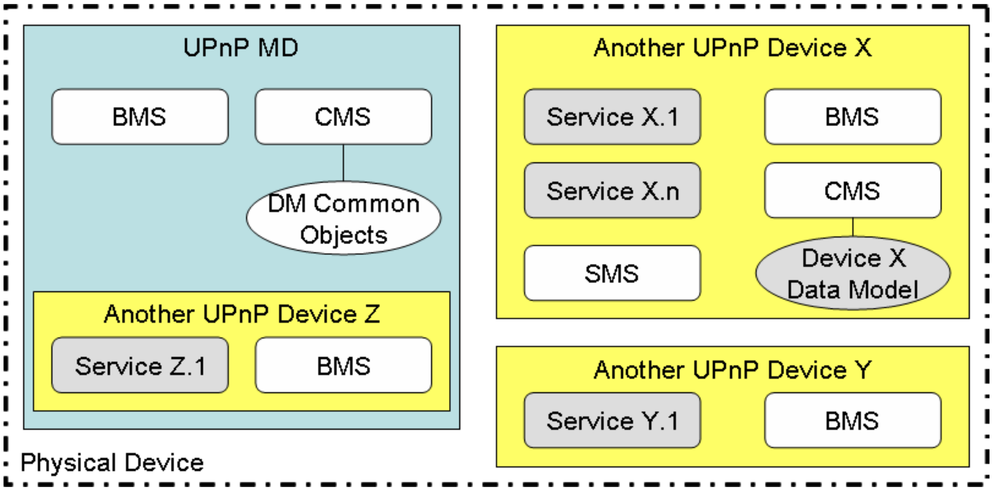

Figure 2-14: An example of complex implementation with device management features

3. XML Device Description

```
<?xml version="1.0"?>
<root xmlns="urn:schemas-upnp-org:device-1-0">
  <specVersion>
    <major>2</major>
    <minor>0</minor>
  </specVersion>
  <device>
    <deviceType>urn:schemas-upnp-org:device:ManageableDevice:2</deviceType>
    <friendlyName>short user-friendly title</friendlyName>
    <manufacturer>manufacturer name</manufacturer>
    <manufacturerURL>URL to manufacturer site</manufacturerURL>
    <modelDescription>long user-friendly title</modelDescription>
    <modelName>model name</modelName>
    <modelNumber>model number</modelNumber>
    <modelURL>URL to model site</modelURL>
    <serialNumber>manufacturer's serial number</serialNumber>
    <UDN>uuid:UUID</UDN>
    <UPC>Universal Product Code</UPC>
    <iconList>
      <icon>
        <mimetype>image/format</mimetype>
        <width>horizontal pixels</width>
        <height>vertical pixels</height>
        <depth>color depth</depth>
        <url>URL to icon</url>
      </icon>
      XML to declare other icons, if any, go here
    </iconList>
    <serviceList>
      <service>
        <serviceType>
            urn:schemas-upnp-org:service:BasicManagement:2
        </serviceType>
        <serviceId>urn:upnp-org:serviceId:BasicManagement</serviceId>
        <SCPDURL>URL to service description</SCPDURL>
        <controlURL>URL for control</controlURL>
        <eventSubURL>URL for eventing</eventSubURL>
      </service>
      <service>
        <serviceType>
            urn:schemas-upnp-org:service:ConfigurationManagement:2
        </serviceType>
        <serviceId>urn:upnp-org:serviceId:ConfigurationManagement</serviceId>
        <SCPDURL>URL to service description</SCPDURL>
        <controlURL>URL for control</controlURL>
        <eventSubURL>URL for eventing</eventSubURL>
      </service>
      <service>
        <serviceType>
            urn:schemas-upnp-org:service:SoftwareManagement:2
        </serviceType>
        <serviceId>urn:upnp-org:serviceId:SoftwareManagement</serviceId>
        <SCPDURL>URL to service description</SCPDURL>
        <controlURL>URL for control</controlURL>
        <eventSubURL>URL for eventing</eventSubURL>
        <serviceType>
            urn:schemas-upnp-org:service:DeviceProtection:1
        </serviceType>
        <serviceId>urn:upnp-org:serviceId:DeviceProtection</serviceId>
        <SCPDURL>URL to service description</SCPDURL>
        <controlURL>URL for control</controlURL>
        <eventSubURL>URL for eventing</eventSubURL>
      </service>
            Declarations for other services defined by a UPnP Forum working
      committee (if any) go here
      Declarations for other services added by UPnP vendor (if any) go here
    </serviceList>
    <deviceList>
      Description of embedded devices defined by a UPnP Forum working
      committee (if any) go here
      Description of embedded devices added by UPnP vendor (if any) go here
    </deviceList>
    <presentationURL>URL for presentation</presentationURL>
  </device>
</root>
```

Appendix A: specification changes

Changes between specification document version 1 and version 2:

- Section 2.4 added: description of *Security* feature using *DeviceProtection* service.
- Section 2.5.5 added: example of *Security* implemented in a root device.
- Section 2.5.6 added: example of *Security* implemented in an embedded device.
- Section 3 added: *DeviceProtection* in the XML device description document.