

Introduction to UPnP Device Management

March 2012



WC co-chair

Kiran Vedula (Samsung)

Xavier Roubaud (Orange)

UPnP DM Working Committee participants

- Alcatel-Lucent
- ATT
- Cisco
- Deutsche Telekom
- France Telecom / Orange
- Huawei
- Nokia
- PACE (ex 2Wire)
- Philips
- Prosys
- Samsung
- Telecom Italia

WC Co-Chairs

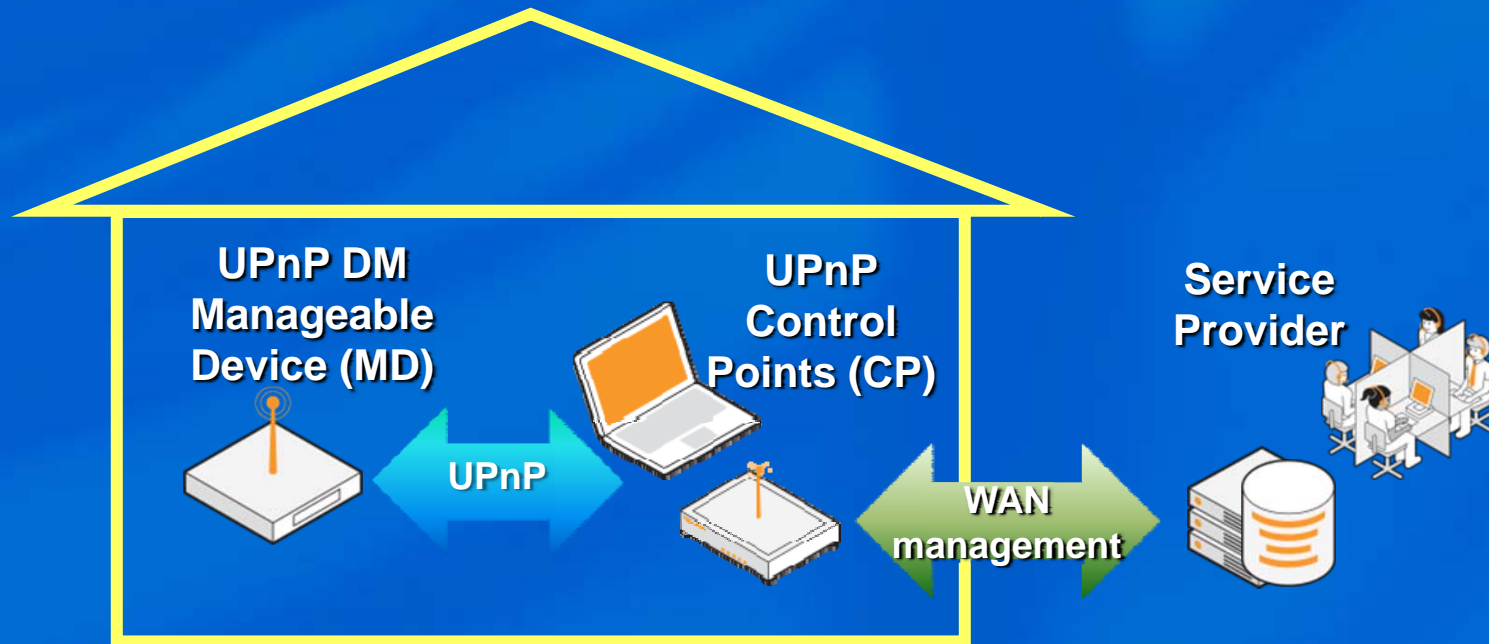


Terms definitions

- **UPnP *ManageableDevice*** (MD): UPnP device defined by the UPnP Device Management Working Committee. It represent the device that will be managed.
- **Control Point** (CP): a software capable of calling Manageable Device actions. In other words, the software that is going to manage the *ManageableDevice*.
- **Deployment Unit** (DU): software package which can be installed, uninstalled or updated. Such binary unit that can be individually deployed on the execution environment. A deployment unit consists of resources such as library files, functional execution units, configuration files (packages, jar files, bundles, assemblies, etc)
- **Execution Unit** (EU): software entity which can be started or stopped. Once started, this functional entity initiates processes to perform tasks or provide services, until that it is stopped. Execution units are deployed by deployment units (services, scripts, software components, MIDlets, etc)

Purpose of UPnP Device Management

- UPnP DM is a Device Control Protocol based on UPnP standard
- Objective is to standardize management operation of LAN IP devices such as troubleshoot, configuration or software updating



UPnP DM services

- Three types of management services are defined in UPnP Device Management:

Basic Management Service (BMS)

- reboot and baseline reset a device
- run self-test in order to diagnose problems
- manage logs (enable, disable, retrieve)

Configuration Management Service (CMS)

- discover data model and current instances
- data model manipulation (read, write, create, delete)
- set alarm on value change
- get or retrieve values
- get and set parameter's attributes
- event on parameter value change
- create or delete multi-instance objects instances (like BBF table rows)

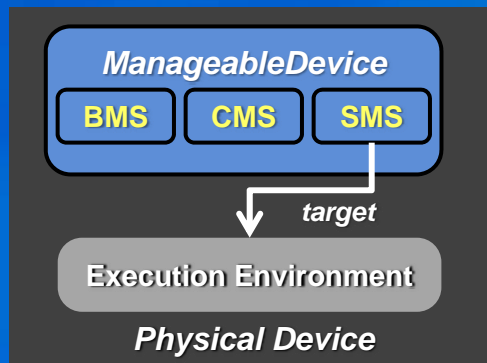
Software Management Service (SMS)

- manage embedded software or firmware
- install, uninstall and update software modules (Deployment Units)
- start and stop software entities (Execution Units)
- software data model describing DUs and EUs

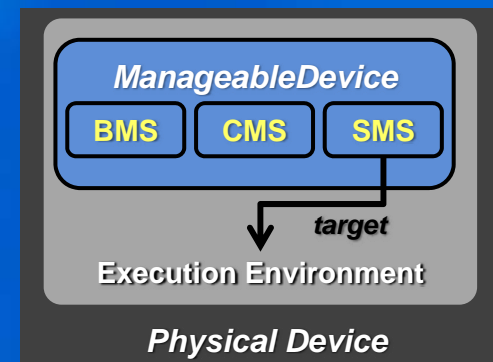
ManageableDevice Deployment

- A **ManageableDevice (MD)** corresponds to a physical or a virtual device with an associated data model and with potentially software entities to be managed
- An Execution Environment could be an Operating System, a Java VM, a virtual machine, etc.
- **BMS** and **CMS** are associated to a Device and **SMS** is associated to an Execution Environment

MD **outside** the Execution Environment



MD **inside** the Execution Environment



Deployment of UPnP DM services

- **ManageableDevice:2** device is a container for device management service with at least:
 - **BasicManagement:2** service.
 - **ConfigurationManagement:2** service that contains the Common Objects which is a data model defining a minimal set of parameter to engage device management operations.
 - **SoftwareManagement:2** service is **optional**.
 - **DeviceProtection:1** service (security and Access Control Lists) is **optional**. Security has been added to prevent any Control Point to run any actions on any UPnP DM device.
- **However each service may be used independently**
 - Any UPnP device can embed one or several UPnP DM services.
 - ex: an Internet Gateway would only need to embed BMS in order to run bandwidth tests.
 - If it embeds the Common Objects, or a data model, it must embed the **ConfigurationManagement:2** service.

Usage examples

- Each action is available locally to the user or remotely to the service provider via a proxy (i.e. : TR-069 to UPnP DM proxy)
- Examples of Device **maintenance / Troubleshooting / Diagnostics**
 - *reboot or reset a device*
 - *Initiate a self-test diagnostic in order to troubleshoot*
 - *runs a bandwidth test to figure out why the video streaming is not smooth*
- Examples of **Provisioning/Configuration**
 - *modify wireless security*
 - *change the password of a service*
 - *monitor alarms and parameters*
- Examples of **Software** management
 - *update a firmware*
 - *install a new codec*
 - *start a service*

BasicManagement Service

● Actions

- *Reboot()*, *BaselineReset()*, *GetDeviceStatus()*
- *SetSequenceMode()*, *GetSequenceMode()*: when a CP is running or planning to run several actions
- *InterfaceReset()*, *GetInterfaceResetResult()*
- *GetLogURIs()*, *SetLogInfo()*, *GetLogInfo()*: to manage log files
- *GetACLData()*: to retrieve security information
- **Test actions**
 - *Ping()*, *GetPingResult()*
 - *NSLookup()*, *GetNSLookupResult()*
 - *Traceroute()*, *GetTracerouteResult()*
 - *GetBandwidthTestInfo()*, *BandwidthTest()*, *GetBandwidthTestResult()*,
 - *SelfTest()*, *GetSelfTestResult()*
 - *GetTestIDs()*, *GetActiveTestIDs()*, *GetTestInfo()*, *CancelTest()*

● State Variables (lists)

- **DeviceStatus**: parent device status, date/time of last change and additional information
- **SequenceMode**: indicates whether a Control Point is executing a sequence of actions
- **TestIDs** and **ActiveTestIDs**: list tests executed and test that are not yet completed.
- ...

ConfigurationManagement Service

● Actions

- ***GetSupportedDataModels(), GetSupportedParameters()*** : to retrieve device datamodel and parameters
- ***GetInstances(), CreateInstance(), DeleteInstance()***: to manage multiple instances of nodes
- ***GetValues, GetSelectedValues(), SetValues()***
- ***GetAttributes(), SetAttributes(), GetAttributeValuesUpdate()***
- ***GetInconsistentStatus()***
- ***GetCurrentConfigurationVersion()***
- ***GetConfigurationUpdate(), GetSupportedDataModelsUpdate(), GetSupportedParametersUpdate()***
- ***GetAlarmsEnabled(), SetAlarmsEnabled()***: to manage alarm on parameters
- ***GetACLData()***: to retrieve security information

● State Variables (lists)

- ***CurrentConfigurationVersion***: retrieve changes between updates
- ***SupportedDataModelUpdate, SupportedParameterUpdate, AttributeValueUpdate***: data model update indicators.
- ...

● Framework to integrate Data model definitions

SoftwareManagement Service

● Actions

- **GetDUInfo()**: information on existing deployment units
- **GetEUInfo()**: information on existing execution units (firmware is identified by EUID=0)
- **GetDUIDs(), Install(), Update(), Uninstall()**: to manage deployment units
- **GetEUIDs(), GetActiveEUIDs(), GetRunningEUIDs(), Start(), Stop()**: to control execution units
- **GetOperationInfo()**
- **GetOperationIDs**
- **GetErrorEUIDs()**
- **GetACLData()**: to retrieve security information

● State Variables (lists)

- **OperationIDs**: on going software management operations
- **DUIDs**: IDs of all unresolved or installed DUs. A DU may contain multiple EUs.
- **EUIDs**: IDs of all installed EUs. Each EU belongs to one only DU
- **ActiveEUIDs**: IDs of all started EUs.
- **RunningEUIDs**: IDs of all EUs observed as running.
- **ErrorEUIDs**: IDs of all EUs in error.
- ...

Working Committee status

● UPnP DM v1

- UPnP Device Management version 1 published in July 2010 on www.upnp.org
- Orange Labs made an UPnP DM:1 reference implementation using pupnp opensource stack and published it in open source (Apache licence) on SourceForge.

● UPnP DM v2

- UPnP Device Management version 2 published in February 2012 on www.upnp.org.
- Orange Labs made a UPnP DM:2 reference implementation (including security) and should publish it in open source on SourceForge
- Security has been added to Device Management v2 since, in UPnP DM v1, any Control Point could potentially run any actions on any UPnP DM device.
 - Security implementation is based on UPnP Device Protection.
- UPnP DM:V2 also includes non security related features
 - Bandwidth tests,
 - Alarm management on parameter value change.



For the interconnected lifestyle