Overview of revisions

ProSyst Remote Manager 6.1

Features

Bosch

Bosch Software Innovations

INST/PSY
Anastas Chobanov
Status: July 2017
### Overview of revisions

<table>
<thead>
<tr>
<th>Date</th>
<th>Changes/Comments</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>17-07-2017</td>
<td>Migration to the new layout /Naming changes.</td>
<td>Anastas Chobanov</td>
</tr>
</tbody>
</table>
Content

1 REMOTE MANAGER FUNCTIONAL OVERVIEW ................................................................. 5

2 DEVICE MANAGEMENT FEATURES ............................................................................ 6

2.1 Remote Manager Generic Device Management Framework ......................................... 6

2.2 OSGi Device Management ............................................................................................ 7

   2.2.1 OSGi Device Registration and Initial Provisioning ....................................................... 8
   2.2.2 Remote Manager Management Protocol ...................................................................... 8
   2.2.3 Device Directory ......................................................................................................... 9
   2.2.4 Retrieving and Storing of the Device Configuration .................................................... 10
   2.2.5 OSGi Framework Settings Management ................................................................... 11
   2.2.6 OSGi Bundle Management ........................................................................................ 11
   2.2.7 Generic Software Management .................................................................................. 12
   2.2.8 OS-Level Management and Monitoring ..................................................................... 12
   2.2.9 Bundle Configuration (Settings) Management .............................................................. 12
   2.2.10 Preferences Data Management and Data Synchronization ........................................... 12
   2.2.11 Providing of Custom Management Interfaces ............................................................ 13
   2.2.12 Home Network Devices Management .................................................................... 13
   2.2.13 Retrieving Device Logs ............................................................................................ 13
   2.2.14 Fault Alerts ................................................................................................................ 14
   2.2.15 Application Monitoring ............................................................................................. 14

2.3 TR-069 Device Management ......................................................................................... 14

2.4 TR-069 OSGi Device Management .............................................................................. 15

2.5 Mobile (OMA DM-Based) Device Management .......................................................... 15

2.6 Mobile Device Types ..................................................................................................... 16

   2.6.1 Generic OMA DM Support .......................................................................................... 16
   2.6.2 OSGi Mobile Management Objects .......................................................................... 17

3 SOFTWARE REPOSITORY ............................................................................................ 19

4 REMOTE MANAGER GUI — MANAGEMENT CONSOLE .............................................. 20

5 API AND INTEGRATION INTERFACES ........................................................................ 21

6 SECURITY ....................................................................................................................... 22

6.1 User Authentication and Authorization ......................................................................... 22

   6.1.1 User Authentication .................................................................................................... 22
   6.1.2 User Authorization ..................................................................................................... 22

6.2 Secure Network Communication .................................................................................. 23
Remote Manager 6.1

6.3 Certificate Management ................................................................. 23

7 REMOTE MANAGER SCALABILITY AND HIGH AVAILABILITY .................. 24
  7.1 Single-server configuration ................................................................. 24
  7.2 Dynamic-clustered configuration ....................................................... 25
  7.3 Mixed-clustered configuration ............................................................ 26

8 PLATFORM REQUIREMENTS ..................................................................... 27
  8.1 Hardware Requirements .................................................................... 27
  8.2 OS & JVM Requirements ................................................................... 27
  8.3 Supported OSGi Frameworks for Remote Manager OSGi Management Agents ................................................................. 27
  8.4 Database Requirements ..................................................................... 27
    DB Storage Space .................................................................................. 28
  8.5 Supported Non-OSGi Devices ............................................................... 28

9 APPENDIX A: RESOURCES .................................................................... 29
1 Remote Manager Functional Overview

ProSyst’s IoT platform provides you with a set of ready services that can be used as building blocks for your IoT applications.
2 Device Management Features

ProSyst’s Remote Manager is a backend management and monitoring system, supporting centralized configuration, software provisioning and service operation support for various class of devices based on (but not limited) to OSGi, OMA-DM and/or TR-069 standards. In short, the Remote Manager is applicable to realize these use cases:

- Platform, Application & Service Lifecycle Management
- Firmware & File Update
- Remote Configuration & Software Provisioning
- Remote Diagnostics (life status checks, logging, monitoring, etc.)
- Remote Security Administration

The Remote Manager is not limited to these use cases as support for additional cases (standardized or customized) can be added at any time.

Remote Manager is unique in its openness, boosting rapid adoption of new requirements and integration with 3rd party systems. The open design has the following dimensions:

- Open for additional (i.e. custom) management protocols
- Open for custom business logic extensions hosted on the Remote Manager
- Open for integration with 3rd party systems through an extensive set of APIs (Java, REST/HTTP, SOAP or JEE JCA based APIs)
- Open for customized management consoles (UIs)

Remote Manager - the ProSyst Backend management system caches all the sensors/actuators information. This allows automation of rules not only on the device itself or locally in bridge devices but also in a cloud. The Remote Manager also acts as a proxy for connecting the end user devices like smart phones or tablets to the gateways.

The Remote Manager is in production use in many deployments in various vertical markets such as: Connected Home, Fleet Management, eHealth Devices and IoT scenarios.

This document describes the main features of the Remote Manager system.

2.1 Remote Manager Generic Device Management Framework

The foundation of the Remote Manager system is a general model and framework for management and monitoring of various device types in a uniform way. The Generic Device Management framework employs a general abstraction for device representation and a number of basic features, which are common for management and monitoring of any device type - regardless of the concrete device capabilities, the network communication protocols and the management interfaces supported by the device.

The interface to the devices of particular type is provided by a set of plugins (adapters), which adapt the concrete device management interface (e.g. OMA DM, TR-069, etc) supported by the devices to the common device model of the Remote Manager.
This model allows extending of the Remote Manager with support of new device types with minimal effort, leveraging the basic common device management functionality.

Remote Manager provides build-in support for the following device classes:

- devices running OSGi compliant frameworks – like residential gateways, home device controllers, set-top-boxes, Telematics head units, etc.
- mobile devices supporting OSGi Mobile R4 specification (JSR - 232)
- mobile devices supporting OMA Device Management protocol (version 1.1 and 1.2)
- devices supporting CPE WAN Management protocol (TR-069) defined by Broadband Forum

### 2.2 OSGi Device Management

Remote Manager supports management of devices equipped with OSGi R5 compliant framework. In order to enable an OSGi device to be managed by the Remote Manager, it has to be equipped with the Remote Manager Management Agent. The Remote Manager Management Agent represents a set of OSGi bundles installed on the target OSGi framework of the device.
The Remote Manager Agent communicates with the management server over ProSyst proprietary messaging protocol. This approach provides rich and efficient management and monitoring functionality of the standard OSGi framework and application resources (bundles, configuration settings, log messages, user preferences, security permissions), as well as management and monitoring of some of the OS level elements, like processes, platform resource utilization (RAM, CPU, network interfaces) and applying of OS level software updates. The protocol is open for extensions providing management and monitoring capabilities to custom platform and/or application specific resources.

2.2.1 OSGi Device Registration and Initial Provisioning

Prosysy’s Remote Manager keeps record of each manageable device. The process of inserting the record about a particular device into the Remote Manager database is referred as “device registration”. Remote Manager supports two general models for OSGi device registration:

Explicit registration – information about a new device is added through the Remote Manager backend either by a custom application (through the Remote Manager API) or manually - by the operator (through the Remote Manager GUI console). This method implies that device is registered before it connects to the Remote Manager for the first time. Remote Manager supports assigning of initial set of configuration settings to the registered device which will be applied upon first device connection.

Automatic – a new OSGi device is automatically registered in the Remote Manager when it connects to the backend server for the first time. Remote Manager automatically retrieves the device identification and configuration information and stores it into the Remote Manager DB. This model can be disabled if there is such requirement (e.g. for security reasons) and then only explicitly registered devices are to be allowed to connect to the Remote Manager system.

In order to become manageable by the Remote Manager, the OSGi device must be provisioned with the Remote Manager Management Agent bundles and suitable configuration settings for establishing management session with the server. This process is called initial provisioning.

Remote Manager supports two general ways for initial provisioning:

Pre-provisioned – the Remote Manager Management Agent and the related configuration can be pre-installed on the device and/or manually provided upon device deployment.

Dynamically provisioned – the Remote Manager / mBS products support the standard OSGi Initial Provisioning mechanism, which defines the provisioning of the OSGi framework with the agent bundles and the related configuration settings needed for connecting to a particular management system (from arbitrary vendor).

In order to start the initial provisioning process, the OSGi Initial Provisioning Service has to be provided with settings for connecting to a provisioning server (e.g. URL, device ID, authentication parameters). These initial provisioning settings can be provided either through static configuration of the OSGi framework, or can be provided dynamically by custom module/application. Remote Manager can act as an OSGi Initial Provisioning server, but it is also possible to use a separate provisioning server.

2.2.2 Remote Manager Management Protocol

The communication between Remote Manager Management Agent running on the manageable OSGi devices and the Remote Manager server is based on ProSyst proprietary communication mechanism - Remote Manager Message Service. The Remote Manager
Message Service utilizes a highly optimized binary protocol - in regards of network data volume, CPU and memory utilization (on device and on server side).

Some of the key features provided by the Remote Manager Message Service are:

**Asynchronous and synchronous** message-based communication.

**Bi-directional message exchange** between the backend server and the OSGi devices - both the server and device can instantly send messages to each other.

**Communication sessions** – the messages are carried over an active management session. The network connection is always initiated by the device, but once the session is open – the server can also send messages to the device (server push). Remote Manager supports different models for session establishment:

**Always online mode** – in this mode, the Remote Manager Management Agent tries to maintain the management session permanently active - once the session is open, it is kept alive indefinitely and if it is closed due to some network problem - the agent periodically tries to re-establish it.

**Periodical connection** – the connection is open at scheduled time intervals (e.g. once per day or week) and after the device updates its status and executes any pending operation for it – the session is closed (or kept active only for some configured time interval of inactivity).

**Event-driven** – the connection may be open in result of fault alert or error log message generated on the device, which must be instantly sent to the backend.

**By user or application request** – through the Remote Manager Management Agent API.

Remote Manager Message Service can operate over various network transport protocols – the design is based on pluggable transport binding components, which handle underlying communication mechanism. Remote Manager provides build-in transport bindings for TCP, SSL/TLS, HTTP(S) and UDP network protocols. Bindings for additional transport protocols can also be added.

**Optional secure transport layer** (in addition to the SSL/TLS based one), which provides secure certificate based authentication and data encryption capabilities on top of any of the available transport bindings. The secure transport layer is useful only if the underlying transport binding does not provide build-in security features (e.g. UDP).

### 2.2.3 Device Directory

Remote Manager provides flexible and convenient mechanisms for grouping, filtering and searching of the registered devices by various built-in and user defined device characteristics and status.

Remote Manager supports two approaches of device grouping (which can be combined):

**Static device grouping** – the devices can be statically organized in user-defined groups. The groups can be organized hierarchically, i.e. one group may have sub-groups, in arbitrary level of nesting. The devices can be moved by the operator from one group to another - at any time.

**Dynamic device grouping** – The Remote Manager supports defining of custom groups based on specified filtering condition. The members of such groups consist of the devices which satisfy the specified condition. The condition can be an arbitrary expression based on any of the static and the dynamic characteristics of the OSGi device maintained by Remote Manager - e.g. online status, device capabilities (e.g. HW, OS, OSGi parameters, etc.), available software components, their runtime status and configuration, etc. including additionally associated custom properties with the devices.
Device grouping helps operator to **monitor and locate devices** through the Remote Manager GUI console and can be also used for targeting of Tasks and Rules (e.g. perform particular task or rule over the devices belonging to a given group).

The Remote Manager also provides features for **direct searching of devices** – either by unique identifier or by more complex filtering condition (as for the dynamic groups).

The Remote Manager supports associating of additional custom properties with each single device or with whole device group. When such property is associated with a device group – it is “inherited” by all sub-groups and devices belonging to that group. Example for such properties can be: location, customer-related data, address, properties for integration with external systems, etc.

These properties are in the form of **key-value pairs**, where the keys are strings and the values can be of type: string, integer, boolean, byte, double or arrays of the above mentioned. Manipulating of these properties can be done either manually - through the Remote Manager GUI console, or by custom applications – through the Remote Manager APIs.

### 2.2.4 Retrieving and Storing of the Device Configuration

Remote Manager collects, retrieves and stores in the database configuration and status information about each device it manages.

The configuration data retrieved and stored by the Remote Manager includes:

- Device capabilities – the characteristics of the device hardware (e.g. CPU model, RAM, Flash/HDD size, etc.) and software platform (OS name and version, JVM vendor and version, OSGi framework vendor, supported specification version, available software component/modules, etc).
- OSGi framework and JVM configuration settings and status.
- Available OSGi bundles and applications, their status and configuration, registered OSGi services and bundle security permissions.
- OS packages, processes, and platform resource utilization statistics.
- Fault alerts, error/warning log messages – the type and content pattern of the log messages that are sent automatically to the server can be configured, but the operator is able to explicitly retrieve any type of log messages on demand.
- Home network devices configuration and status – this includes the data from the devices in the home network accessible through the OSGi framework acting as residential gateway or home device controller.
- Configuration of the additional custom resources.
- Custom monitorable data.

The information is collected and transferred to the management server by the Remote Manager Management Agent – either automatically (implicitly) or on-demand (upon explicit request). The exact type of data which is subject of automatic synchronization can be configured, while the information that is not synchronized automatically - is delivered to the backend on demand.

Device configuration synchronization is an optimized process – it does not lead to transferring of full information about the configuration and status every time (unless the device connects for the first time to the Remote Manager), but only checksums and changes. The mechanism can be described briefly as follows:
1. When the device connects to the backend (Remote Manager server) the changes in the device configuration are transferred to the server and updated in the Remote Manager database. If the device connects for the first time, this step effectively retrieves and stores full device configuration data (which is subject of automatic synchronization).

2. Once the session is established the device (Remote Manager Management Agent) instantly sends update events for changes occurred in the device configuration and state (the data which is subject to automatic synchronization – e.g. application started/stopped, changed properties, fault alerts and application alarms, error/warning log messages, etc). This data is updated in the Remote Manager database and appropriate notifications are sent to the interested applications.

The device configuration data is available to system operators – through the Remote Manager GUI management console, and to the custom backend applications – through the Remote Manager API.

2.2.5 OSGi Framework Settings Management

Remote Manager supports retrieval and remote manipulation of the global OSGi framework parameters:

- **Framework start level** – compatible with OSGi Start Level specification
- OSGi/JVM system properties
- **Framework security policy** – includes management of default bundle permissions and conditional permissions, compatible with the OSGi Permission Admin and Conditional Permission Admin specifications

2.2.6 OSGi Bundle Management

Remote Manager supports management of OSGi based software on two alternative levels of granularity:

- **Low-level bundle management** – provides management of separate OSGi bundles on the target device:
  - Collects information about installed OSGi bundles for each device – installed through either the Remote Manager or via any other source (or preinstalled in the OSGi framework). Provides information about static properties and runtime status for each bundle – manifest headers, ID, location, start level, state (STARTED, RESOLVED, INSTALLED, etc.) as well as registered OSGi services at runtime.
  - Provides means for remote life-cycle management of OSGi bundles – installing of new bundles and starting/stopping/updating/uninstalling of the existing bundles.

- **High-level service application management** – supports logical grouping of a set of (one or more) software components (e.g. OSGi bundles, Deployment Packages) and possibly additional resources and content files and provides ability to provision and manage their lifecycle (update, start/stop, uninstall) together – as a whole application, handling issues like dependency and compatibility resolving, sharing of components between applications, etc.

OSGi bundle management features described in this point operate directly over the standard API of the OSGi framework.
2.2.7 Generic Software Management

Remote Manager supports remote software components deployment and lifecycle management based on the Software Admin service of the ProSyst mBS OSGi Runtime and mBS SDK.

The Software Admin service provides universal facade for installation and life-cycle management of software components of different types – e.g. OSGi bundles, deployment packages, midlets, widgets, OS modules (Debian Packages, IPKGS, etc), Android applications (APK). The Software Admin employs a pluggable model which allows adding support for new software component types to be achieved by developing and installing of corresponding plugins (in the form of OSGi bundles) on the OSGi framework. The model can be used for manipulation of the software not only on the local device running the OSGi framework / Software Admin service but also to other home devices connected to the OSGi device.

2.2.8 OS-Level Management and Monitoring

System monitoring – provides information about the current hardware configuration and current utilization of the platform resources – CPU load, free/occupied RAM, network interfaces – utilized bandwidth and traffic statistics, storage (HDD, flash) – free occupied space.

OS process management – provides information about the currently running OS processes on the target device and status information about each process – PID, command line, status, used memory, priority. Provides the means to start new and stop of the running OS processes.

Native software deployment and update – provides support to retrieve information about installed native packages on devices as well as ability to uninstall/update/uninstall these packages remotely. Remote Manager supports management of Debian and iPKG packages, as well as a ZIP based package format - for devices that do not have build-in package management capabilities.

Distribution of software as ZIP Packages – provides generic way for software distribution. The ZIP package may contain arbitrary file/dir content, which is extracted to a specified directory on the target device. The ZIP package may optionally contain two OS shell scripts – install.sh and uninstall.sh, which are executed by the Remote Manager Management Agent upon ZIP package installation and uninstallation respectively.

2.2.9 Bundle Configuration (Settings) Management

Remote Manager supports retrieving and remote manipulation of the bundle configuration settings though the standard OSGi Configuration Admin service. Configuration settings management is applicable to both system level and application bundles installed on the target OSGi framework.

2.2.10 Preferences Data Management and Data Synchronization

Remote Manager Preferences service is a data storage service compatible with OSGi Preference Service specification. It allows storing of application data associated it particular device or user of the Remote Manager system. The data is maintained persistently in the Remote Manager backend database, but can be also replicated locally on the target devices, where it can be accessed and possibly changed through the OSGi Preferences service APIs. In this way the preferences data can be accessed/manipulated either by backend-side applications (including Remote Manager GUI console) or by device-side applications. The Remote Manager Preferences service provides efficient bi-directional data synchronization.
mechanism for replicating changes in the preferences data between the device-side preferences storage and the central backend data base.

2.2.11 Providing of Custom Management Interfaces

Remote Manager and mBS SDK provide a powerful mechanism for extending the management capabilities of the system, by exporting management interface to custom resources. The mechanism is based on the so called “Control Unit” model. The control unit data model allows unified representation of diverse type of resources (devices, software or hardware components, etc.) along with a common way to manage and monitor these resources.

The key characteristics of the control unit data model are:

**State variables** – represent the status and configuration information of the modeled resource

**Actions** – determine the operations which can be performed over the modeled resource

**Type and identifier of control units** – identifies uniquely the control unit instances

**Hierarchy of control units** – control units can be organized hierarchically, allowing modeling of more complex systems (by decomposing them in sub-subsystem model as separate sub-control units)

**Control Unit Meta-data** – provide information about the state variables and actions supported by particular control unit as well as information to the user

Custom bundles may provide control unit compliant interface to a particular resource by implementing a dedicated java interface and registering it as an OSGi service on the target platform. The Remote Manager agent automatically discovers the exported control units and makes them available for management through the Remote Manager server, including:

- Retrieving of the control unit state (control unit instances and their state variables)
- Storing control unit data in the Remote Manager database
- Viewing and manipulating control units through the Remote Manager GUI console and through the Remote Manager API
- Invoking of the control unit actions – either on a single concrete device or in the scope of the tasks execution or rule.

2.2.12 Home Network Devices Management

The Control Unit model described in the previous paragraph can also be used for providing remote monitoring and control interface to the home network devices (e.g. ZigBee devices) connected to the OSGi device (when it acts as a residential gateway / home controller).

Providing a control unit compliant interface to such devices in the OSGi framework makes them accessible and manageable through the Remote Manager.

2.2.13 Retrieving Device Logs

Remote Manager provides remote interface for retrieving of log messages generated on the managed OSGi devices. The interface is compatible with standard OSGi Log Service and supports both automatic and on-demand retrieval of the device logs.

**Automatic log transfer** – the Remote Manager Log Agent can be configured to automatically send log messages satisfying particular criteria – based on severity level, source and/or log
message text template. Such messages are instantly delivered to the Remote Manager server, where they are stored and made available: to the operator – through Remote Manager GUI or for custom applications – through the Remote Manager APIs. Enabling/disabling of the automatic log transfer feature and the criteria for automatically transferred log messages can be configured at runtime through the Remote Manager.

Explicit log retrieval – the operator can request explicit retrieval of the log files generated on a particular OSGi device.

2.2.14 Fault Alerts
Remote Manager provides a mechanism through which OSGi device applications can raise notifications about some temporal or permanent malfunctions, specific for the concrete application. The alert notifications are transferred to the Remote Manager and made available to the system operators though the Remote Manager GUI console. They can be also received by custom application modules (through the Remote Manager API) which can generate appropriate notifications, e.g. SMS, emails, etc.

Similarly to the log messages, the alerts have severity level, source, and text description. The main difference between alerts and log messages is that the alarms reflect the current status of the system malfunctions, i.e. the alert is automatically closed when the fault situation is resolved.

2.2.15 Application Monitoring
Remote Manager supports retrieving of monitoring data provided through the standard OSGi Monitor Admin service. The Monitor Admin service can be used by any application running in the OSGi framework to export arbitrary runtime status information, which can change dynamically.

2.3 TR-069 Device Management
Remote Manager supports configuration and firmware update of customer devices (CPEs) over TR-069 protocol standardized by the Broadband Forum. The basic features of the Remote Manager TR-069 based device management are:

- Support for TR-69 Amendment 5 (including Annexes F and G)
- Support for Software Module Management as specified in TR-069 Amendment 5 and TR-157 Amendment 3
- Support for TR-069 based management of OSGi devices compliant with the OSGi Residential Management Tree specification
- Remote monitoring and control of home network appliances (e.g. ZigBee, Z-Wave devices) through an OSGi based gateway (based on the Home Device Management layer of the mBS Smart Home)
- Support of custom data models and vendor specific methods - sophisticated device modeling framework, allowing representation of the manageable elements of the devices as domain-oriented objects, e.g. OSGi bundles, software modules, processes, etc. This allows convenient and intuitive monitoring and manipulation of the devices by the operators. Remote Manager provides built-in object data models for OSGi devices, as well as for some of the standard (Broadband Forum defined)
profiles, and it supports adding of additional custom provided object models for concrete device classes/models

- Secure communication - connections over HTTP/HTTPS with Digest and Basic Authentication
- Support downloading of firmware updates and software modules from the Remote Manager Software Repository
- Configurable (data model-based) retrieval/synchronization of device parameters - controllable (rule-based) way for initial retrieving and subsequent refreshing of the device parameters, based on the knowledge provided by system integrator/operator about data model (profiles) supported by concrete classes of devices. This provides significant improvement in the performance and scalability of the system and reduces the network bandwidth utilization

2.4 TR-069 OSGi Device Management

ProSyst also provides a management solution for TR-069 enabled devices running or proxying OSGi platforms. It is compliant with OSGi Residential Management Tree specification, which defines a TR-069 compliant management interface to the OSGi framework and its resources.

The Remote Manager executes TR-069 methods over the parameter objects defined in the OSGi model. These objects provide access to functionality related to the essence of an OSGi service platform, i.e. to dynamic deployment and management of bundles, setting up bundles by means of configuration dictionaries, etc. The key feature of the TR-069 OSGi Device Management Package is the user-friendly control over TR-069 devices running an OSGi environment.

To provide convenient management of the OSGi features, Remote Manager uses the following OSGi-related parameter nodes with Support:

Device.Services.
  - **OSGiNumberOfEntries** – Represents the number of OSGi platforms on the device
  - **OSGi.<instance_id>**

The server-side stack of the Remote Manager and the client-side stack of the mBS are not strictly bound to each other – i.e. due to the standard and the general implementation, it is possible to use Remote Manager for management of any TR-069 device (not just based on mBS/OSGi/Java), and it is possible to use other TR-069 compliant DM systems for remote management of the mBS.

2.5 Mobile (OMA DM-Based) Device Management

Remote Manager supports management of devices over the OMA DM protocol. OMA DM protocol is defined by Open Mobile Alliance (http://www.openmobilealliance.org) and is widely used in mobile devices.

With the means of the flexible and pluggable architecture of the Remote Manager it is quite easy to implement any kind of management object (available on the device) to be managed from the backend. Yet the Remote Manager also provides built in management of some of the common management objects on an OMA DM enabled devices.
Remote Manager 6.1

The device manager of Remote Manager’s OMA device management system offers means for control of OMA DM devices through the Remote Manager generic model for management of arbitrary device types. The model is based on the Control Unit abstraction, and allows representing miscellaneous devices and their resources as hierarchies of control unit objects sharing a common interface for control.

The Remote Manager obtains a device’s configuration by inspecting the control unit structure and querying the values of the exported state variables and is able to change the device configuration by invoking the corresponding actions.

The Mobile Device Manager provides a generic representation of OMA DM enabled devices as control units by implementing Control Unit Providers responsible for the following features:

- Communication with devices over the OMA DM protocol - The system uses Remote Manager’s implementation of the server-side part of the OMA DM Protocol (including SyncML Representation Protocol) and several network transport bindings.
- Representing OMA DM enabled devices as hierarchical sets of control unit objects - This imposes the need of creating a flexible way for mapping between the data models supported by OMA DM and by Remote Manager – DMT and the control unit models, respectively. Such a mapping mechanism is achieved by means of control unit to DMT Mapping (CU-DMTM) files. You can define such a file by using a dedicated XML format for describing the DMT structure of a particular class of devices or their specific management objects as a set of control units.

2.6 Mobile Device Types

As other device classes supported by Remote Manager, the OMA DM devices are also identified by their type and unique ID. The unique ID usually corresponds to the IMEI of the device and is provided by the device during the initialization of the OMA DM management session. The ID is retrieved from the DevInfo management object (./DevInfo/DevId).

The Remote Manager assigns type Remote Manager.generic.oma-dm.device (Generic Mobile Device) to each device, which is not recognized as a more concrete custom or built-in Remote Manager mobile type (such as Mobile OSGi Device).

2.6.1 Generic OMA DM Support

OMA DM Specification defines several common management objects for every OMA DM enabled device. These include:

- ./DevInfo Management Object with information about the device
- ./DevDetail Management Object with details on the device software
- ./DMAcc Management Object for device preconfigured OMA DM account

Remote Manager provides a build-in support for these common managements objects. There are some differences in the structure of the OMA DM standard objects between versions 1.1 and 1.2 of the OMA DM protocol. The Remote Manager automatically handles the differences and provides adequate representation of the standard management objects.
2.6.2 OSGi Mobile Management Objects

OSGi Alliance has released OSGi Mobile Specification R4 that intends to unify the management objects on OSGi enabled devices to be managed via OMA DM protocol.

This specification includes the following management objects for OSGi devices that are supported by the Remote Manager:

- Deployment – for deploying and managing software components on OSGi enabled devices
- Policy – for managing device security policies
- Configuration – for configuring devices
- Log – for extracting device logs
- Monitor – monitoring different monitorable indicators on the device
- Application – for lifecycle management of installed on the device applications

Management Objects introduced by the Remote Manager are:

- **Bundle** (`./<path_to_osgi_root>/Remote Manager`) – Provides support for getting information about OSGi-compliant bundles and for executing basic lifecycle operations on them as such a feature is not accessible over the OSGi-defined management objects.

Management Objects introduced by the OMA Alliance are:

- **SCOMO** (`./SCM`) – Stands for Software COmponent Management Object. Provides support for installing different types of software components in accordance with the target device features. Presently, the mBS Mobile Runtimes by ProSyst implement the SCOMO.
- **FUMO** (`./FUMO`) – Stands for Firmware Update Management Object. Provides support for the DownloadAndUpdate action on existing FUMO node.
- **Lock and Wipe** (`./LAWMO`) – Provides a sub-tree for locking a device from unauthorized use and for wiping data.

Scripting and Rule-Based Management

The scripting language, adopted in Remote Manager, for runtime programming and appointing management tasks over the system is **Groovy** – a multi-faceted language for the Java platform.

Groovy scripts may access dedicated APIs that are part of the Remote Manager functionality, therefore organizing arbitrary management scenarios. Scripting can be used to appoint various management actions upon devices managed by Remote Manager (install software, configure devices), as well as to operate the backend Remote Manager system itself (operate system configuration, generate alert notifications and reports).

A **Groovy** script may be used for a:

Task, when launched as a single execution. Script executions are called Scripting Tasks. Once a script is launched it will be turned into a Task.
Rule, when used for defining management automation. A Rule as a general concept has two basic parts:

- **Condition**
- **Action**

When the **Condition** is satisfied, its **Action** (a Task or a set of Tasks) will be executed.

The system that organizes scripting in Remote Manager, launching of Tasks and Rules out of Groovy scripts, provides execution statistics and detailed execution monitoring is called **Remote Manager Rule Engine**.

**Control Unit Filters** are used generically for searching and filtering of devices and components in the Remote Manager system. They can be used:

- to appoint extensive searches in the system
- in the scopes of Tasks and Rules to identify the set of devices to be involved in mass executions and automations
- in the GDM Script Service as method arguments for controlling the filtered set of tweaked devices components via Remote Manager Scripting
- in the GDM Java API as method arguments for programming of java applications and Remote Manager plugins

There are scripts that are already generated and available in the system:

- DM Script Service
- Alert Script Service
- Log Script Service Device
- Groups Script Service
- Software Repository Script Service

The Remote Manager maintains a persistent queue for management/control commands sent to each device. If the device is not currently connected to the Remote Manager – these commands are sent when the device is connected again.
3 Software Repository

Remote Manager Software Repository is a backend software and content database compliant with J2EE Client Provisioning specification (JSR-124), which maintains the meta-information and content files of the software components (applications, system modules/drivers, firmware update files, etc.). It can be remotely provisioned on the target devices through the Remote Manager.

The main features of the Software Repository are:

**Support for various software components** – The Software Repository maintains the functional characteristics of OSGi Bundles, Bundle Groups, Deployment Packages, Provisioning Archives and MIDlet Suites. A powerful characteristic of the extensible Software Repository model is the ability to widen the range of supported software component types.

**Common persistent storage** – The software resources and all additional properties ensuring the successful delivery and deployment on the client devices are stored persistently in the database.

**Dependency management** – The Software Repository introduces mechanisms for manual and automatic definition of the dependencies and the compatibility relations between the software components and their versions.

**Software requirement management** – Required device capabilities can be defined automatically or by the system administrator.

**Dependency solving and capability matching** – The Software Repository determines the software components and versions which are suitable for the different devices.

**Device Platform Capabilities** – Maintenance of platform properties including compatibility relations between them.
4 Remote Manager GUI – Management Console

Remote Manager provides system operators with rich and interactive GUI tools for performing system administration and remote device management and monitoring tasks. The Management Console is available in two forms:

- Stand-alone (Windows-like) GUI applications – connecting remotely to the Remote Manager server.
- Web-based interface – provides GUI to the Remote Manager accessible through a web browser.

Both versions of the Remote Manager Management Console provide access to the full functionality of the Remote Manager. The Management Console is based on the Eclipse RPC and RAP technologies and is open for customization and extension with custom modules and functionality.
5 API and Integration Interfaces

Remote Manager provides an extensive and open API to its full functionality allowing extending, customization and integration of the system with custom applications and backend systems.

The Remote Manager API is available in the following forms:

Remote Manager OSGi API – Remote Manager servers are based on OSGi technology, which allows deployment of custom applications inside the Remote Manager OSGi framework. These custom bundles may use the Remote Manager API in the form of OSGi services.

Remote Manager Remote Java API – API in the form of Java library providing any external Java based application/system with remote interface to the Remote Manager.

Remote Manager JEE Resource Adapter – Java Connector Architecture compliant API providing JEE applications with standard interface to the Remote Manager.

Web Service API – provides SOAP 1.2 and REST-based API to either non-Java (.NET, C++, others) or Java-based applications.
6 Security

6.1 User Authentication and Authorization

Remote Manager User Management package provides rich functionality for creating user accounts and user groups and for assigning the access rights on a user and/or user group level. The user accounts are used for getting access to the Remote Manager through the Remote Manager GUI console applications and also for accessing (by external systems and applications) the remote backend APIs of the Remote Manager described in the previous paragraph.

6.1.1 User Authentication

Remote Manager supports basic (username/password based) and secure (based on personal user certificates) authentication methods.

The Remote Manager provides pluggable hook mechanism for authenticating connections coming from an OSGi device. This mechanism is an additional guarantee that only authenticated devices will be connected to the Remote Manager, because the device ID can be obtained comparatively easy and it is not enough to connect the device to the backend securely.

The authentication can be bidirectional - the device can also verify some credentials sent from the Remote Manager to prevent control from unauthorized management system.

The Remote Manager provides two ready-to-use verifiers:

- **Basic Authentication Verifier** – the Remote Manager can be configured to require username/password-based authentication and check the provided client credentials.

- **Client Certificate Verifier** – the Remote Manager can be configured to verify that the client certificate Common Name is equal to the device identifier.

If the users need their own “device – Remote Manager” authentication, a custom Authentication Verifier can be added.

6.1.2 User Authorization

Remote Manager utilizes flexible and fine-grained access control model based on roles. The control can be done either on specific user level, as well as on a defined group of users, such as system administrators, service providers, end users. The access control restrictions to the protected resources and functionality maintained by the Remote Manager is enforced on all backend interfaces provided by the Remote Manager – GUI tools and remote APIs provided for integration with external systems and applications.

Below are some of the build-in Remote Manager roles, which can be assigned or not to the users/user groups (thus allowing or denying user access to particular resources):

- **Bundle-Import** – grants rights to import new bundles into the Remote Manager repository

- **Bundle-View(<bundle_id>)** – parameterized role granting rights to view/use the client bundle available in the Remote Manager Software Repository, with the specified
<bundle_id>. This ID may correspond either to single OSGi bundle or composition (application/service).

**Bundle-Manager(<bundle_id>)** – parameterized role granting rights to edit (e.g. rules, properties, etc) the client bundle available in the Remote Manager Software Repository, with the specified <bundle_id>. This ID may correspond either to single OSGi bundle or composition (application/service).

**Device-View(<device_node_path>)** – parameterized role granting rights to view/monitor a specified device registered in the Remote Manager. The <device_node_path> may refer either to concrete device or to a group of devices (in the Remote Manager device tree).

**Device-Manager(<node_path>)** – parameterized role granting rights to manipulate (e.g. install/upgrade software, change settings, etc.) a specified device registered in the Remote Manager. The <device_node_path> may refer either to concrete device or to a group of devices (in the Remote Manager device tree).

There are also other roles supported by the Remote Manager – e.g. related to scripting and rule-based management, tasks, certificates, user management, etc. Additional custom roles can also be defined in the Remote Manager and used by applications to guard access to their protected resources.

The roles can be defined and assigned to the users and user groups through the mConsole GUI or through the Remote Manager API.

### 6.2 Secure Network Communication

Remote Manager supports secure (SSL or HTTPS based) communication over its remote interfaces – with the devices and the backend interface – with the Remote Manager GUI console applications and external systems.

### 6.3 Certificate Management

Remote Manager provides common certificate management module, which maintain Remote Manager own private key and certificates, as well as trusted certificates. There can be different certificates (own and trusted) specified for the different interfaces of the Remote Manager – e.g. for Remote Manager Message service, for HTTP server, for backend API access, for user authentication.

Remote Manager Certificate Manager supports all of the most widely used formats for importing and exporting of the certificate and private key data. It also support certificate revocation lists (CRL).
7 Remote Manager Scalability and High Availability

Remote Manager supports both **vertical** (achieved by using more powerful server hardware - number of CPUs/CPU cores and amount of RAM) and **horizontal** (using a cluster of multiple Remote Manager servers, which number can be increased with increasing load) scalability approaches.

Remote Manager Cluster ensures high scalability and availability of the Remote Manager system, by allowing a number of backend servers to act as a single management server. Remote Manager supports different clustering models:

- **Static load distribution** – every management server manages a different sub-set of explicitly specified devices

- **Dynamic** – more than one backend server instance is managing the same subset of devices, distributing the load automatically and dynamically

- **Mixed** – the system may contain multiple clustered servers, each of them responsible for a different sub-set of devices, and every cluster contains multiple backend servers, which distribute the corresponding device sub-set dynamically

The diagrams below show three Remote Manager example deployment configurations:

### 7.1 Single-server configuration

![Diagram showing Single-server configuration](image)

The figure depicts the simplest Remote Manager configuration, suitable for development and evaluation purposes. It consists of only one backend server, which hosts Remote Manager Control Center (CC), Management Server (MS) and Remote Access Server (RAS).
7.2 Dynamic-clustered configuration

The figure shows a Remote Manager multi-server configuration with a single clustered Management Server and a single RDBMS (which in turn can be clustered). The Clustered Management Server manages all registered devices and distributes the load between the server nodes. RAS servers are deployed on every Remote Manager host and distribute requests from the other external backend systems.
7.3 Mixed-clustered configuration

This figure shows a multi-host configuration, with several clustered Management Servers, each of them working with a separate database server for storing the device configuration data. This multi-host configuration is suitable for distributed system deployments. Hosts of one MS cluster and the corresponding DB server are connected in a local area network (fast speed). Different MS clusters, CC and RAS cluster communicate via Internet.
8 Platform Requirements

8.1 Hardware Requirements

- Remote Manager Backend Server

<table>
<thead>
<tr>
<th>RAM</th>
<th>1024 MB minimum</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2048 MB or more recommended</td>
</tr>
</tbody>
</table>

| Disk space, required for Installation | 400 MB for the Development Edition. The space for the Production Edition depends on selected components Recommended free space - at least 500 MB |

- Remote Manager Management Console

<table>
<thead>
<tr>
<th>RAM</th>
<th>1024 MB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disk space, required for Installation</td>
<td>50 MB</td>
</tr>
</tbody>
</table>

8.2 OS & JVM Requirements

For Remote Manager Server(s) a Java SE VM equivalent to Oracle/Sun JDK 1.7 or higher is required. Any OS equipped with such JVM can be used (Linux, Windows, Solaris, etc.).

For Remote Manager OSGi Management Agents the system requirements are J2ME CDC 1.0/Foundation 1.0 compatible JVM.

For the Remote Manager Management Console it is required to have Java SE VM equivalent to Oracle/Sun JDK 1.7 or higher.

8.3 Supported OSGi Frameworks for Remote Manager OSGi Management Agents

The supported OSGi frameworks are:

- ProSyst mBS SDK 6.1 or higher
- Apache Felix 2.0, 3.0.6 and 3.2.0
- Equinox 3.5.1, 3.6.1 and 3.6.2

The Remote Manager should be compatible with any other OSGi framework compatible with the OSGi Specifications Release 5 or higher.

8.4 Database Requirements

The Remote Manager requires an external (3rd party) SQL RDBMS. The supported DB servers are:

Oracle Database 9i, 10g, 11g, 12c (recommended for large scale deployments)
MySQL 5.1 or higher

H2 1.1.11 – an embedded Java RDBMS, which is included in the Remote Manager distribution for development purposes only and not recommended for production deployments.

DB Storage Space

The following general guideline can be used for determine needed DB storage space for Oracle DB: 500 MB data space + 100 MB transactional space + 100 MB of additional data space for every 1000 devices registered in the system.

8.5 Supported Non-OSGi Devices

OMA DM compatible devices – Remote Manager supports OMA DM devices compatible with OMA DM protocol version 1.1 & 1.2

TR-069 devices – Remote Manager supports management of TR-069 protocol (TR-069, TR-069 Amendment 1, TR-069 Amendment 2 and TR-069 Amendment 3) compliant devices.
9 Appendix A: Resources

You can find more resources about the Remote Manager at:

- [http://dz.prosyst.com](http://dz.prosyst.com) - ProSyst Developer’s Zone