Design Document

Remote Monitoring of Power Consumption in a Cloud Server Network

Version DD 1.2 - May 19, 2014

Customer: Patrik Arlos

Chief Executive Officer: Dragos Ilie

Developer Team Zenoss		
Student 1	Name	Sarat Chandra Pasumarthy
	E-Mail	sapa14@student.bth.se
	Social Security Number	921231-0297
Student 2	Name	Manish Reddy Guduru
	E-Mail	magu14@student.bth.se
	Social Security Number	930706-0518
Student 3	Name	Sukesh Kumar Tedla
	E-Mail	sute14@student.bth.se
	Social Security Number	940224-5352
Student 4	Name	Prakhyath Neelagiri
	E-Mail	prne14@student.bth.se
	Social Security Number	930326-1359
Student 5	Name	Hadassah Pearlyn Nagathota
	E-Mail	hana14@student.bth.se
	Social Security Number	930719-9548
Student 6	Name	Preetham Kalakuntla
	E-Mail	prka14@student.bth.se
	Social Security Number	920523-3779

Preface

The document is the third release of the Design Document concerned with this project. This document is primarily aimed at the parties involved in this project, that is, the customer and the company involved in providing solution to the customer's needs. The company comprises of Chief Executive Officer (CEO) and the developer team Zenoss. Firstly, the document starts with glossary and abbreviations which define the technical terms and abbreviations used in the document. Secondly, we have sections describing the purpose of each module comprising the system architecture, which requirements does it satisfy and its interaction with other modules. Each section has detailed designs of the module's implementation and related test specifications. Lastly, there is a references section giving a list of the references used for the creation of this document.

This is version DD1.2 of the document and the version history is as follows:

Released v DD1.2 on 2014-05-12

- Made changes in Figure 3 by including SNMP as one of the data retrieval methods.
- Added SNMP as one of the data retrieval methods in Section 4.
- Modified M1T1, M1T2, M2T1 and M2T2 tests by providing deterministic results.
- The purpose of tests M3T6 and M3T7 have been explained in detail.

Released v DD1.1 on 2014-05-12

- Description of the databases is added in section 3.1
- Figure3 has been changed.
- Text in section 4.1 is added to describe Figure3.
- New module test requirements M3T1, M3T2, M3T3, M3T3, M3T4, M3T5, M3T6, and M3T7 have been added and described.

Released v DD1.0 on 2014-05-05

- Initial release

1. Glossary and abbreviations

API- *Application Program Interface*

An interface, generally specified as a set of operations, that allows access to an application program's functionality.

CN- *Computer Node*

HTTP- *Hypertext Transfer Protocol*

It is an application protocol for distributed, collaborative and hypermedia information systems.

NMS- *Network Management System*

It constantly monitors a computer network and notifies the network administrator in case of outages.

PDU- *Power Distribution Unit*

It is a device designed to distribute electric power with multiple outputs.

SNMP- *Simple Network Management Protocol*

It is an internet-standard protocol for managing devices on IP networks.

SSH- *Secure Shell*

It is a cryptographic network protocol for secure network services between two networked computers.

UPS- *Uninterruptable Power Supply*

It is an electrical device that provides emergency power when the input power supply fails.

VM- *Virtual Machine*

It is a software based emulation of a computer. Virtual Machines operate based on the computer architecture and functions of a real or hypothetical computer.

Test Identification string

Used to uniquely identify tests for modules. Follows the following format:

M <Module no.> T <Test no.>

2. Module 1 – Data Retrieval

This module illustrates the process in which the data is retrieved from the remote devices. The local server, after getting a request from the user, polls the concerned devices using SNMP, to retrieve the data about the power consumption of the devices. This process is repeated multiple times to get the data in real time scenario.

This module interfaces with module 2, which is device database management module. The data is dynamically obtained and the database is updated consequently.

2.1. *Detailed design*

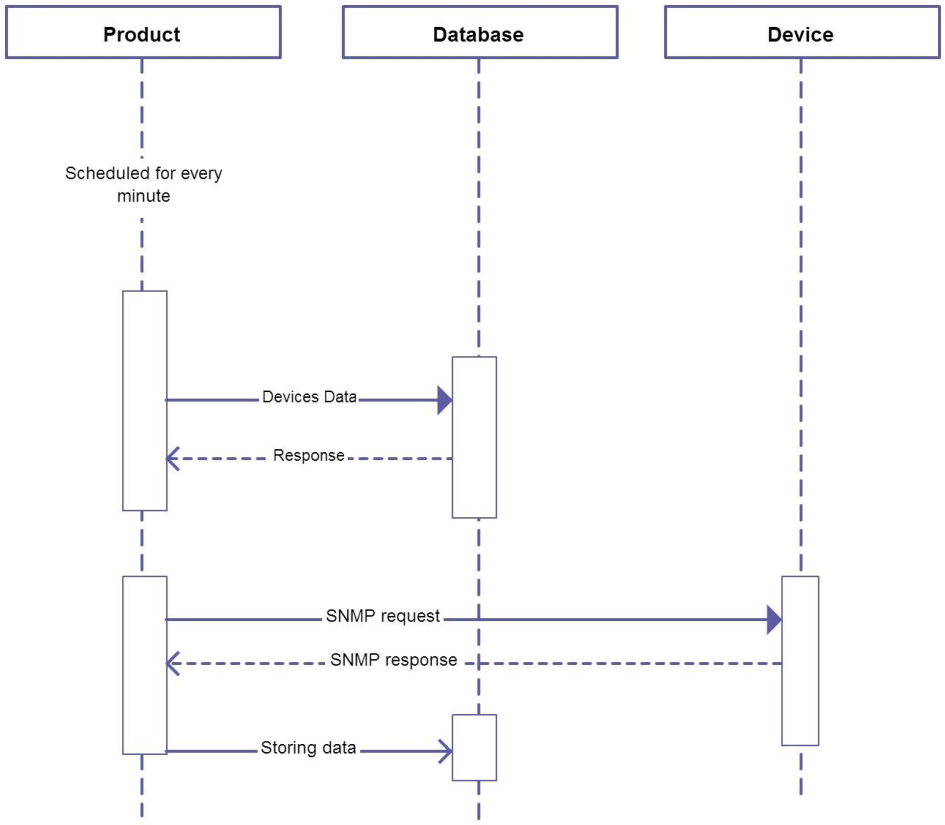

Figure1: UML diagram illustrating the process of data retrieval using SNMP

The product first obtains the device data from the database which is storing the parameters such as IP address, SNMP community, vendor, etc. of the devices to be monitored. Based upon the response received (here the device information) it dynamically polls the device every minute to retrieve the data related to the device. This is carried out through a SNMP request. After that the device sends a corresponding SNMP response concerned to the request. Then the response is processed, after which the database is updated.

2.2. *Unit test plan*

The tests which are going to be run for this module are as follows:

M1T1 - Whether the product is performing SNMP communication

Purpose	The test is primarily conducted to analyze whether the module is performing SNMP communication with the device under test.
Requirements	UFR4, SFR3, SNR1
Environment	To perform this test the module has to be tested against a SNMP enabled device. There should be a network connection such that the module and the device can interact with each other.
Operation	The details of the SNMP enabled device should be entered in the module. Then the SNMP requests are generated dynamically by the module to the device. This is performed through the program which is present in the product.
Expected result	The SNMP request should return with the corresponding SNMP response. This response should be observed by the tester. This response should match the response expected from the request.

M1T2 - Whether the retrieved data is being updated in the database

Purpose	The test is primarily conducted to check if the data retrieved through SNMP communication is being updated in the database or not.
Requirements	SFR3, SFR4, SNR1
Environment	To perform this test the module has to be tested against a SNMP enabled device. There should be a network connection such that the module and the device can interact with each other.
Operation	The details of the SNMP enabled device should be entered in the module. Then the SNMP requests are generated dynamically by the module to the device. The data that is retrieved from the SNMP response should be dynamically updated in the database. This is performed through the

program which is present in the module. Then the tester should run operational commands associated with that database through which the tester will have know whether the database is updated from the response or not.

Expected result The SNMP response that is obtained should be updated in the database based upon the program present in the module. This should be indicated to the tester when he/she runs the associated database operations.

3. Module 2 – Device Database Management

This module illustrates the architecture associated with the database management. The network devices information is provided by the customer. The managing entity functions to send request to the remote server to retrieve the data from the devices. The data when received from the server is stored in the local database. This data from the local database is used to create graphs and provide related notifications to the customer.

3.1. Detailed design

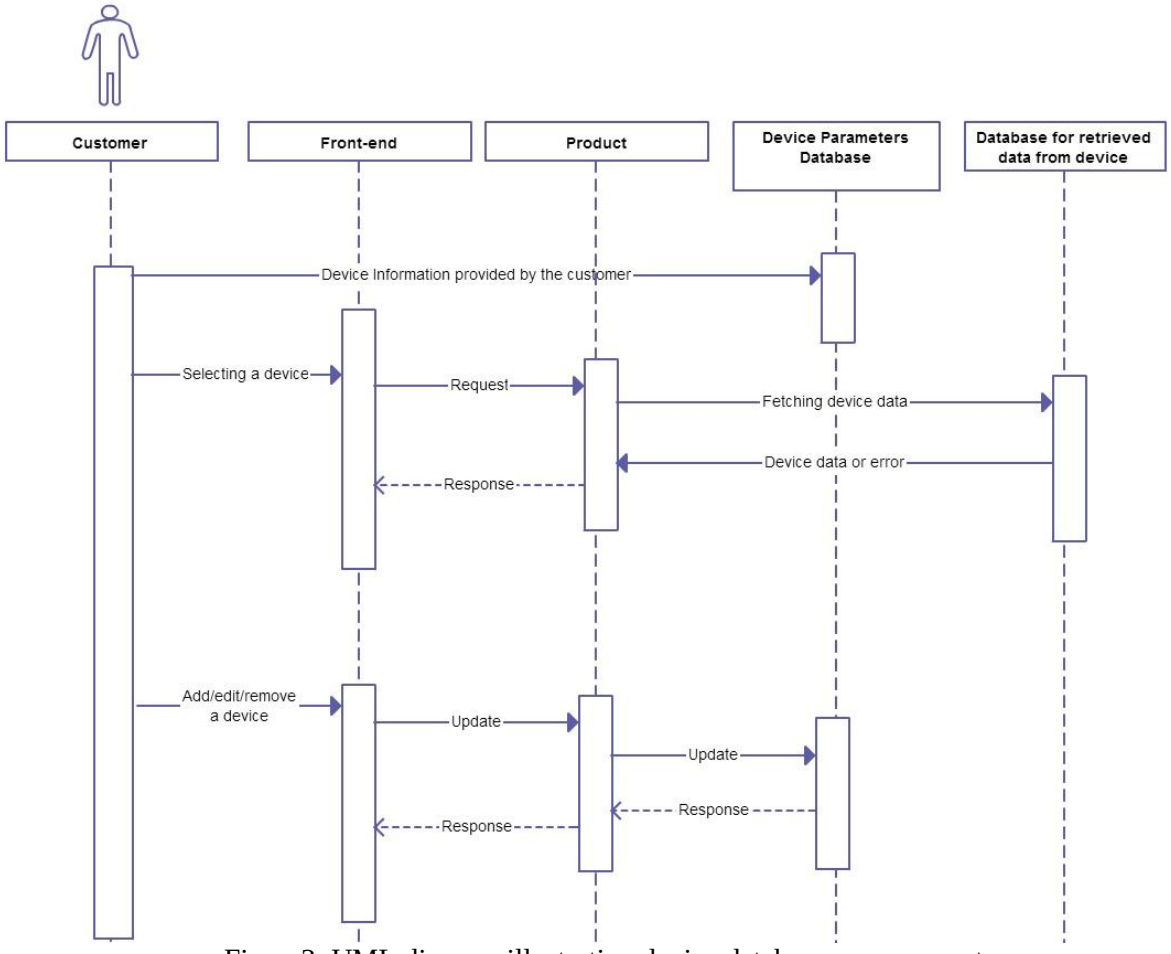

Figure2: UML diagram illustrating device database management

The module is designed to enable effective device database management. The module will enable synchronization between the device data and the corresponding data which is retrieved from the requests which were sent to the device. The corresponding information associated with the device is updated when there is a modification associated with the device. Here the two different databases which are present are part of the same database server. They are both located in one location. The two different databases are maintained for efficient mapping between the device and its corresponding data.

3.2. *Unit test plan*

M2T1 - Whether the module is updating the device information in the database:

Purpose	The test is primarily conducted to analyze whether the module is updating the device information.
Requirements	UFR1, SFR4
Environment	To perform this test the module has to be tested against a SNMP enabled device. The database should be configured in a proper manner. There should be a network connection such that the module and the device can interact with each other.
Operation	The details of the SNMP enabled device should be entered in the module. Then the SNMP requests are generated dynamically by the module to the device. This is performed through the program which is present in the product.
Expected result	The SNMP request should return with the corresponding SNMP response. This response should be observed by the tester. This response should match the response expected from the request.

M2T2 - Whether the module is updating the information retrieved for a corresponding device.

Purpose	The test is primarily conducted to check if the data retrieved through SNMP communication is being updated in the database or not. Most importantly this test is conducted to verify whether the information retrieved is for the corresponding device that has been polled.
Requirements	UFR1, UFR4, SFR4
Environment	All the modules mentioned should be integrated with each other, at least one device should be working and the user should be able to

probe this device to obtain data dynamically in real time.

Operation	The tester first logins to the front end. The tester should perform functions associated with the device. Then the tester can select a particular device to monitor the data which has been retrieved. The tester should perform run tests associated with the database to retrieve the values that have been stored and updated.
Expected result	When the operation is performed, the database should be updated based upon the corresponding information retrieved. The data obtained should be from the device which has been polled.

4. Module 3: Using the product on the front end

This module illustrates how the output is presented in the front end. The front end used here is Zenoss. The data from the product is retrieved and open source tools are used to create orderly graphs in real time. The graphs are plotted between power consumption and time on different timescales. The product also notifies the user if there are any important developments in the network especially a power failure of any of the device. The user can add, select, edit and remove any particular device for which the power consumption is to be monitored.

4.1. Detailed design

Selecting a device:

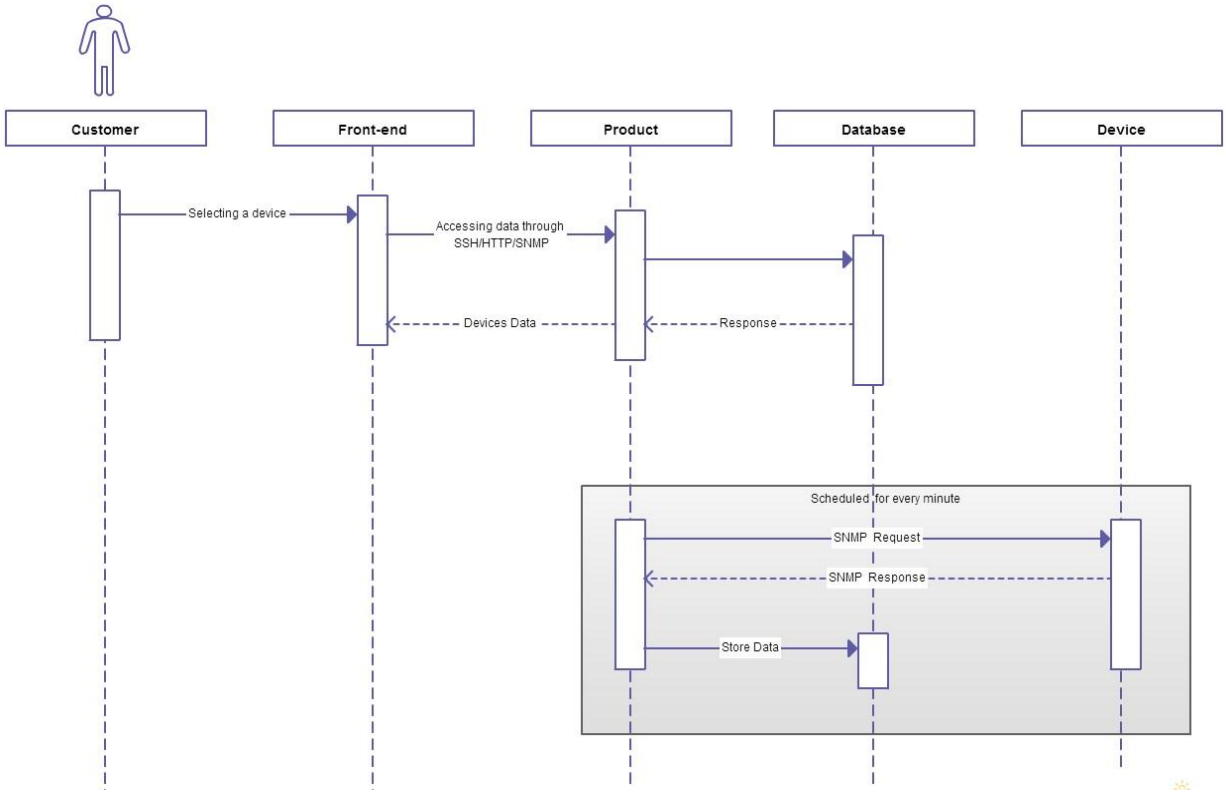

Figure3: Selecting a Device

Here when the user selects a particular device to monitor, a request for device data is sent. The request is sent and the response is received. This is either through HTTP or SSH or SNMP. The request is processed and the device data is obtained from the database. This information stored in the database is the data which is updated from the time the device is added. The user can monitor power consumption information in the form of graphs on different time scales. After that, the user gets the information in a dynamic form and the graph is updated based on the probe rate. The devices are polled using SNMP.

Adding/Editing/Removing a device:

Here the methods of adding/editing/removing a particular device to be monitored by the user from the front-end is depicted and explained. The request is sent and the device data is processed in the database accordingly. This information stored in the database is the data which is updated from the time the device is added (in cases of editing and removing). A response is sent to convey the success/failure of the operation.

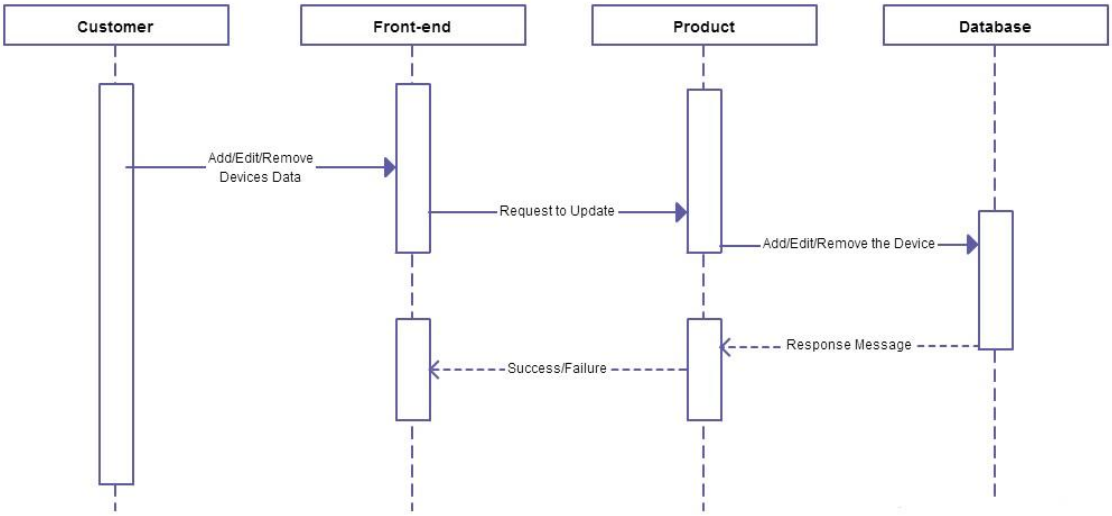

Figure4 : UML diagram illustrating the process of adding/editing/removing a device

4.2. *Unit test plan*

The tests which are going to be run on this module are as follows:

M3T1 – Adding a device:

Purpose	The test is primarily conducted to check the user is able to add a device.
Requirements	UFR1, UFR4, UFR5, UFR6, UFR7, SFR4
Environment	All the modules mentioned should be integrated with each other, at least one device should be working and the user should be able to probe this device to obtain data dynamically in real time.
Operation	The tester should login into the front end. Then, the tester should add a device. This is achieved by following the steps provided in the user documentation.
Expected result	When the device is added successfully it should be reflected in the front end.

M3T2 – Deleting an existing device:

Purpose	The test is primarily conducted to check whether the user is able to delete an existing device.
Requirements	UFR1, UFR4, UFR5, UFR6, UFR7, SFR4
Environment	All the modules mentioned should be integrated with each other, at least one device should be working and the user should be able to probe this device to obtain data dynamically in real time.
Operation	The tester should login into the front end. Then, the tester should delete an existing device. This is achieved by following the steps provided in the user documentation.
Expected result	When an existing device is deleted successfully it should be reflected in the front end.

M3T3 – Monitoring an existing device:

Purpose	The test is primarily conducted to check whether the user is able to monitor an existing device.
Requirements	UFR1, UFR4, UFR5, UFR6, UFR7, SFR4
Environment	All the modules mentioned should be integrated with each other, at least one device should be working and the user should be able to probe this device to obtain data dynamically in real time.
Operation	The tester should login into the front end. Then, the tester should select an existing device to monitor. This is achieved by following the steps provided in the user documentation.
Expected result	The user should successfully be able to select an existing device and monitor it.

M3T4 – Edit an existing device:

Purpose	The test is primarily conducted to check whether the user is able to edit the information of an existing device.
Requirements	UFR1, UFR4, UFR5, UFR6, UFR7, SFR4
Environment	All the modules mentioned should be integrated with each other, at least one device should be working and the user should be able to probe this device to obtain data dynamically in real time.
Operation	The tester should login into the front end. Then, the tester should select an existing device. After that the tester should edit the information of that device. This is achieved by following the steps provided in the user documentation.
Expected result	The tester should be able to successfully edit the information concerned with the device he/she selects.

M3T5 - Updating of graphs

Purpose	The test is primarily conducted to check whether the graphs are being updated using the data retrieved from the devices every 1 minute.
Requirements	UFR1, UFR2, UFR4, UFR5, SFR4
Environment	All the modules mentioned should be integrated with each other, at least one device should be working and the user should be able to probe this device to obtain data dynamically in real time.
Operation	The tester should add a device or select/edit/remove an existing device. The power consumption graphs related to the device should be displayed from the time it has been added. (This is subject to the condition that the manager is running as a demon). The graph should be displayed on an XY scale. Then the tester should monitor whether the graph is being updated every minute.
Expected result	The tester here should ideally observe that the graph for a particular device is being updated every minute.

M3T6 - Real-time Scenario Test

Purpose	The main purpose of this is test is to verify whether the product is reacting to changes in the dynamic network environment. This is to establish the credibility of the product in real time scenario.
Requirements	UFR1, UFR2, UFR3, UFR4, UFR5, SFR4
Environment	All the modules mentioned should be integrated with each other, at least one device should be working and the user should be able to probe this device to obtain data dynamically in real time.
Operation	The tester here should have physical access to the devices that are being monitored. The real time scenario test part 1 consists of the following: • Disconnecting a particular device from the network dynamically. The tester then should monitor the result of the operations performed on the front end.
Expected result	The result should be as follows: • The graph should not be updated for the disconnected device.

M3T7 – Notifications from the devices

Purpose	The main aim of this test to see whether the product is able to deliver notifications in the dynamically changing network environment.
Requirements	UFR1, UFR2, UFR3, UFR4, UFR5, SFR4
Environment	All the modules mentioned should be integrated with each other, at least one device should be working and the user should be able to probe this device to obtain data dynamically in real time.
Operation	The tester here should have physical access to the devices that are being monitored. The tester should perform the following: • Disconnecting power supply to a UPS which is being monitored. The tester then should monitor the result of the operations performed on the front end.
Expected result	The result should be as follows: • The tester should receive period updates based on the severity of the condition. The severity here is dependent on the battery capacity of the UPS.